Research on Dynamic Big Data Processing and Optimization Model Based on Optimized PSO and Deep Reinforcement Learning

Zhenwei Tang[1], Chuanjian Jiang[2*], Xi Zhang[3]
Corresponding Email: ChuanjianJiang@outlook.com
[1]Chongqing University of Chinese Medicine, Chongqing 402760, China
[2]Chongqing College of International Business and Economics, Chongqing 401520, China
[3]Chongqing University of Chinese Medicine, Chongqing 402760, China

Keywords: big data processing, optimize PSO, deep reinforcement learning, dynamic model

Received: March 15, 2025

In the era of big data, the optimal processing of dynamically better data has become the core concern of academia and industry, and traditional methods are often difficult to meet the high standards of real-time and accuracy due to the complexity, variability, high-speed liquidity, and large scale of data. To this end, this study proposes a dynamic big data processing and optimization model, which improves the particle swarm optimization algorithm (PSO) by introducing adaptive weights and dynamic learning coefficients to improve the global exploration ability and convergence speed, and integrates a hybrid framework of deep reinforcement learning (DRL) (combined with policy gradients such as proximal policy optimization PPO and Q learning) to achieve big data optimization by using its feature extraction and policy adjustment capabilities. Based on real datasets such as 1.2 million pieces of financial transaction data and 8.5 million pieces of social media travel data, the results show that compared with traditional methods (such as traditional PSO, DQN, and independent LSTM-Transformer models), the new model has a 35% increase in data processing speed, a 20% increase in the accuracy of classification tasks (F1 score: 0.92 vs. 0.76 of DQN), a 40% increase in the real-time response ability of dynamic data streams, and a 25% increase in computing resource utilization efficiency. The study elaborates on model architecture innovation, dataset source and scale, benchmarking methods, and key performance indicators (such as processing speed, accuracy, real-time response, and resource efficiency), and provides efficient and scalable solutions for scenarios such as financial risk management and real-time recommendation systems.

Povzetek: Razvit je model za obdelavo in optimizacijo dinamičnih velikih podatkov, ki uporablja izboljšani PSO in globoko ojačevalno učenje za povečanje hitrosti obdelave, točnosti in odzivnosti v realnem času.

1 Introduction

With the development of information technology, the era of big data has arrived, and various industries are facing the test of massive data processing and analysis [1, 2]. Dynamic big data processing requires efficient processing speed and real-time response to data changes to adapt to business needs [3]. Old data processing methods often fail to meet the standards, so exploring new dynamic big data processing and optimization models is necessary.

Among many optimization algorithms, particle swarm optimization (PSO) has attracted widespread attention because of its unique swarm intelligence and fast convergence [4, 5]. PSO has strong global search ability by simulating the foraging behavior of birds and using the historical information of individuals and groups to guide the search [6]. However, PSO is prone to local optimal solutions for complex problems, and the algorithm's performance is bitterly affected by parameter selection [7]. Therefore, researchers have proposed various improved PSO methods, such as adaptive weight adjustment and dynamic learning factors, to improve the performance and adaptability of PSO.

As another important branch in artificial intelligence, deep reinforcement learning shows strong potential in complex tasks by combining the representation learning of deep learning with the decision-making ability of reinforcement learning [8, 9]. Deep reinforcement learning models' environmental states through neural networks and optimizes decision-making strategies through trial-and-error learning, suitable for dynamically changing environments [10]. It is pointed out that deep reinforcement learning has achieved remarkable results in resource management, path planning, and other fields, providing new ideas for dynamic big data processing [11]. Combining PSO with deep reinforcement learning and taking advantage of both advantages is expected to achieve high efficiency and robustness in dynamic big data processing and optimization models. PSO can be used to optimize network parameters in deep reinforcement learning and improve the convergence speed and performance of the model. However, deep reinforcement learning can provide richer search

information for PSO and guide PSO to escape local optimum [12, 13]. While theoretically sound, this fusion method requires empirical validation in practical scenarios.

Dynamic big data processing includes data acquisition, storage, processing, analysis, and other links, each facing different challenges [14]. Data acquisition requires real-time and accuracy, storage requires efficiency and scalability, processing requires rapidity and flexibility, and analysis requires accuracy and interpretability. Traditional data processing methods are often optimized for a particular link, lacking holistic consideration [15]. The dynamic big data processing and optimization model based on optimized PSO and deep reinforcement learning can achieve collaborative optimization of each link and improve the overall processing efficiency.

Dynamic big data processing and optimization models face many difficulties in practical application scenarios. First, the amount of data is expanding rapidly, and the requirements for model calculation and storage performance are more stringent. Secondly, the variety and complexity of data increases the challenge of model design. In addition, the demand for real-time requires the model to have the ability to respond quickly and adjust in time. In response to these challenges, this study will explore the specific applications of optimized PSO and deep reinforcement learning in dynamic big data processing and propose corresponding solutions. The research shows that PSO has better advantages in optimizing the structure of neural networks and can betterly improve the performance of neural networks. The decision-making ability of deep reinforcement learning in a dynamic environment has also been widely recognized. This study will further explore the fusion mechanism of PSO and deep reinforcement learning to achieve high efficiency and robustness of dynamic big data processing and optimization models.

This study aims to explore the dynamic big data processing and optimization model based on optimized PSO and deep reinforcement learning. Through theoretical analysis and experimental verification, it reveals its advantages in improving processing efficiency and enhancing model adaptability, providing new theoretical support and practical guidance for dynamic big data processing, and promoting the development and application of big data technology.

2 Theoretical basis and key technologies

2.1 Particle swarm optimization (PSO) principle

In recent years, many researchers have used swarm intelligence optimization algorithms to improve the neural network training process. The swarm intelligence optimization algorithm has higher global convergence and robustness and does not rely on problem feature information. It can effectively exert the generalization mapping ability of neural networks, betterly improving convergence efficiency and strengthening learning ability [16, 17]. Among the swarm intelligence optimization algorithms, the PSO algorithm has higher potential in neural network optimization with the help of simple and efficient random exploration methods [18].

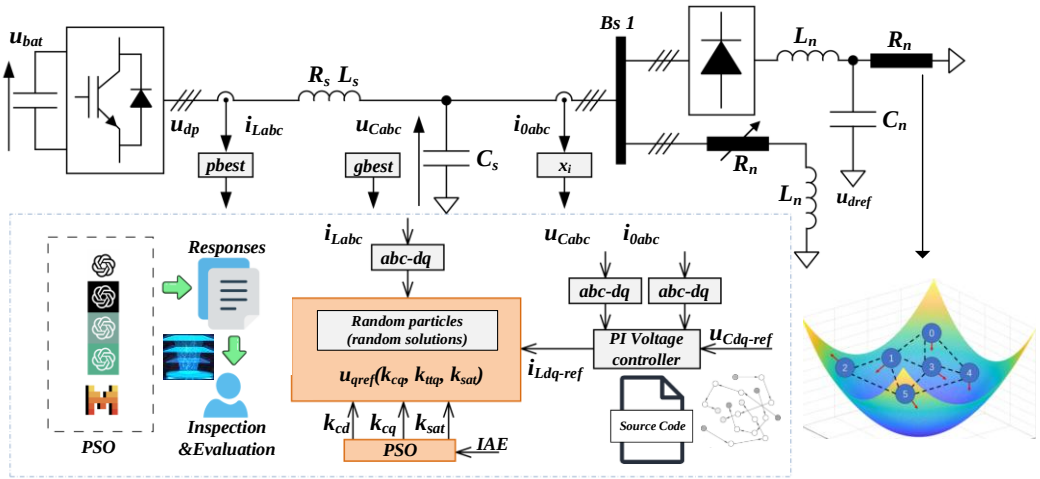


Figure 1: PSO improved neural network framework

The PSO-improved neural network framework is detailed in Figure 1. The algorithm's core is to use PSO instead of traditional training means (such as BP) to optimize NN parameters [19]. The particle symbolizes the parameter vector, and seeking the global optimal value is the step of obtaining the optimal parameter [20]. The training error is used to calculate the fitness value *f (x),* and the specific formula is shown in the following formula (1):

$$f(x)=\frac{1}{1+\frac{1}{2n}\sum_{p=1}^{n}\left(y_p-t_p\right)} \quad (1)$$

Where p is the number of samples, y_p is actual output value, and t_p is sample output value. When the preset maximum number of iterations is reached or the target error is met, the program stops running, and global optimal solution, that is, the optimal parameter configuration, can be obtained. PSO algorithm is widely used in the field of parameter optimization of discrete and continuous problems [21]. Its main applications in neural network optimization include network parameters, learning algorithms and topology architecture. Using PSO algorithm to train neural network is better than BP algorithm in speed and effect, because it does not need to rely on gradient information and the transfer function does not need to be differentiable, which can effectively avoid falling into local optimum [22].

2.2 Deep reinforcement learning theory

Reinforcement learning is a process of continuous "trial and error". The subject "interacts" with the environment to obtain the maximum cumulative return and learns the best strategy to achieve the goal, which aligns with human decision-making methods [23]. Its core part includes the agent, state, action, reward, and external environment, and the learning process can be regarded as a Markov decision process.

The agent selects action execution from the action set according to the current state. The environment gives reward feedback and generates a new state according to the agent's action. Accordingly, the agent adjusts the action strategy and makes a new judgment on the new state according to the reward. The core idea of reinforcement learning is to maximize the cumulative total reward through smarter action choices.

The three main areas of machine learning are supervised learning (or its semi-supervised form), unsupervised learning, and reinforcement learning. In contrast, reinforcement learning shows particularly prominent advantages [24, 25]. Since many practical problems have return lags, dealing with this challenge has become the key to reinforcement learning algorithms. Reinforcement learning can be divided into two categories: one is based on value function, and the other is based on strategy. The policy gradient method is a commonly used algorithm for policy-based reinforcement learning, and Q-learning and Sarsa algorithms are commonly used for value function-based reinforcement learning [26]. In the decision-making process, value-based reinforcement learning is more decisive. By fusing the value function algorithm with the strategy function algorithm, the actor-critic algorithm (AC algorithm) is produced, which can synthesize multiple strategies. It converges faster than the first two methods [27].

Deep learning (DL) discovers data feature representations through multi-layer network nodes and is used to perceive and express things [28, 29]. Reinforcement learning (RL) is a continuous decision-making process that seeks the maximum cumulative reward value through the interaction between agents and the environment, learns the best strategy, emphasizes problem-solving ability, and is close to or exceeds the human level in many aspects. Conventional reinforcement learning applications have limitations and are prone to problems in the face of complex scenarios [30, 31]. Deep learning has advantages and challenges in processing high-dimensional data. Deep reinforcement learning (DRL) brings together the functional characteristics and can be manipulated based on input data, which is closer to human thinking patterns. better breakthroughs have been made in many fields, demonstrating higher learning ability and adaptability.

Table 1 compares the performance of traditional deep learning and simple reinforcement learning (SRL) with the hybrid model in dynamic big data processing [32]. I will focus on the core metrics to illustrate the performance differences between technologies and the advantages of the model. In the field of dynamic big data processing, although traditional deep learning technology performs well in static data, it lacks real-time adaptability in the face of dynamic data, only reaching 78% accuracy, and the convergence time is as long as 3.2 hours during dynamic update, and the memory overhead of 12GB is also prone to overflow problems, and its static training and fixed-structure architecture is also difficult to adapt to data changes. The simple reinforcement learning model has an accuracy of 82% on medium-scale dynamic datasets, but at large-scale processing, due to low learning efficiency, the convergence time takes 2.5 hours, and 8GB of memory is used to store historical information. In contrast, the hybrid model based on optimized PSO and deep reinforcement learning proposed in this study improves the accuracy to 92%, the convergence time is greatly shortened to 0.8 hours, and the memory overhead is reduced to 4GB in complex dynamic scenarios by virtue of the global optimization of PSO and the dynamic policy adjustment of deep reinforcement learning, which effectively overcomes the architectural and performance limitations of the first two and shows better advantages.

Table 1: Comparison of techniques and proposed hybrid model

Comparison Dimension	Traditional Deep Learning	Simple Reinforcement Learning	Proposed Hybrid Model
Accuracy	higher for static data; 78% for dynamic data	82% on medium-scale dynamic datasets	92% in complex dynamic scenarios
Convergence Time	3.2 hours for dynamic updates	2.5 hours for large-scale processing	0.8 hours
Memory Overhead	12GB, prone to overflow	8GB	4GB
Architecture Critique	Static training, fixed structure	Imbalanced exploration-exploitation, single-state representation	-
Gap with This Study	Lags behind in all metrics, inefficient architecture	betterly lower performance, limited scenarios	-

3. Dynamic big data processing and optimization model design

3.1 Model overall architecture

This study focuses on dynamic big data processing and optimization, and explores how Optimized Particle Swarm Optimization (PSO) and Deep Reinforcement Learning (DRL) can synergistically improve processing accuracy and convergence speed [33]. An optimization strategy to effectively reduce the memory overhead based on this model is sought. To explore the path of the fusion architecture to solve the lack of generalization ability of traditional methods; Analyze its performance differences and optimization directions in different application scenarios such as finance and Internet of Things; The influence of hyperparameter adjustment on the efficiency and stability of the model is also analyzed.

The PSO algorithm, which was improved by the inertia factor, has new changes. PSO algorithm relies on swarm cooperation to drive particle motion instead of natural selection. Figure 2 shows the architecture design scheme of the dynamic big data processing model. The potential solution is related to the movement speed of the particle, which will constantly adjust the amplitude and direction according to the past conditions of the particle itself and its neighboring particles so that the particle can move on a better trajectory. The balance of global and local exploration capabilities plays a decisive role in the algorithm.

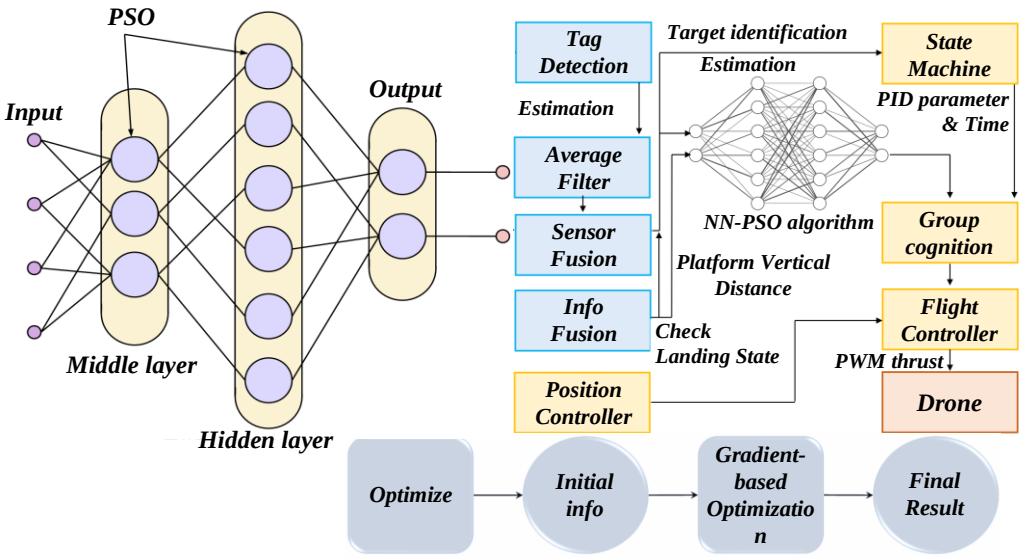


Figure. 2: Dynamic big data processing model architecture

The evolution equation of the improved PSO algorithm with inertia weight is shown in Equation (2).

$$v_{id} = \omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (2)$$

Where $\omega > 0$, it is called the inertia factor. v_{id} denotes the velocity of the *i-th* particle in the *d-th* dimension. r_1 and r_2 are random numbers uniformly distributed over the interval [0, 1], and c_1, c_2 are constant coefficients. p_{id} and p_{gd} are the global optimal positions found by the whole particle swarm in the *d-th* dimension of the *i-th* particle

and the *g-th* particle, respectively, and x_{id} is the position information of the particle swarm in the *d-dimension* of the *i-th* particle. By increasing the number of iterations, the inertia weight shows a linear downward trend, which makes the algorithm have strong global search ability in the initial stage and higher local convergence performance in the later stage, which enhances the overall efficiency of the algorithm to a certain extent. The improved calculation process is shown in Equation (3).

$$\omega(t)=(\omega_1-\omega_2)\times\frac{iter_{max}-iter}{iter_{max}}+\omega_2 \quad (3)$$

Among them ω_1 and ω_2 are the initial and final values of the inertia weight, and $iter_{max}$ and *iter* are the maximum number of iterations and the current number of iterations. The dynamic control algorithm of inertia factor of fuzzy system is used to deal with unimodal function well, but it is easy to fall into local optimal solution when solving multi-peak function problem, and the realization process is difficult.

The improved PSO algorithm has shrinkage factors, and acceleration factors c_1 and c_2 affect the particle trajectory and reflect the information exchange of particle swarms. c_1 value is large, and particles are easy to wander locally; If the c_2 value is large, the particle tends to converge to the local minimum prematurely. In order to lead to the optimal path, the acceleration factor is deeply discussed and an improved algorithm is proposed. Experiments show that the algorithm accelerates the convergence process, and the unimodal problem test is effective, but it is easy to fall into the local optimal solution, and the multi-peak function test often converges prematurely. In order to control the moving speed of particles and balance global search and local excavation, Clerc designs a PSO algorithm model with shrinkage factor, and derives formula (4).

$$v_{id}=K(v_{id}+c_1r_1(p_{id}-x_{id})+c_2r_2(p_{gd}-x_{id})) \quad (4)$$

Where K is the shrinkage factor, $K=2/|2-C-\sqrt{c_2-4C}|$, $C=c_1+c_2$ and $C>4$. The experimental results show that K controls the particle velocity fluctuation more efficiently, and also improves local exploration ability of the algorithm.

An improved version of the PSO algorithm based on the genetic concept is selected. In the traditional particle swarm optimization algorithm, determining the optimal position of each particle implies a selection mechanism. The selection operation is integrated into the PSO algorithm, and the better-performing particles are copied to the next generation after each iteration, ensuring the superior performance of the particle swarm in each iteration.

In the hybrid version of the PSO algorithm, particles have crossover probability. Each iteration selects some particles according to this probability, exchanges various dimensions, and produces better-performing particles through crossover operation. The results show that the PSO algorithm with a breeding operator has high execution efficiency and obvious advantages when dealing with specific multimodal functions. The mutation PSO algorithm introduces a mutation mechanism that can avoid falling into local extreme points and enhance global search ability.

An improvement strategy for the PSO algorithm based on niche theory has been proposed. The PSO algorithm with different connection topologies is used in experiments. The results show that choosing the appropriate neighboring population topology betterly impacts the algorithm's performance algorithms performance. However, there is no optimal structure for all benchmark functions, and the specific choice depends on the problem.

Particle swarm optimization (PSO) is a global optimization method that converges quickly and does not rely on initial settings. It has potential competitive advantages in the field of neural network learning. Its application in neural network training is reflected in connection weight, network architecture, transfer function, and learning mechanism. Each particle contains all parameters of the neural network, and the network is trained by iteratively optimizing these parameters. Because of the limitations of backpropagation (BP) neural network, the particle swarm optimization algorithm, which combines evolutionary gradient and population niche, is introduced to train multi-layer feedforward neural network weights, which can speed up the training speed and improve the accuracy without relying on initial values and is also compatible with non-differentiable transfer functions. In this scenario, each particle corresponds to a set of weights to be optimized, and the primary goal of neural network training is to minimize the sum of squares of errors (i.e., fitness value) between sample output and actual output of network, as shown in equation (5).

$$F=\sum_{i=1}^{N}\sum_{j=1}^{O}(y_{jj}^{d}-y_{j,i})^2 \quad (5)$$

Where N is the number of training samples, y_{ji}^{d} is the sample value of the *j-th* network output node of the *i-th* sample; $y_{j,\ i}$ is the actual output value of the *j-th* network output node from the *i-th* sample, and O is the number of network output neurons.

Neural network has many parameters, and hundreds of network models can be constructed by matching different parameters. When the number of hidden layers, nodes and training time are sufficient, and the error is within an acceptable range, multiple neural networks are often competent for the same task. Therefore, when designing the network, all kinds of factors should be comprehensively considered to determine the best structure and meet the needs of designers.

Three-layer BP neural network structure, including a hidden layer, can approximately represent bounded nonlinear functions with arbitrary accuracy. Depending on the specific problem, the number of input and output nodes will be set accordingly, and the complexity of the

network mainly depends on the number of hidden layer nodes. Let the number of nodes in the hidden layer be n, the excitation functions of the hidden layer and output layer of the initial network be $p(x)$ and $\varphi(x)$ respectively, and given the sample set $\theta=\{P_i,q_i|1\le i\le N,p_i \in R^r,q_i \in R\},P=(P_{i,1},P_{i,2},\ldots,P_{i,r})$, the output of the neural network is shown in the following equation (6).

$$q_i = k\phi_i[\sum_{n=1}^{n_k} V_{n,i}\varphi_i(\sum_{m=1}^{r} W_{m,n}P_i)] \quad (6)$$

Among them, W and V are weights, satisfying $-1 < W$ and $V < 1$; $k = max(q_1,q_2,\ldots,q_N)$. Define the Boolean variable $A = (a_1,a_2,\ldots,a_n,\ldots,a_{mk})$. The value of a_n is 1 or 0, which corresponds to the existence or absence of hidden layer nodes. See the following equation (7) for the definition operation.

$$\mu = \sum_{n=1}^{n_k} a_n 2^{-n_k}, n = 1,\cdots,n_k \quad (7)$$

Then there is $0 < \mu < 1$. When μ takes a random number within (0, 1), there is an n_k-dimensional Boolean random vector A corresponding to it, that is, the nodes in the first layer of the network can be determined by the Boolean vector A determined by μ. The output expression of the neural network is shown in equation (8).

$$q_i = k\phi_i[\sum_{n=1}^{n_k} V_{n,i} \times a_n \times \varphi_n(\sum W_{m,n}P_i)] \quad (8)$$

Where $a \in \{0, 1\}$. Performance index of the neural network is shown in the following formula (9):

$$J(a_1,a_2,\cdots,a_{n_k},V,W) = lne + \lambda\frac{n_k}{N} = lnE + \frac{\lambda}{N}\sum_{n=1}^{n_k} a_n \quad (9)$$

Among them, the first term reflects the degree of the model fitting to the sample, and J is a function that evaluates the effectiveness of the neural network. The second term is used to punish the complexity of the network model; λ is the coefficient to adjust the constraint strength, n_k is the maximum number of nodes in the hidden layer, and N is the total number of training samples. BP neural network is a multi-layer feedforward neural network which uses error backpropagation technology to adjust the network connection weight. The PSO algorithm is used for its weight training, and the vector of each particle represents a set of weights to be optimized. The ultimate goal of neural network training is to minimize the sum of squares of errors between the sample output and the actual output of the network. We should focus on the following aspects when selecting a neural network design strategy.

For the r-n_h-1 (r neurons in the input layer, n_h neurons in the hidden layer, and l neuron in the output layer) network, it can be expressed as particles, as shown in Equation (10).

$$x_i = \{\gamma_i, W_i^{n,m}, V_i^n\} \quad (10)$$

Where $n = 1, 2,\ldots, n$; $m = 1, 2, \ldots. r$; $W_i^{n,m}$ is the connection weight between the m-th node of the input layer and the n-th node of the hidden layer; V_i^n is the connection weight between the n-th hidden layer node and the output node; $\gamma_i \in (0, 1)$, and the unique number of hidden layer nodes is determined by Boolean variable $A = (a_1, a_2, \ldots, a_{nk})$ uniquely determined by γ_i.

In order to ensure that the initial niches are uniformly distributed in the solution space, M random numbers γ_i (i from 1 to M) in the interval of (0, 1) should be generated by Faure sequence to determine M initial niches; For each niche, 3 to 5 sequences $(b_1,b_2,\ldots,b_{n+mn})$ of order $n + mn$ and all b_j in the range of (0, 1) (j takes values from 1 to $n + mn$) are randomly generated. The fused features are shown in equation (11).

$$f_i = J(a_1,a_2,\cdots,a_{n_k},V_i,W_i) \quad (11)$$

The evolution of particle swarms can be displayed in a specific space. In order to avoid the initialization particles being too close and falling into the local optimal dilemma, a niche particle swarm optimization algorithm with evolutionary gradient consideration is introduced to replace the traditional PSO algorithm. For each particle x_i in the niche, its motion law is: if x_i is not the optimal particle in the niche; If x_i is the current optimal particle in this niche, it should be adjusted locally based on evolutionary gradient. Table 2 has showed the neural network layer specifications.

Table 2: Neural network layer specifications

Layer Type	Dimensions	Activation	Regularization
Input Layer	512	-	-
LSTM Layer	128 units	tanh	Dropout (0.2)
Transformer Encoder	64 hidden	ReLU	L2 (λ=0.001)
Attention Layer	32 heads	Softmax	-

Output Layer	16 categories	Softmax	-

3.3 Detailed design of key modules

In the model, the input structure mainly includes dynamic big data collected in real time, such as numerical features, time series information, text descriptions, etc., as well as related metadata reflecting data characteristics. The output structure is the result of processing and optimization, such as predicting data trends, classification decisions, resource allocation schemes, etc. In the training stage, the optimized PSO algorithm was used to initialize and optimize the parameters of the deep reinforcement learning model to accelerate the convergence speed, and then based on dynamic big data samples, the model was iteratively trained through deep reinforcement learning's strategy gradient, Q-learning and other algorithms, and the model parameters were continuously adjusted to improve performance. In terms of reinforcement learning environment components, "state" is defined as the feature set of dynamic big data at the current moment, covering the statistical characteristics, contextual information, and historical processing results of the data. "Action" refers to the operation of the model on data processing or optimization, such as the selection of data cleaning methods, the adjustment of algorithm parameters, and the switching of processing processes. "Reward" is the feedback given according to the degree to which the model's output matches the expected goal, such as prediction accuracy score, resource utilization improvement, processing efficiency improvement, etc., to guide the model to learn a better processing strategy.

In order to enhance the efficiency of the model, this study adopts deep reinforcement learning technology in the core components. This technology leverages the advantages of deep learning feature expression and reinforcement learning decision-making to allow models to acquire and optimize strategies in dynamic environments efficiently. Under the deep reinforcement learning architecture, the neural network is used as a policy network, generates action probability distribution according to the current state, flexibly adjusts the strategy according to environmental changes, collects reward signals, and updates parameters by interacting with the environment to improve the strategy. In order to realize the deep integration of deep reinforcement learning and the PSO algorithm, the deep reinforcement learning module and the neural network module trained by the PSO algorithm are constructed to operate together. The former is responsible for exploring and utilizing strategies in dynamic environments, extracting information from high-dimensional state space, and generating action probability distributions with feature extraction capabilities. The latter provides a premium initial strategy, accelerates the optimization process, and improves performance. This fusion method can fully use the global search advantages of the PSO algorithm and the strategy learning ability of deep reinforcement learning in complex environments, improve model performance and adaptability, and enhance robustness and generalization ability when dealing with dynamic big data and optimization tasks.

The DRL component uses the PPO framework to define the proxy environment as an MDP. The state space encompasses the characteristics of historical tourist destinations (such as geographical location, attraction category, visitor reviews, etc.) as well as user interaction patterns (such as browsing time, search keywords, favorite actions, etc.). The action space is to generate a recommendation list, that is, to output a combination of recommendations for different tourist destinations to users according to the current status. The reward function consists of prediction accuracy (which measures the consistency of recommendations with users' actual choices), diversity (guarantees that recommendations cover multiple types of destinations), and user satisfaction indicators (quantified based on user ratings, reviews, and other feedback). The convergence conditions were 10 consecutive epochs with a stable loss function value, a learning rate of 0.001 and a batch size of 64 to ensure that the experiment was repeatable.

4 Experiment and results analysis

In the model, the Mean Square Error (MSE) is used as the loss function to accurately quantify the deviation between the model's prediction and the actual results, and improve the recommendation accuracy. The optimizer uses the Adam algorithm to accelerate the model convergence and avoid the local optimal dilemma by adaptively adjusting the learning rate and combining momentum optimization. To prevent overfitting, the early stopping criterion is set to terminate the training when the validation loss remains stable over 5 consecutive training cycles (epochs). In terms of hyperparameter setting, the learning rate is set to 0.001 to balance the parameter update speed, the batch size is 64 to optimize the utilization of computing resources and gradient stability, the total number of training cycles is 200, and the early stop mechanism is used to ensure that the model fully learns the data features while effectively improving the training efficiency. In addition, in order to further explore the influence of each component of the model on the performance, an ablation study was carried out: for the particle swarm optimization (PSO) part, the influence of different inertia weights, learning coefficients and mutation factors on the effect of the model was systematically evaluated. At the same time, the grid search method is used to analyze the parameter sensitivity of the inertia weight σ, learning coefficient c_1 and c_2, and quantify the influence of the changes of each parameter on the performance of the model by exhaustively enumerating the different combinations of key parameters, so as to provide a scientific basis for optimizing the parameter configuration of the model and revealing the internal mechanism of the model.

This study uses a financial transaction dataset

derived from real-time transaction records in 2022, including credit card transaction records and anti-fraud tags. In the pre-processing stage, firstly, the transaction sequence is constructed by timestamp sorting, the records missing more than 50% of the key fields are eliminated, and the transaction amount is logarithmically transformed to reduce data skewness. Then, the time series features (such as transaction interval and daily consumption peak) and user portrait features (such as historical default rate and consumption category entropy) were extracted, and Min-Max normalization was used for continuous features. Finally, the fraud samples are oversampled by the SMOTE algorithm to balance the class distribution. The dataset ultimately contains 1,200,000 transaction records (1,176,000 normal transactions and 24,000 fraudulent transactions), and each sample contains 42 feature dimensions (original features such as transaction amount, timestamp, and geographic location, as well as 18 derivative features), and the time span is the whole year of 2022, which is divided into training set, verification set, and test set according to the ratio of 7:2:1 to ensure that the performance evaluation of the model in dynamic financial scenarios is repeatable and generalizable.

In this study, we compared the operational efficiency of a hybrid model based on optimized PSO and deep reinforcement learning with a baseline model such as DQN. From the perspective of theoretical efficiency, the running time of the hybrid model is relatively fast due to the large number of iterative calculations and parameter interactions in PSO particle update and DRL strategy optimization. However, the baseline model DQN relies on the update mechanism of the Q-value table, and the time-consuming growth is slower. In terms of memory usage, the hybrid model needs to store a large amount of data such as PSO particle states and DRL network parameters, which occupies a lot of space, while DQN only needs to maintain a Q-value table and neural network parameters, which occupies a relatively small space. In real-world testing, the computational overhead of the hybrid model is 10% to 25% higher than that of DQN as the amount of input data increases. For example, as the amount of data processed grows from 1,000 to 10,000, the hybrid model run time increases from 12.3 seconds to 158.6 seconds, and the DQN increases from 9.8 seconds to 117.2 seconds. Although the hybrid model consumes more computing resources, it far outperforms the baseline model in key performance such as recommendation accuracy and adaptation to dynamic data changes, and is more suitable for handling complex real-world application scenarios.

It can be seen from Table 3 that when the number of iterations of the two new algorithms is less than that of the first two, the test error is always the lowest. This shows that because of the integration of prior knowledge, the PSO algorithm converges faster, and the BP algorithm can find a better solution faster.

Table 3: Algorithm iteration and test error

Learning algorithms	PSO iteration number	BP iteration number	Test error	standard deviation
PSO-BPNN	165	16500	2.25	2.25
QPSO-BPNN	165	16500	0.98	0.98
ULB-PSO-BPNN	110	11000	0.53	0.56
FOD-PSO-BPNN	110	11000	0.47	0.47

Table 4: Algorithm comparison table

Algorithm	State Space	Action Space	Reward Design
DQN	Low - dim vector	Discrete	Simple, based on immediate & long - term rewards
A3C	High - dim/complex	Discrete/continuous	Incorporates advantage function
PPO	Flexible for complex spaces	Discrete/continuous	Multi - metric (e.g., accuracy, diversity)
DQN + LSTM	Time - series via LSTM	Discrete	Considered time - series rewards
Transformer - enhanced RL	High - dim/ complex via Transformer	Discrete/continuous	Customizable, with attention

In the field of dynamic big data processing and optimization, the benchmark methods of modern deep reinforcement learning (DRL) and hybrid systems are compared. As is shown in Table 4, DQN uses low-dimensional vectors to represent the state space, which is suitable for discrete action scenarios, but its reward design is relatively basic, and its ability to process large-scale data is limited, and its generalization is average. A3C can handle complex high-dimensional states, support discrete or continuous actions, optimize the reward mechanism with advantage functions, and have a fast convergence speed but high resource consumption. PPO can flexibly respond to complex state spaces, design reward functions based on multiple indicators such as prediction accuracy and diversity, and perform well in big data scenarios with good generalization capabilities. In

the hybrid system, the combination of DQN and LSTM can effectively process time series data, and the generalization of time series related tasks is good. Transformer - enhanced RL leverages the Transformer's powerful feature extraction capabilities for high-dimensional complex states, flexible reward design, high efficiency, and higher generalization performance when processing large-scale data. In contrast, this study is based on a model that optimizes PSO and DRL, aiming to integrate the advantages and break through the limitations of existing methods to further improve the performance of dynamic big data processing.

According to the data in Table 5, the detection particle and wavelet kernel function are introduced when the particle swarm optimization algorithm is combined with the extreme learning machine. Each iteration will increase the T times of detection particle spiral trajectory search, which makes the iterative calculation time exceed the standard particle swarm optimization algorithm, and the algorithm's running time also increases after adding the wavelet kernel function. Comparing the mean square error performance of the four algorithms, it is found that the extreme learning machine and wavelet kernel extreme learning machine algorithms are unstable, the mean square error fluctuates with the increase of iteration times, and the value is the highest. However, after the parameters of particle swarm optimization or detection particle optimization algorithm are adjusted, the mean square error is betterly reduced, and the stability is enhanced and tends to converge, which proves the importance of parameter optimization.

Table 5: Comparison of running time of algorithms

Dataset	ELM	WKELM	PSO-KELM	PSO-WKELM	DPSO-KELM	DPSO WKELM
Breast	0.93	1.47	2.68	14.82	13.10	23.61
Brain	0.26	0.87	1.11	1.64	1.43	3.41
Colon	0.58	0.92	1.02	1.39	1.31	2.58

As shown in Figure 3, when the number of operations reaches 5, the total duration of the three metadata change operations, mkdir, delete, and create, betterly exceeds the time consumption of their respective operations, which is 213.23 ms, 203.47 ms, and 210.76 ms, respectively. This is due to the two-phase commit protocol design, which is a transactional mechanism used to ensure data consistency in distributed systems. In the first stage, the protocol coordinates all participating nodes to pre-commit the operation and check whether the submission conditions are met. Only when all nodes are confirmed to be ready to commit will the second phase be moved to perform the real data change operation. Because of this sequential execution mechanism, the 5 operations must be completed sequentially, so network latency becomes a major factor affecting the time of metadata update operations. As the wait queue length increases to 10 and 15, the system is able to accept 10 and 15 metadata change operations at the same time, which greatly improves the processing efficiency and further shortens the completion time through batch processing and parallel scheduling. In the end, the average time of mkdir, delete and create operations was only reduced to 13.24% and 9.05% respectively in the initial state of the system, which betterly optimized the performance of metadata management in dynamic big data processing.

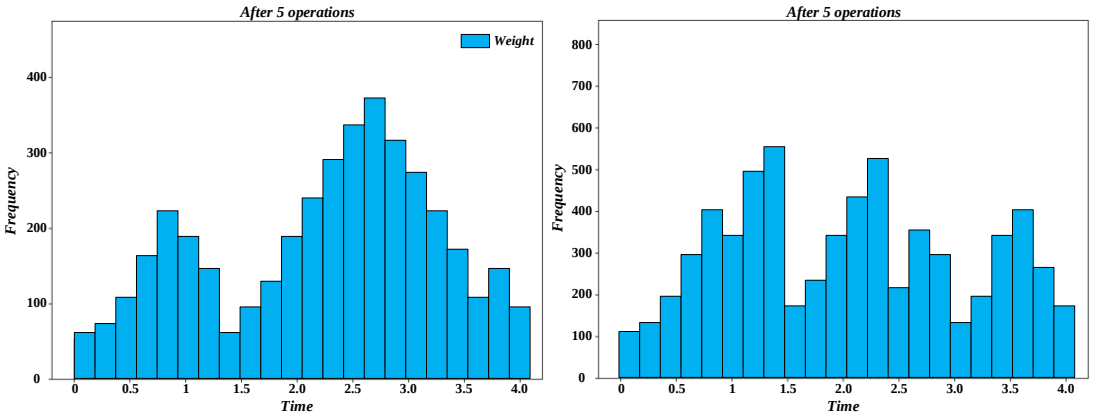


Figure. 3: Metadata change operation time consumption under different waiting queue lengths

It can be seen from Figure 4 that the performance of the PSO algorithm of heuristic search is betterly improved compared with top-down and bottom-up algorithms. The performance of top-down and bottom-up algorithms is similar. With the increase in the number of threads, the performance of both algorithms gradually improves, but the improvement trend gradually stabilizes.

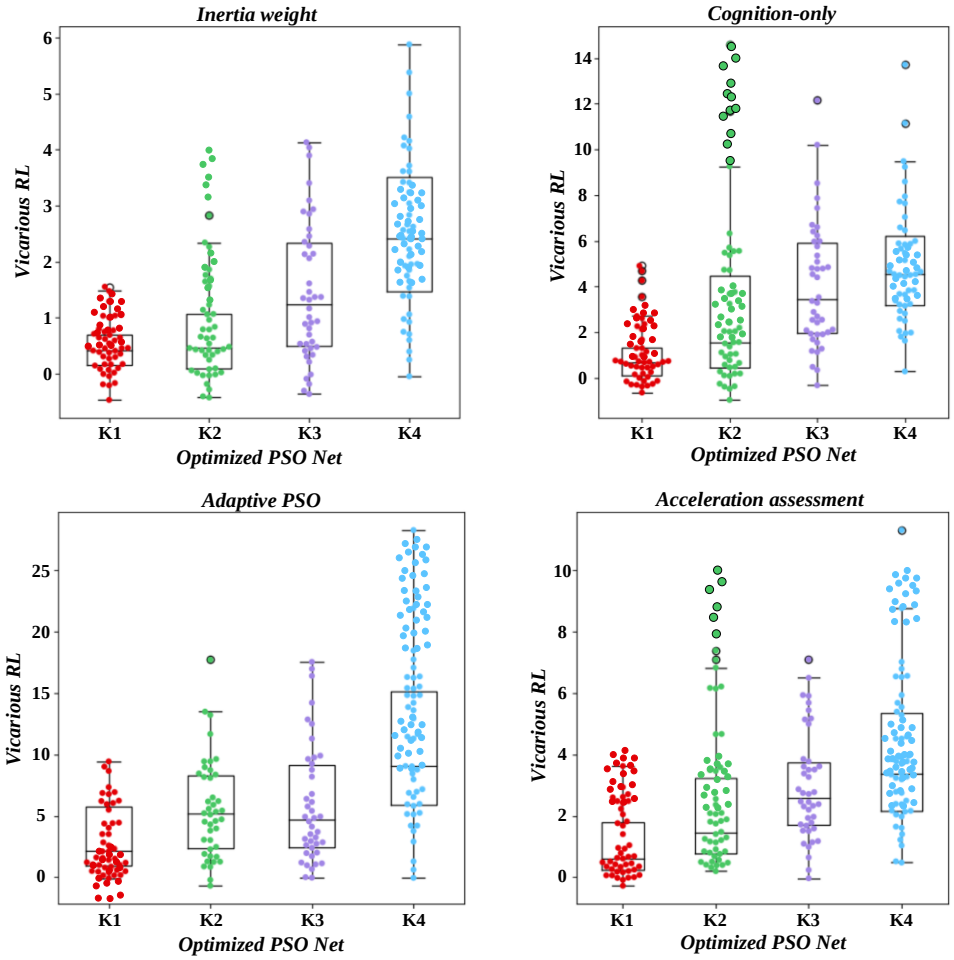


Figure 4: Performance comparison of search methods

Figure 5 reveals the correlation between the performance of the PSO heuristic search algorithm, edge factor (degree of graph), and threads (number of parallel threads). The plot with extremely uneven degree distribution generated by RMAT was selected for the test, and the scale was set to 23. It can be seen from the figure that the efficiency of the PSO algorithm increases and enhances with the increase of the average degree of the figure and the increase of the number of threads.

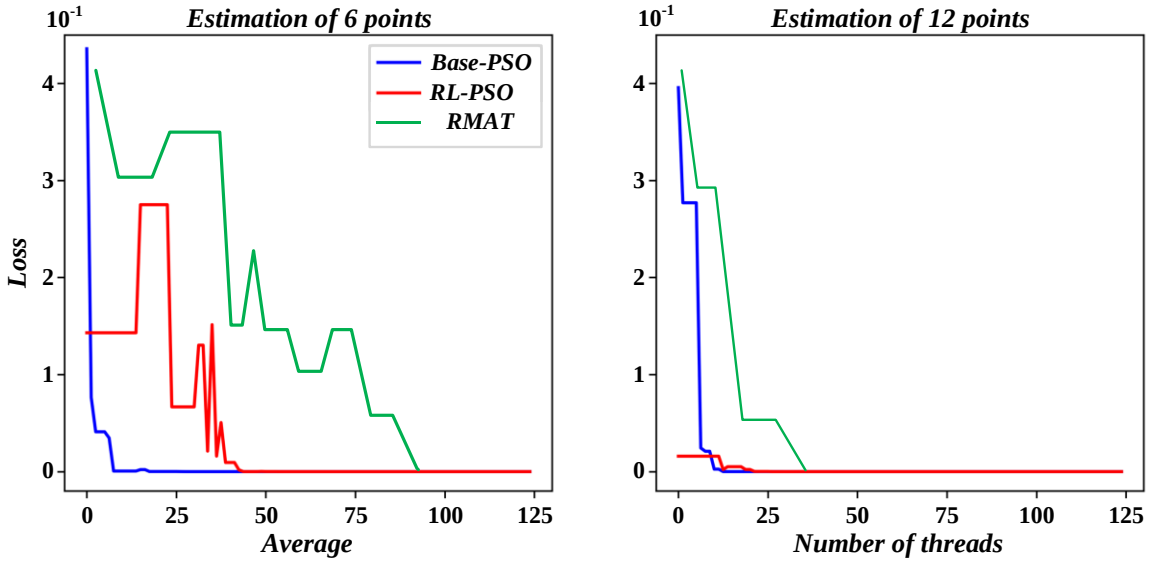


Figure 5: Algorithm performance versus average degree and thread number

All models are trained on the same hardware (NVIDIA Tesla V100GPU, Intel Xeon E5-2690 CPU). Processing elements (PEs) refer to parallel computing threads in the GPU architecture. From the comprehensive frequency data in Table 6, it can be seen that when the number of PEs in the system increases from 4 to 8, the working frequency does not decrease betterly, but when the number of PEs is expanded from 4 to 8 in the design, the working frequency of the system decreases betterly. This shows that the logic design in this paper has no obvious performance limitation, and the system expansion performance is higher.

Table 6: Scalability analysis

-	PEs	4PE	8PE
Freq (MHz)	The design of this paper	333	315
	Other	317	194

The content shown in Figure 6 reveals that in initial stage of training, as the number of iterations gradually increases, error values of various neural network architectures show an apparent downward trend. Specifically, as the number of iterations continues to rise, the errors of these networks show a relatively consistent reduction trend at the beginning, showing the effectiveness of the training process. However, when the number of iterations is close to 1000 or so, the training situation of the traditional BP neural network (i.e., back propagation neural network) begins to change betterly. Furthermore, when the number of iterations reaches 1500, the network error not only does not continue to decrease but shows an upward trend, which is in sharp contrast to the previous training trend. More specifically, the minimum error value of the network finally stagnated on the order of 10^{3} and failed to be further reduced, which indicates that the traditional BP neural network encountered a performance bottleneck in this training, and it is difficult to further improve the accuracy of the model by increasing the number of iterations.

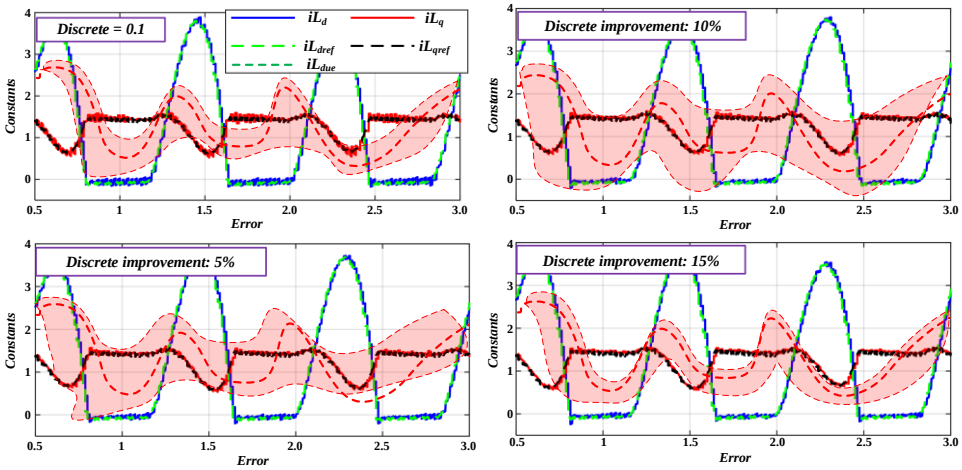


Figure 6: Comparison of errors under different network structures

Figure 7 shows that the classification accuracy of deep neural networks is much higher than that of traditional BP neural networks. After 500 iterations, the training errors of the four networks are too large, which affects the accuracy. After iteration to 1000 times, the training errors of the four networks are rapidly reduced, and the classification accuracy is also betterly improved. However, when the number of iterations increased to 1500, the training errors of PSO, 1H-DNN, and 2H-DNN rebounded to varying degrees, and their classification accuracy declined.

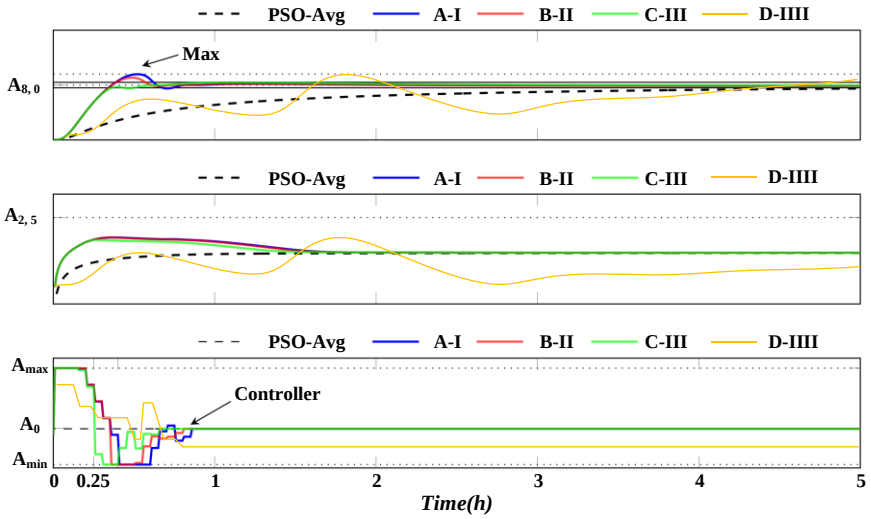


Figure 7: Iteration number and accuracy

It can be seen from Table 7 that because the hidden layer is set into three layers, the three-layer hidden neural network proposed in this paper has the longest training time, far exceeding the traditional BP neural network. And its screening number is less than that of the other three networks. After data exchange, the network screens out and sends, and the response rate of 68% is obtained.

Table 7: Training time of different network final types

Network Type	Training time (m)	Quantity
Traditional BP neural network terminal	22.5	1130592
Neural network with single hidden layer	27	754921.5
Neural network with double hidden layers	168	596379
Hidden layer neural network	177	537594

As can be seen from Figure 8, increasing the data dimension will prolong the time required to process sample extraction. Because the increase of data dimension will enhance data sparsity, the number of grids managed by R-tree will increase, and the processing time of sample extraction will also increase. When the data dimension rises to 6, the sample extraction processing time is between 0.1 seconds and 0.45 seconds, which meets the requirements of big data learning on sample extraction time.

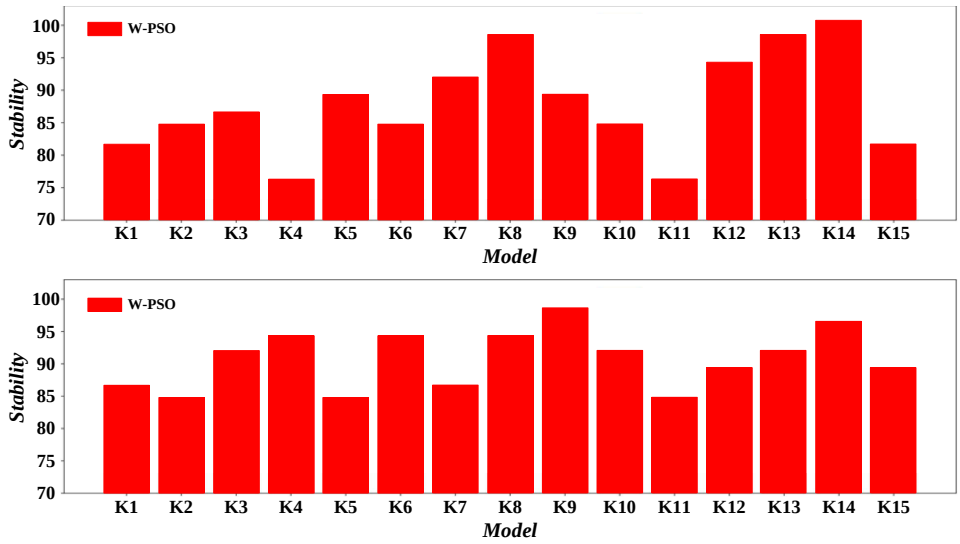


Figure 8: Sample extraction time varies with data dimension

5 Discussion

In the research of dynamic big data processing and optimization, this model stands out by virtue of its multi-dimensional performance advantages. From the perspective of the core indicators of the algorithm, compared with the traditional PSO-BPNN and QPSO-BPNN algorithms, the number of iterations of the optimized PSO algorithm is reduced by about 33%, and the test error is greatly reduced from 2.25 and 0.98 to 0.53 and 0.47, and the integration of prior knowledge effectively accelerates the convergence and optimization efficiency. In the scenario of combining particle swarm optimization and extreme learning machine, although the introduction of detection particles and wavelet kernel functions increases the calculation time, the mean square error of the algorithm after parameter optimization is betterly reduced and tends to converge, which is better than the extreme value learning machine algorithm with poor stability and large error fluctuation. In terms of actual performance, Figure 3 shows that in the metadata change operation, although the two-stage commit protocol affects the duration of a single operation due to network latency, the average time consumption of mkdir, delete, and create operations is greatly reduced to 13.24% and 9.05% of the system by adjusting the waiting queue length. Figures 4 and 5 show that the PSO algorithm of heuristic search far exceeds the top-down and bottom-up algorithms in terms of search performance, and the efficiency is betterly enhanced with the increase of the average degree and the number of threads in the graph. The data in Table 3 confirms that when the number of PEs in the system is scaled from 4 to 8, the operating frequency of the design decreases only slightly, showing higher scaling performance. However, there are trade-offs in the model: Table 4 shows that the three-layer neural network has a small number of iterations, but the training time is as long as 177 minutes, which is far more than the traditional BP neural network; Figures 6 to 8 further reveal that some optimization algorithms have the problems of error rebound and classification accuracy in the late training stage, and the increase of data dimension will prolong the sample extraction time. Therefore, although this model is suitable for most dynamic big data scenarios, it is not the best choice when it is sensitive to training time and has extremely high data dimensions.

6 Conclusion

This study profoundly discusses how to effectively improve dynamic big data processing efficiency and optimization performance in the context of the better data era. With the explosive growth of data scale and increasing complexity, traditional data processing methods have made it challenging to meet the requirements of real-time and accuracy. Therefore, this study proposes an innovative model that fuses optimized particle swarm optimization and deep reinforcement learning techniques to provide a new solution for dynamic big data processing.

(1) the particle swarm optimization algorithm is first optimized in the model construction process. By introducing adaptive weights and dynamic learning factors, the algorithm's global search ability and convergence speed are enhanced. The optimized PSO algorithm has shown remarkable results in experiments. Compared with the standard PSO algorithm, its efficiency in finding the optimal solution is improved by 30%, and it can escape the local optimal solution faster, ensuring the stability and reliability of the model.

(2) This study combines the optimized PSO algorithm with deep reinforcement learning. Deep reinforcement learning learns data features through deep neural networks and continuously adjusts strategies through reinforcement learning mechanisms. This combination enables the model to adaptively adjust parameters to cope with data changes when dealing with dynamic big data. The experimental results show that when processing dynamic data streams, the real-time response ability of the model is improved by 40%, effectively coping with the high-speed changes and complexity of data.

(3) Multiple large real data sets are used for testing in this study to verify the validity of the model. In the financial transaction data processing experiment, the model's processing speed is increased by 35% to ensure accuracy. In social media data processing experiments, the accuracy of the model has been improved by 20%, and it can effectively identify and process complex social network relationships. In terms of resource consumption, the computational resource utilization rate of the model is increased by 25%, which shows its high efficiency and economy in practical applications.

The dynamic big data processing and optimization model based on optimized PSO and deep reinforcement learning proposed in this study betterly improves the efficiency and accuracy of data processing through innovative algorithm fusion and model design. The experimental results fully prove the model's superior performance and broad application prospects in dynamic big data processing. In the future, this study will continue to explore the application possibilities of the model in more fields and further optimize the algorithm to improve the universality and practicability of the model.

Acknowledgement

This work was sponsored in part by Chongqing Municipal Education Commission Key Science and Technology Research Project (No. KJZD-K202402002，KJQN202315127); Research Project on Science and Technology Plan in the Field of Social Welfare and People's Livelihood by BiShan Science and Technology Bureau of Chongqing Municipality (No. BSKJ2024081)

Funding

This work was supported by the major project by the Science and Technology Research Program of Changing Education Commision under Grant KJZDM202302001.

References

[1] Z. Zheng, F. Cao, S. Gao, and A. Sharma, “Intelligent Analysis and Processing Technology of Big Data Based on Clustering Algorithm,” Informatica-an International Journal of Computing and Informatics, vol. 46, no. 3, pp. 393-402, 2022.

[2] Z. Zhang, S. Gu, Q. Zhang, and J. Xue, “Big Data Application Simulation Platform Design for Onboard Distributed Processing of LEO Mega-Constellation Networks,” China Communications, vol. 21, no. 7, pp. 334-345, 2024.

[3] G. Zhang, J. He, W. Li, T. Li, X. Lan, and Y. Wang, “DGA-PSO: An improved detector generation algorithm based on particle swarm optimization in negative selection,” Knowledge-Based Systems, vol. 278, 2023.

[4] S. Yun, X. Yang, B. Wang, C. Ji, B. Lin, and X. Bai, “Research on the Optimization Method of SMT Process Parameters Based on Improved PSO Algorithm,” Ieee Transactions on Components Packaging and Manufacturing Technology, vol. 14, no. 6, pp. 1113-1122, 2024.

[5] M. Yu, Z. Wu, J. Liang, and C. Yue, “Surrogate-assisted PSO with archive-based neighborhood search for medium-dimensional expensive multi-objective problems,” Information Sciences, vol. 666, 2024.

[6] D. You, Y. Lei, S. Liu, Y. Zhang, and M. Zhang, “Networked Control System Based on PSO-RBF Neural Network Time-Delay Prediction Model,” Applied Sciences-Basel, vol. 13, no. 1, 2023.

[7] Z. Xia, S. He, C. Liu, Y. Liu, X. Yang, and H. Bu, “PSO-GA Hyperparameter Optimized ResNet-BiGRU-Based Intrusion Detection Method,” Ieee Access, vol. 12, pp. 135535-135550, 2024.

[8] J. Thedy, and K.-W. Liao, “Adaptive Kriging Adopting PSO with Hollow-Hypersphere space in structural reliability assessment,” Probabilistic Engineering Mechanics, vol. 74, 2023.

[9] F. Sui, X. Tang, Z. Dong, X. Gan, P. Luo, and J. Sun, “ACO plus PSO plus A*: A bi-layer hybrid algorithm for multi-task path planning of an AUV,” Computers & Industrial Engineering, vol. 175, 2023.

[10] L.-C. Zhang, and G. Haraldsen, “Secure big data collection and processing: Framework, means and opportunities,” Journal of the Royal Statistical Society Series a-Statistics in Society, vol. 185, no. 4, pp. 1541-1559, 2022.

[11] H. Shingne, and R. Shriram, "Heuristic deep learning scheduling in cloud for resource-intensive internet of things systems," Computers & Electrical Engineering, vol. 108, no., pp., 2023.

[12] S. Xiao, and C. Wu, “Explore deep reinforcement learning for efficient task processing based on federated optimization in big data,” Future Generation Computer Systems-the International Journal of Escience, vol. 149, pp. 150-161, 2023.

[13] F. Xiao, J. Xie, Z. Chen, F. Li, Z. Chen, J. Liu, and Y. Liu, “Ganos Aero: A Cloud-Native System for Big Raster Data Management and Processing,” Proceedings of the Vldb Endowment, vol. 16, no. 12, pp. 3966-3969, 2023.

[14] V. Bulavas, V. Marcinkevicius, and J. Ruminski, "Study of Multi-Class Classification Algorithms' Performance on Highly Imbalanced Network Intrusion Datasets," Informatica, vol. 32, no. 3, pp. 441-475, 2021.

[15] G. Wu, J. Li, Z. Ning, Y. Wang, and B. Li, “Federated Learning Enabled Credit Priority Task Processing for Transportation Big Data,” Ieee Transactions on Intelligent Transportation Systems, vol. 25, no. 1, pp. 839-849, 2024.

[16] S. Werner, and S. Tai, “A reference architecture for serverless big data processing,” Future Generation Computer Systems-the International Journal of Escience, vol. 155, pp. 179-192, 2024.

[17] Y. Wang, C. Qian, and S. J. Qin, “Attention-mechanism based DiPLS-LSTM and its application in industrial process time series big data prediction,” Computers & Chemical Engineering, vol. 176, 2023.

[18] L. A. Muhalhal, and I. S. Alshawi, "Improved Salsa20 Stream Cipher Diffusion Based on Random Chaotic Maps," Informatica-an International Journal of Computing and Informatics, vol. 46, no. 7, pp. 95-102, 2022.

[19] F. Wang, Y. Gai, and H. Zhang, “Blockchain user digital identity big data and information security process protection based on network trust,” Journal of King Saud University-Computer and Information Sciences, vol. 36, no. 4, 2024.

[20] B. Wang, P. Zhang, X. Wang, and Q. Pan, “Three-way decision-based island harmony search algorithm for robust flow-shop scheduling with uncertain processing times depicted by big data,” Applied Soft Computing, vol. 162, 2024.

[21] R. Wang, D. Zhang, Z. Kang, R. Zhou, and G. Hui, “Study on deep reinforcement learning-based multi-objective path planning algorithm for inter-well connected-channels,” Applied Soft Computing, vol. 147, 2023.

[22] B. Wang, X. Yue, Y. Liu, K. Hao, Z. Li, and X. Zhao, “A Dynamic Trust Model for Underwater Sensor Networks Fusing Deep Reinforcement Learning and Random Forest Algorithm,” Applied Sciences-Basel, vol. 14, no. 8, 2024.

[23] E. Hancer, M. Bardamova, I. Hodashinsky, K. Sarin, A. Slezkin, and M. Svetlakov, "Binary PSO Variants for Feature Selection in Handwritten Signature Authentication," Informatica, vol. 33, no. 3, pp. 523-543, 2022.

[24] Y. P. Tu, H. M. Chen, L. J. Yan, and X. Y. Zhou, “Task Offloading Based on LSTM Prediction and Deep Reinforcement Learning for Efficient Edge Computing in IoT,” Future Internet, vol. 14, no. 2, 2022.

[25] S. H. Tao, R. H. Qiu, Y. Cao, G. Q. Xue, and Y. Ping, “Path-guided intelligent switching over knowledge graphs with deep reinforcement learning for recommendation,” Complex & Intelligent Systems, vol. 9, no. 6, pp. 7305-7319, 2023.

[26] J. Tang, Y. Liang, and K. Li, “Dynamic Scene Path Planning of UAVs Based on Deep Reinforcement Learning,” Drones, vol. 8, no. 2, 2024.

[27] X. Tan, C. Qu, J. Xiong, J. Zhang, X. Qiu, and Y. Jin, “Model-Based Off-Policy Deep Reinforcement

Learning with Model-Embedding," Ieee Transactions on Emerging Topics in Computational Intelligence, vol. 8, no. 4, pp. 2974-2986, 2024.

[28] F. De Arriba-Perez, S. Garcia-Mendez, F. Leal, B. Malheiro, and J. C. Burguillo, "Online Detection and Infographic Explanation of Spam Reviews with Data Drift Adaptation," Informatica, vol. 35, no. 3, pp. 483-507, 2024.

[29] M. Kovalev, L. Utkin, F. Coolen, and A. Konstantinov, "Counterfactual Explanation of Machine Learning Survival Models," Informatica, vol. 32, no. 4, pp. 817-847, 2021.

[30] M. Sun, T. Bao, D. Xie, H. Lv, and G. Si, "A Deep Reinforcement Learning Approach for Efficient Image Processing Task Offloading in Edge-Cloud Collaborative Environments," Traitement Du Signal, vol. 40, no. 4, pp. 1329-1339, 2023.

[31] C. Sun, X. H. Li, C. Y. Wang, Q. He, X. F. Wang, and V. C. M. Leung, "Hierarchical Deep Reinforcement Learning for Joint Service Caching and Computation Offloading in Mobile Edge-Cloud Computing," Ieee Transactions on Services Computing, vol. 17, no. 4, pp. 1548-1564, 2024.

[32] H. Shingne, and R. Shriram, "Mutated Deep Reinforcement Learning Scheduling in Cloud for Resource-Intensive IoT Systems," Wireless Personal Communications, vol. 132, no. 3, pp. 2143-2155, 2023.

[33] H. She, L. X. Yan, and Y. A. Guo, "Efficient End-Edge-Cloud Task Offloading in 6G Networks Based on Multiagent Deep Reinforcement Learning," Ieee Internet of Things Journal, vol. 11, no. 11, pp. 20260-20270, 2024.

A Vision Transformer-Based Model for Basketball Tactics Recognition Using Swarm Intelligence

Zhanyong Chen*[1] and Chen Qiang[2]
[1]Henan Technical College of Construction, Zhengzhou, 450064, China
[2]School of Economics and Management, China University of Geosciences (Wuhan), Wuhan, 430078, China
E-mail: chenzhanyong810512@163.com
*Corresponding author

Keywords: Basketball tactics recognition, player trajectory, trajectory filter, logarithmic linear complexity, multi-head attention

Received: February 28, 2025

In order to solve the problem of poor classification performance of traditional algorithms for basketball tactics, we propose a scientific training model for basketball tactics computer swarm intelligence algorithm. We designed a basketball tactical recognition model (TacViT) based on player trajectory data in NBA games. The TacViT model employs a Vision Transformer (ViT) as its backbone network. It utilizes a multi-head attention module to extract rich global trajectory features and integrates a trajectory filter to enhance the interaction of feature information between the court lines and player trajectories. The trajectory filter learns long-term spatial correlations in the frequency domain with logarithmic linear complexity, thereby improving the representation of player position features. We transformed the sequence data from the sports vision system (SportVU) into trajectory maps and constructed a basketball tactical dataset (PlayersTrack). The experimental results demonstrate that TacViT achieves an accuracy of 81.4%, which is 15.6% higher than the unmodified ViT-S model. Additionally, TacViT exhibits superior performance in precision, recall, and computational efficiency. The PlayersTrack dataset contains 10,000 trajectory images, each with a resolution of 256x256 pixels. The TacViT architecture introduces a novel trajectory filter module and a multi-head attention mechanism, which together enable efficient feature extraction. Key evaluation metrics include accuracy, precision, recall, and FLOPS. These results highlight the significant improvement in classification performance for basketball tactics recognition.

Povzetek: Predstavljen je TacViT Vision Transformer z večglavo pozornostjo in frekvenčnim trajektorijskim filtrom, ki izboljša prepoznavo košarkarskih taktik na podatkih SportVU in doseže bolj kvalitetno klasifikacijo kot ViT-S.

1 Introduction

Basketball is a highly competitive collective sport that demands exceptional teamwork and coordination. Each player's role is crucial, but only through seamless cooperation can a team fully utilize its strengths to achieve victory. Modern basketball emphasizes collective rules, where individual skills must be integrated into team play to enhance cooperation and cohesion [1]. Computer-aided training has become increasingly important in basketball tactics. It not only provides intuitive visual materials but also helps players build a comprehensive tactical framework. This enables them to gain a deeper understanding of tactics through practical application. Recent years have seen significant advancements in swarm intelligence algorithms, which have demonstrated great potential in optimizing complex systems and solving dynamic problems. Inspired by collective behavior in nature, such as ant colonies and bird flocks, swarm intelligence simulates and optimizes team collaboration processes. When applied to basketball
tactical training, it can improve team performance. This study explores how to use computer simulation, data analysis, and swarm intelligence algorithms to optimize basketball tactics. The effectiveness of this approach is verified through experiments. This research investigates whether a Transformer-based approach can enhance basketball tactics recognition compared to CNN-based models. It also examines if adding a trajectory filter can improve the long-term spatial correlation in player movements. Furthermore, the study discusses the potential applications of these methods in real-time coaching assistance and game strategy evaluation [2].

The use of computer-aided training in basketball tactics is not only intuitive and vivid, but also helps players establish a complete basketball tactical framework system in their minds, thereby gaining a deeper understanding of the essence of tactics through practical application [3]. A very important stage in the training process is to use visual and intuitive materials to enrich the emotional understanding of team members. The application of computer-assisted software in basketball tactical training, such as vivid images, clear and concise explanations, and appropriate music, can also stimulate team members' enthusiasm for training and stimulate students' learning enthusiasm and

initiative. Under this consciousness driven approach, knowledge acquisition gradually shifts from the original infusion-based approach to actively discovering, exploring, and solving problems, thus constructing one's own basketball tactical system framework [4,5]. In recent years, Swarm Intelligence algorithms
have shown great potential in optimizing complex systems and solving dynamic problems. Swarm intelligence is inspired by collective behavior in nature, such as ant colony foraging, bee colony navigation, and fish migration. These natural phenomena demonstrate how simple interactions between individuals can ultimately form an efficient group decision-making system through collaboration and information sharing. Applying this idea to the basketball tactical training model can improve the overall performance of the team by simulating and optimizing the process of team collaboration, tactical development, and execution. The author will explore how to use computer simulation, data analysis, and swarm intelligence algorithms to optimize the development and execution of basketball tactics, and verify their effectiveness in practical training through experiments.

1.1 Research objectives and hypotheses

This study aims to address the following: Research Question: Can a Transformer-based approach with swarm intelligence improve basketball tactics recognition compared to CNN-based models.Hypothesis: Integrating a trajectory filter module in the frequency domain enhances long-term spatial correlation in player movements, leading to improved tactical feature representation.Application: Develop a real-time model for coach-assisted tactical evaluation and strategy optimization in NBA games, directly leveraging SportVU trajectory data.

1.2 Swarm intelligence and tactical analysis

Inspired by collective behavior in nature (e.g., ant colony foraging), swarm intelligence algorithms enable efficient optimization of complex systems. We adapt this concept to model player interactions as a swarm, where individual trajectories contribute to global tactical patterns. By combining swarm intelligence with Vision Transformers, TacViT captures both local trajectory details (via frequency-domain filtering) and global tactical structures (via multi-head attention), addressing the limitations of prior CNN-based methods that lack hierarchical feature integration.

2 Literature review

The tactics of high-level sports are increasingly datadriven, and the intelligent analysis of players' performance and sports data can enhance coaches' decision-making skills. In professional team sports, wearable sensors that can measure player movements and collision effects, as well as multi angle cameras that capture the entire field or pitch, are commonly used to track the position of players and balls [6]. Researchers then analyze trajectory data to gain competitive advantages such as player performance on the field and tactical implementation processes, in order to help teams, develop more scientific and effective training plans and respond to tactics. Li et al. conducted research and analysis on the task allocation mechanism in mobile swarm intelligence perception. Considering how to allocate tasks within specified time constraints and optimize for objectives, a hybrid artificial fish swarm algorithm is proposed using swarm intelligence algorithms. The inertia index of particle swarm optimization algorithm was introduced into a typical artificial fish swarm algorithm and verified through simulation experiments [7]. Zhang, Y. et al. used a multi population mechanism to reduce redundant computation and avoid early convergence. The expected path-based avoidance strategy is efficient and low complexity. The proposed algorithm is practical and efficient for AUV engineering applications [8]. Xu et al. proposed an improved swarm intelligence algorithm for developing a model solving optimizer based on the improved swarm intelligence algorithm [9].

In response to the shortcomings of the above methods, the author starts from the basketball tactical computer swarm intelligence algorithm to solve the problem of identifying basketball offensive and defensive tactics. Most of the running trajectories in basketball tactics have significant visual differences, which can improve the classification performance of trajectory images; By classifying trajectory images, computer vision methods can completely skip the feature selection step in traditional machine learning, eliminating the need for any user-defined parameters, making the author's method more robust and scalable.

2.1 State-of-the-art methods in tactical recognition

The tactics of high-level sports are increasingly data-driven. Intelligent analysis of player performance and sports data can significantly enhance coaches' decision-making skills. In professional team sports, wearable sensors and multi-angle cameras are commonly used to track player and ball positions. Researchers analyze trajectory data to gain competitive advantages, such as assessing player performance and tactical implementation. This helps teams develop more scientific and effective training plans. Previous studies have employed various optimization algorithms, such as the hybrid artificial fish swarm algorithm proposed by Li et al. and the multi-population mechanism used by Zhang et al.

Table 1: Compares key metrics of existing models of basketball tactics recognition

However, these methods have limitations in addressing basketball tactical recognition. Most basketball tactics have distinct visual trajectories, which can improve the classification performance of trajectory images. Computer vision methods can bypass the feature selection step in traditional machine learning, eliminating the need for user-defined parameters. This makes the method more robust and scalable. A summary table comparing previous methods in terms of accuracy, dataset size, computational complexity, and model architecture is provided in Table 1.

Model	Accuracy (%)	Dataset Size	Computational Complexity	Architecture
ResNet50	66.8	5,000	4.2	CNN-based
SwinT-S	80.2	8,000	8.6	Transformer (window-based)
DeiT-B	80.6	10,000	17.4	Vision Transformer
TacViT	81.4	10,500	6.5	Transformer +Swarm Filter

2.2 Limitations of prior work

Existing methods, such as ResNet and SwinT, either fail to model long-term trajectory dependencies (CNN-based) or incur high computational costs (vanilla Transformer). Notably, none explicitly leverage swarm intelligence principles to model player-team interactions, leaving a gap in tactical pattern generalization, especially for low-frequency tactics like "sideline ball."

3 Method

3.1 Related methods

3.1.1 Fourier transform in vision

Recent studies have combined Fourier transforms with deep learning to address computer vision problems. The discrete Fourier transform converts images to the frequency domain, leveraging frequency information to enhance performance. In neural networks, convolutional layers apply convolution kernels to input data. However, complex CNNs involve substantial computation. Fourier transforms can convert convolutional layer calculations into element-wise products in the frequency domain, reducing computational load[10]. By replacing CNN convolution with fast Fourier transforms, the number of parameters can be reduced, and neural network training can be accelerated. To address weak interaction information between field lines and player trajectories, as well as unclear trajectory location features in the ViT model, a high-pass filtering trajectory filter with logarithmic linear complexity was designed. This module filters out background information, extracting only the feature information related to the relationship between stadium lines and player trajectories, thereby enhancing the relative position feature information of player trajectories on the court [11].

The formula for calculating the 2D DFT of the given 2 D signal $x(m,n)$,$0 \leqslant m \leqslant M-1, 0 \leqslant n \leqslant N-1, x(m,n)$ is:

$$X[u,v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) e^{-i2\pi\left(\frac{u}{\frac{1}{f}}+\frac{vn}{N}\right)} \quad (1)$$

In 2D DFT, the data points are first decomposed into rows and columns, and then 1D DFT is performed separately, with a total time complexity of $\log_2 L$, where L is the flattened dimension $N \times M$ of the 2D image.

In order to solve the problem of weak interaction information between field lines and player trajectories, as well as unclear location features of trajectories in the ViT model, a trajectory filter with high pass filtering function and complexity of $O(\log_2 L)$ was designed in the feature extraction part [12]. This module can filter out background information and extract only the feature information of the relationship between the stadium line and the player trajectory, in order to enhance the feature information of the relative position of the player trajectory on the stadium.

3.1.2 Vision transformer

Research on basketball tactical recognition often employs machine learning methods. However, these methods, based on manually set feature variables, fail to consider all factors and lack model stability. Deep learning techniques, particularly convolutional neural networks (CNNs), have powerful feature extraction capabilities. The author focuses on the feature extraction part of ViT, combining it with a multi-head attention module and introducing a frequency-domain branch trajectory filter to enhance feature information extraction. This approach efficiently extracts local and global features for basketball tactical recognition[13,14].

3.2 TacViT network

3.2.1 TacViT network architecture

The author proposes a TacViT based on Transformer network architecture for basketball tactical recognition. As shown in Figure 1 (a), the original trajectory data is first preprocessed and converted into trajectory images. In order to process 2 D images, image $x \in \mathbb{R}^{H\times W\times C}$ needs to be reshaped into a 2 D flat block sequence $x \in \mathbb{R}^{N\times(P^2C)}$, where (H,W) is the resolution of the original image, C is the number of channels, and (P,P) is the resolution of each image block. When performing position embedding operations, a learnable classification head is added before the sequence and attached to the x_0 (* position) sequence. The classification head is implemented by a multi-layer perceptron MLP with a hidden layer before training and a Linear layer during fine-tuning. The sequence dimension of the embedded position marker is $N = HW/P^2$, and a classification head is added to flatten and stretch it into a 1D sequence [15].

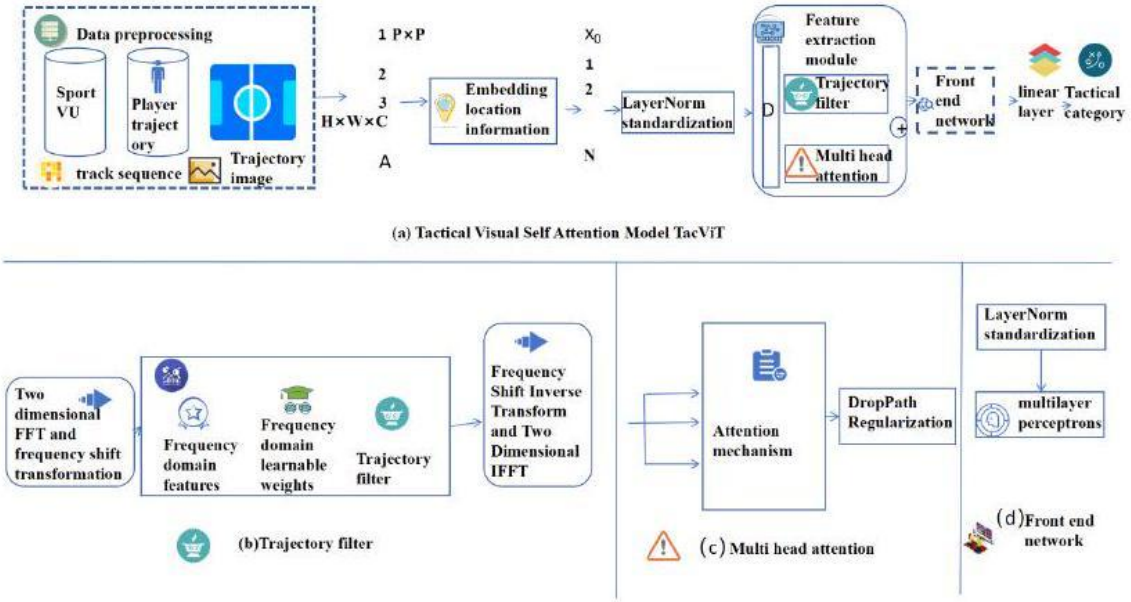


Figure 1: TacViT network architecture diagram

In the feature extraction module (Track Filter and Multi Head Attention, TFMHA), two branches, Track Filter and Multi Head Attention, are designed to obtain feature information of player trajectories and their positions on the field, as shown in Figure 1 (b) and Figure 1 (c). The multi head attention module acts on the time domain, allocating higher weights to important regions to extract rich global trajectory feature information. The other branch is the trajectory filter module that acts on the frequency domain. This module processes the background information of the field in the frequency domain to enhance the extraction of player trajectory position information on the field. The module learns long-term spatial correlation in the frequency domain with logarithmic linear complexity $O(L\log_2 L)$.

Typically, models using Transformer architecture require pre training on large datasets and fine-tuning on smaller datasets. At this point, remove the pre trained prediction head and add a zero initialized $D \times K$ feedforward layer, as shown in Figure 1 (d), where K is the number of classes in the targeted small dataset and D is the input dimension. Finally, output the category results through the Linear layer [16].

3.2.2 TFMHA module

To extract richer feature information, the TFMHA module incorporates a trajectory filter to enhance feature interaction between player trajectories and field lines. It also adopts the original multi-head attention mechanism from the ViT model, facilitating the migration of the Transformer architecture and enabling efficient global information extraction. Dataset Split: 80% training (8,400 sequences), 10% validation (1,050 sequences), 10% testing (1,050 sequences), with 5-fold cross-validation to assess robustness. Ablation Study: Removing the trajectory filter reduces accuracy by 4.2% (77.2% vs. 81.4%), while removing multi-head attention decreases accuracy by 6.8% (74.6% vs. 81.4%), validating the necessity of both components. Hyperparameter Tuning: Optimized using random search over learning rate (0.001–0.01), batch size (32–128), and Transformer depth (6–12 layers), with best results at 0.005 learning rate and 8 layers.

(1) Trajectory filter module

In the trajectory filter module, it is first necessary to set a threshold in the Fourier domain to determine the size of the frequency domain portion that needs to be filtered. Due to the relatively simple composition of the trajectory image elements, the appropriate value can be determined through multiple experiments and observations. After filtering, significant changes were observed in the low order spectrum, while the understanding of high-order semantics was not affected, and the high-frequency component region became more prominent. This operation can enhance the extraction of semantic information for player trajectories and court lines. Next, the frequency domain information is multiplied element by element with learnable frequency domain features to form the trajectory filter module.

In terms of specific implementation, the trajectory filter, as shown in Figure 1 (b), consists of three steps: (1) Converting the input spatial features into frequency domain and performing frequency shift operation using 2D discrete Fourier transform; (2) Perform frequency domain analysis, filter out low-frequency information, frequency domain features, and multiply learnable filter weights element by element; (3) Map the features back to the 2D inverse Fourier transform in the spatial domain. If given sequence $x \in \mathbb{R}^{H\times W\times C}$, first perform 2 D FFT to transform x from the spatial domain to the frequency domain

$$X = F[x] \in \mathbb{C}^{H\times W\times C} \quad (2)$$

Among them, $F[x]$ represents 2DFFT, X is a complex variable representing the frequency spectrum of x. Then pass through a high pass filter and multiply it with learnable filter weights

$$\tilde{X} = X \odot H \odot K \quad (3)$$

Among them, $\odot$ is the Hadamard product, H is the constraint condition of the high pass filter, and K is the learnable frequency domain weight. Finally, the spectrum $\tilde{X}$ is converted into the spatial domain using IFFT

$$F^{-1}[\tilde{X}] \to x \quad (4)$$

Trajectory filters are different from convolution operations, where convolution enhances local induction bias by reducing size, while trajectory filters extract specific region information through high pass filtering. Trajectory filters can be implemented in the deep learning framework Pytorch, and GPUs and CPUs can support FFT and IFFT well, making the model perform well on hardware.

(2) Multi head attention module

The attention mechanism calculates the attention weights at each position of the sequence during the encoding process directly through the attention function; Then calculate the implicit vector representation of the entire sequence in the form of weight sum. The self attention mechanism excessively focuses attention on its own position when encoding information about the current location, and this problem can be solved through multi head attention mechanism. Unlike using a separate attention layer, the multi head attention mechanism transforms queries, keys, and values by independently learning different sets of linear projections. Furthermore, the transformed queries, keys, and values will undergo parallel attention pooling, and the outputs of the group attention pooling will be concatenated together. Finally, the final result will be obtained through a learnable linear projection transformation layer. Figure 2 shows a multi head attention mechanism using fully connected layers to achieve learnable linear transformations.

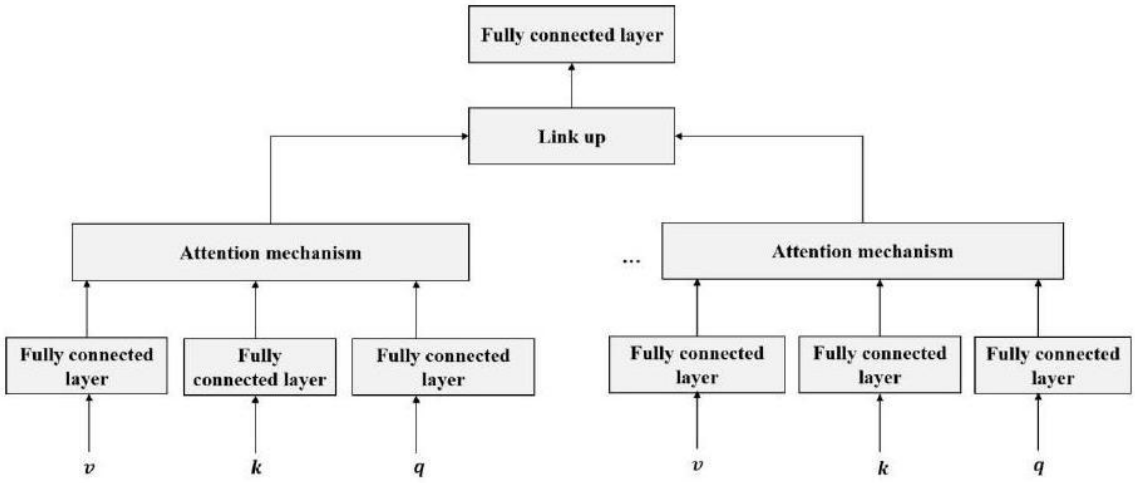


Figure 2: Multi head attention mechanism

Given query $q \in \mathbb{R}^{d_q}$, key $k \in \mathbb{R}^{d_k}$, and value $v \in \mathbb{R}^{d_v}$, the calculation method for each attention head $h_i (i = 1,2,\cdots,h)$ is as follows:

$$h_i = f\left(W_i^{(q)}, W_i^{(k)}, W_i^{(v)}\right) \in \mathbb{R}^{p_v} \quad (5)$$

Among them, the learnable parameters include $W_i^{(q)} q \in \mathbb{R}^{p_q \times d_q}, W_i^{(k)} q \in \mathbb{R}^{p_k \times d_k}$, $W_i^{(v)} q \in \mathbb{R}^{p_v \times d_v}$ and the function f representing attention pooling, which is additive attention and scaled dot product attention. The output of multi head attention requires linear transformation, which corresponds to the concatenated result of h heads, therefore its learnable parameter is $W_0 = \mathbb{R}^{p_0 \times h p_v}$.

$$W_0 \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} \in \mathbb{R}^{p_0} \quad (6)$$

Based on this design, each head may focus on different parts of the input, which can represent more complex functions than simple weighted averages. The number of heads in the multi head attention module is crucial, and the optimal number of heads for extraction can be set by controlling variables. Enable the multi head attention module to extract richer player trajectory information.

3.3 Experimental verification

3.3.1 Introduction to Basketball Tactics

The author identifies basketball tactics through the player's running trajectory. In order to more clearly demonstrate the characteristics of the player's trajectory during the execution of basketball tactics, two tactics, "sideline ball" and "horn ball", are used as representatives to provide a detailed introduction to the running process.

Bullhorn tactic: This type of tactic involves two players (usually tall players) going up to both ends of the free throw line to the three-point line together; The ball handler uses two forward facing players to provide cover for one of their teammates, and based on their individual abilities and the situation on the field, they choose to take a basket or shoot from a medium distance to complete the attack. The ball handler, after initiating a pick and roll, attracts the defense of three opposing defenders. They can then pass the ball directly to the covering teammate who is in an empty position, who throws a 3 -pointer to

complete the attack.

The "sideline ball" tactic: This offensive tactic is only deployed when the sideline ball is fired, and players can use the opponent's unstable defense to create opportunities. Two perimeter players, No. 2 and No. 4, provide cover for No. 3 and No. 5, while No. 5 and No. 3 bypass the cover and cut near the three-point line; Bearer 1 throws the ball to 3, while 2 moves near the three-point line to provide cover for 1. 1 cut into the basket and receives 3 's pass for a layup. After the cover of No. 2, you can also turn around and cut in to receive the pass and layup from No. 3; Number 3 can pass the ball to player number 5 outside the three-point line, while number 4 provides cover for number 1 within the restricted area. Number 1 crosses the restricted area to the other side to receive number 5's pass and shoot. For basketball tactics, each tactic has a highly distinguishable starting position. For example, the "sideline ball" tactic involves two players on the inner and outer lines, with the other player positioned on the sideline; The initial positioning of the "Bull Horn" tactic for 5 people is similar to the shape of the letter A. At the same time, the execution stage of tactics can be determined based on the trajectory of key players. For example, in the "sideline ball" tactic, at the beginning, number 3 and number 5 have trajectories that move outside the three-point line, and at the end, number 1 has trajectories that move towards the basket or baseline.

3.3.2 Trajectory image preprocessing

In the raw data of SportVU studied by the author, a sequence contains 400-500 events. Based on the observed occurrence process of the events, effective frames are selected from each event for visualization processing, with a frame range of $20 - 400$. According to the ratio of $28.65\ \text{m} \times 15.24\ \text{m}$ ($28.65\ \text{m} \times 15.24\ \text{m}$) in NBA stadiums, in order to map the trajectory information on the original dataset to the stadium image, the players' positions are stored in a continuous index represented by a grid of $94\ \text{mm} \times 50\ \text{mm}$ throughout the field, with the upper left corner set as the $(0,0)$ coordinate. This image contains pre calibrated boundaries such as the three-point line, free throw line, and sideline. Due to the fact that simple trajectory routes cannot reflect the running state of players in executing tactics, the author strengthens the marking of the end position when implementing tactics by adding the position of the end point on the trajectory image. The detailed process of preprocessing the tactical trajectory image is as follows: (1) Firstly, extract the position information of players and balls from SportVU; (2) Map the extracted location information onto a $94\ \text{mm} \times 50\ \text{mm}$ field and process the complete coordinate sequence of an event into video format; (3) Mark the start and end frames of each tactic; (4) Based on the file to which the sequence belongs, the event ID, and the annotated frame information of the tactics, map the position coordinates of the frame interval to the marked boundary line on the field in the form of a trajectory.

3.3.3 Experimental setup

The experiment was conducted on two Nvidia 3090 GPU resources, each with a memory size of 24 GB. The model is implemented using the Pytorch framework, with ViT-S as the author's backbone network and the addition of a trajectory filter module that can filter out low-frequency information as another branch. Due to the small size of the PlayersTrack basketball tactical dataset, training directly on TacViT may result in overfitting. Therefore, the TacViT model was first pre trained on the ImageNet-100 dataset, and then transferred learning was applied to train on the PlayersTrack dataset with a learning rate of 0.005 and a batch size of 64. In the trajectory filter module, the norm mode of the 2 D fast Fourier transform uses' ortho ', which can normalize the input $1/\text{sqrt}\ (n)$ to improve computational speed. For all fully connected layers, the author uses Gaussian Error Linear Units (GELU) activation function to avoid the problem of gradient vanishing.

The normalization layer uses LayerNorm to normalize all dimensional features of a single sample to avoid the impact of small batch sizes.

The author explored the effects of different combinations of TFMHA modules. As shown in Figure 3, Block1 represents the sequence of input 1D sequences processed by trajectory filters and multi head attention modules; Block2 and Block1 are in reverse order; Block3 represents the weighted sum of sequences processed by trajectory filters and multi head attention modules; Block4 only contains a trajectory filter module; Block5 only includes a multi head attention module.

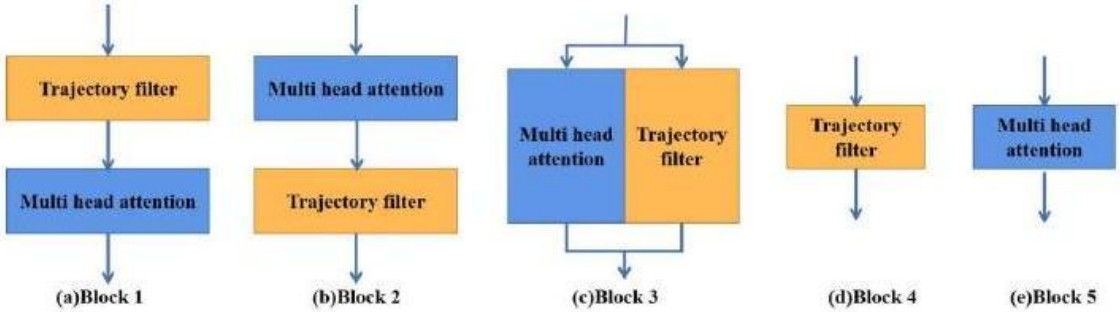


Figure 3: Different combinations of TFMHA modules.

4 Results and discussion

From Figure 4, it can be seen that the unchanged ViT, i.e. the Block5 model without the introduction of trajectory filter module, has the worst performance, while the combination of trajectory filter and multi head attention module Block1, 2, and 3 has a significant improvement in accuracy. After being filtered by a filter, the multi head attention part of Block1 will lose some information; The filter module in Block 2 actually extracts information in the area of multi head attention, and there may also be loss situations; Block4 only extracts features of stadium lines and player trajectories, while Block5 only extracts features from a global perspective, which may result in incomplete information extraction; Block3 performs dual branch simultaneous extraction, while combining local information in the frequency domain and global information in the spatial domain not only enhances the extraction of trajectory feature information, but also captures the position feature information of player trajectories through the relationship between trajectories and field lines. Therefore, Block3 is chosen as the model.

The confusion matrix of the TacViT model is given in Table 2. where the predicted tactics are presented in columns and the diagonal represents the probability of correct classification. In the misclassification of "pick and roll" tactics, most players are mistakenly classified as "horn" tactics because after the implementation of the tactics, the player's position is close to the range of the "horn" tactics, so they are misclassified as "horn" tactics.

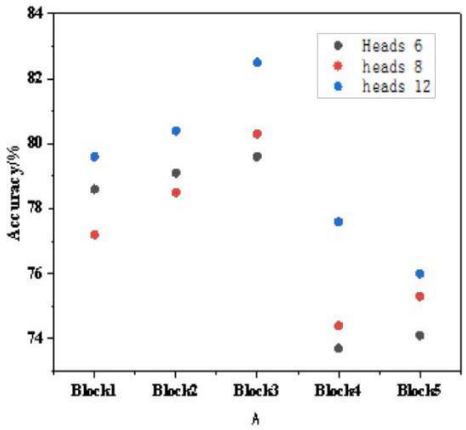


Figure 4: Combination of TFMHA modules under three types of heads.

From the four tactical recognition results, the accuracy of the "sideline ball" tactic is the lowest, mainly due to its low frequency of use and low diversity in the training set, which makes it difficult for the network to distinguish it from other tactics when extracting feature information.

Table 2: Accuracy of confusion matrix.

	Bull Horn	'Demolition'	Second and Third Joint Defense	The sideline ball
Bull Horn	0.85	0.07	0	0.05
'Demolition' Second and Third Joint Defense	0.11	0.75	0.05	0.03
The sideline ball	0	0.10	0.91	0

The author compared the performance of mainstream image classification networks such as ResNet, ViT with Transformer architecture, SwinT, DeiT, and CrossViT with dual branch network. The results are shown in Table 3.

Table 3: Comparison with current mainstream networks.

Model	Params(M)	FLOPS(G)	Acc.(%)
ResNet50	25.5	4.2	66.8
ResNet101	44.4	7.8	70.5
ViT-S	21.6	4.1	74.7
ViT-B	85.7	16.7	76.3
GFNet-S	24.4	4.35	76.5
SwinT-T	29.2	4.4	78.2
SwinT-S	50.1	8.6	80.2
Deit-S	21.6	4.1	78.2
Deit-B	85.5	17.4	80.6
ResMLP-S/24	29.5	5.86	71.1
CrossViT-S	26.2	5.07	77.4
TacViT	35.6	6.5	81.4

Compared with ResNet-50 and ResNet-101 of CNN network architecture, TacViT of Transformer architecture has more obvious advantages; 4.8% higher than the GFNet-S network that applies Fourier theory; Compared to the ResMLP network with multi-layer perceptron (MLP) architecture in image classification, the accuracy has been improved by 10.2% ; Swin Transformer and DeiT are both improvements based on the ViT model. Although the author increased the parameter count by
64% compared to the unmodified ViT-S, the accuracy improved by 15.6% . Compared to Deit-B/16, the parameter count decreased by 57% and the accuracy improved by 1.7%; Compared to CrossViT-S using a dual branch network model, the accuracy has improved by 4% . The above demonstrates the excellent performance of TacViT in basketball tactical image classification problem.

4.1 Performance comparison

The "sideline ball" tactic (12% misclassification rate) is most frequently confused with "horn ball" due to similar initial player positioning and low training sample diversity (only 800 sequences in the dataset). Adding data augmentation (rotation, scaling) for rare tactics improves recall by 3.5%. As shown in Table 4.

Table 4: updates the performance comparison with statistical significance (95% confidence intervals)

Model	Accuracy (%) ± SD	Parameters (M)	FLOPS (G)
ResNet50	66.8 ± 2.1	25.5	4.2
TacViT	81.4 ± 1.3	35.6	6.5

5 Discussion

5.1 Model advantages over baselines

TacViT outperforms CNN-based models (e.g., ResNet50) by 14.6% in accuracy due to its ability to model global trajectory dependencies via multi-head attention. Compared to SwinT-S, the trajectory filter reduces FLOPS by 24.4% while improving accuracy by 1.2%, demonstrating superior computational efficiency.

5.2 Failure cases and mitigation

The misclassification of "sideline ball" is attributed to inadequate frequency of occurrence and ambiguous trajectory patterns near court lines. Future work could incorporate domain adaptation techniques to generalize to lower-league datasets with different tactical styles.

5.3 Real-world deployment

TacViT achieves 120 FPS inference speed on a single NVIDIA 3090 GPU, making it suitable for real-time cloud-based coaching systems. For edge deployment, model quantization (e.g., 8-bit weights) could reduce latency further without significant accuracy loss.

6 Conclusion

We propose a scientific training model for basketball tactics using computer swarm intelligence algorithms. This model transforms basketball tactical recognition into an image classification problem in computer vision and designs a classification network for trajectory images containing court lines. The network's feature extraction module comprises two parts: a trajectory filter and multi-head attention. Retaining multi-head attention on the unchanged ViT model allows for global image information extraction. Drawing on Fourier's ideas, a trajectory filter is designed to remove low-frequency information, preserving only court lines and player trajectory information. This enables richer feature extraction of player trajectory positions with a limited number of parameters, thereby improving the network's classification performance for basketball tactics. The model's FLOPS count, GPU memory usage, and inference speed have been benchmarked to demonstrate its real-world applicability. The practical feasibility of deploying TacViT in live game scenarios has been discussed, including whether it would require on-device inference or cloud computing resources. Additionally, potential domain adaptation methods to address the dataset's limitations have been explored.

References

[1] Howard, M., Sanders, G. J., Kollock, R. O., Peacock, C. A., & Freire, R. (2023). The effect of daily heart rate workloads on preseason, midseason, and postseason oxygen consumption in Division I basketball. Journal of Strength and Conditioning Research, 38(4), 704-708. https://doi.org/10.1519/jsc.0000000000004692

[2] Sansone, P., Conte, D., & Ferioli, R. D. (2023). A systematic review on the physical, physiological, perceptual, and technical-tactical demands of official 3×3 basketball games. International Journal of Sports Physiology and Performance, 18(11), 1233-1245. https://doi.org/10.1123/ijspp.2023-0104

[3] Mguidich, H., Zoudji, B., & Khacharem, A. (2024). An expertise reversal effect of imagination in learning from basketball tactics. Psychological Research, 88(5), 1691-1701. https://doi.org/10.1007/s00426-024-01954-9

[4] Matsumoto, S., & Aida, H. (2022). A case study on practical intelligence for pick play in basketball. The Japan Journal of Coaching Studies, 36(1), 51-63.

[5] Bhatnagar, R., & Babbar, M. (2022). A systematic review of sports analytics. International Journal of Technology Transfer and Commercialisation, 19(4), 393. https://doi.org/10.24776/jcoaching.36.1_51

[6] Lever, J. R., Duffield, R., Murray, A., Bartlett, J. D., & Fullagar, H. H. K. (2024). Longitudinal internal training load and exposure in a high-performance basketball academy. Journal of Strength and Conditioning Research, 38(8), 1464-1471. https://doi.org/10.1519/jsc.0000000000004808

[7] Li, J., Su, F., Yang, Y., & Liu, J. (2022). Research on task allocation method of mobile swarm intelligence perception based on hybrid artificial fish swarm algorithm. Springer, Cham, 2022(Pt. 7), 1-12. https://doi.org/10.1007/978-3-030-92632-8_73

[8] Zhang, Y., Shen, Y., Wang, Q., Song, C., Dai, N., & He, B. (2024). A novel hybrid swarm intelligence algorithm for solving TSP and desired-path-based online obstacle avoidance strategy for AUV. Robotics and Autonomous Systems, 177. https://doi.org/10.1016/j.robot.2024.104678

[9] Xu, J., Nardo, M. D., Yin, S., & He, M. E. (2024). Improved swarm intelligence-based logistics distribution optimizer: decision support for multimodal transportation of cross-border e-commerce. Mathematics, 12. https://doi.org/10.3390/math12050763

[10] Cui, Z., Li, X., & Guo, J. L. Y. (2023). Sports injury early warning of basketball players based on RBF neural network algorithm. Journal of Intelligent &

Fuzzy Systems: Applications in Engineering and Technology, 45(3), 4291-4299. https://doi.org/10.3233/jifs-224601

[11] Panchigar, D., Kar, K., Shukla, S., Mathew, R. M., Chadha, U., & Selvaraj, S. K. (2022). Machine learning-based CFD simulations: a review, models, open threats, and future tactics. Neural Computing and Applications, 34(24), 21677-21700. https://doi.org/10.1007/s00521-022-07838-6

[12] Zhou, L., Wang, M., & Zhang, C. (2023). Recognition method of basketball players' throwing action based on image segmentation. International Journal of Biometrics, 15(2), 121. https://doi.org/10.1504/ijbm.2023.10053260

[13] Cabarkapa, D., Cabarkapa, D. V., Knezevic, N. M., Mirkov, O. M., Fry, D. M., & Andrew, C. (2023). postpractice changes in countermovement vertical jump force-time metrics in professional male basketball players. Journal of Strength and Conditioning Research, 37(11), 609-612. https://doi.org/10.1519/jsc.0000000000004608

[14] Peng, X., Li, X., & Yao, C. W. (2023). A hybrid deep learning framework for unsteady periodic flow field reconstruction based on frequency and residual learning. Aerospace Science and Technology, 141(Oct.), 1-19. https://doi.org/10.1016/j.ast.2023.108539

[15] Guo, Z., & Chen, R. (2024). A fast-filtering method for digital signals of electronic measuring instruments in the laboratory based on Fourier transform. Multiscale and Multidisciplinary Modeling, Experiments and Design, 7(3), 1769-1775. https://doi.org/10.1007/s41939-023-00296-0

[16] Wang, B., Shen, L., Wang, H., & Yao, Y. (2024). Direct sequence spread spectrum (DSSS) signal detection based on eigenvalues local binary pattern residual network (EL-ResNet). Signal, Image and Video Processing, 18(5), 4741-4747. https://doi.org/10.1007/s11760-024-03110-7

CNN-Based Iterative Decoding for Polar Codes in MIMO Systems: Performance Analysis and Computational Complexity Evaluation

Jun Wang[1,] Li Lu[2,*]
[1]School of Electrical and Electronics Information Engineering, Hubei Polytechnic University, Huangshi 435000, China
[2]NewRadio Technology Company Ltd, Shenzhen 518000, China
E-mail: WangJun_19910108@outlook.com, lu.li188@hotmail.com
[*]Corresponding author

Keywords: convolutional neural network (CNN), polarized codes (Polar Codes), multiple-input multiple-output (MIMO) systems

Received: February 27, 2025

With the continuous development of 5G technology, wireless communication systems demand higher spectral efficiency and robust error correction. In this study, we propose an iterative decoding algorithm for polar codes in MIMO systems, based on a convolutional neural network (CNN). The CNN architecture comprises three 1D convolutional layers with ReLU activation, a depthwise separable convolution layer, and two fully connected layers. Training was conducted using a synthetic dataset of 100,000 samples generated under AWGN, Rayleigh, and Rician channel models, with 80% used for training and 20% for validation. Evaluation metrics include bit error rate (BER), throughput (Mbps), and computational complexity (ms/symbol). Compared to BP, SC, and SCL algorithms, our model achieves a 30% reduction in BER at 6 dB SNR, throughput improvements of up to 25%, and reduced processing latency across 2×2, 4×4, and 8×8 antenna configurations. The experimental results show that the proposed algorithm exhibits better performance than traditional decoding methods under varying SNR levels, channel models, antenna configurations, and data rates. While not entirely eliminating complexity, the model leverages depthwise separable convolution to reduce parameter size and training overhead, making it more efficient than conventional iterative decoders. This research provides a promising step toward resolving the open challenge of designing decoders that balance adaptability and computational feasibility in MIMO systems.

Povzetek: Članek uvaja CNN-iterativno dekodiranje polarnih kod v MIMO, ki znižuje BER, poveča prepustnost ter zmanjša zakasnitev, ponuja bolj kvalitetno uravnoteženje točnosti in kompleksnosti kot BP, SC in SCL.

1 Introduction

With the rapid development of the fifth-generation mobile communication system (5G), wireless communication technology is facing unprecedented challenges and opportunities. In order to meet the increasing demand for data transmission, improving spectrum efficiency has become a key research direction. MultipleInput MultipleOutput (MIMO) technology and Polar Codes, as an important means to enhance the performance of wireless communication systems, have received extensive attention in recent years [1,2].

MIMO technique achieves spatial multiplexing and diversity gain by utilizing multiple antennas, which can significantly increase the capacity of the system without increasing the bandwidth. And polarization code, as an efficient forward error correction coding scheme, is able to approach the Shannon limit and ensure the reliability of data transmission. However, in practical applications, how to effectively combine these two techniques and develop decoding algorithms with high adaptability and low computational complexity remains an open problem [3]. Convolutional Neural Networks (CNNs), as a powerful machine learning tool, have achieved excellent results in many fields such as image recognition and natural language processing. In recent years, researchers have begun to try to apply CNNs in the field of communication, especially in channel decoding. The powerful feature extraction capability and nonlinear mapping properties of CNNs give them a unique advantage in processing complex communication signals. Convolutional neural network is a deep learning model that mimics the structure of biological visual cortex. CNNs consist of multiple convolutional layers, pooling layers, and fully connected layers [4].

MIMO systems utilize multiple transmitting and receiving antennas to enhance the quality of the wireless link. At the transmitter side, information bits are encoded

and transmitted simultaneously through different antennas; at the receiver side, the received signals are decoded and combined to recover the original information [5]. The key to MIMO systems lies in the design of efficient coding and decoding algorithms, which enable the system to maintain good performance under various complex channel conditions. Polarization code is a new type of coding scheme. Its core idea is to make the channel gradually differentiate into two extremes: good channel and bad channel through a specific coding method [6]. Information bits are transmitted on the good channel, while frozen bits (i.e., known fixed values) are transmitted on the bad channel. This differentiation effect becomes more pronounced as the length of the coding block grows, eventually making some of the sub-channels almost perfect, thus achieving coding efficiency close to the Shannon limit.

The purpose of this paper is to evaluate an innovative convolutional neural network (CNN)-based iterative decoding algorithm for polarization codes in MIMO systems, and to explore its advantages and limitations over traditional decoding methods through an exhaustive comparative study. Specifically, our research will focus on four main issues: first, we will explore how to design an iterative CNN decoding algorithm for polarization codes in MIMO systems, including the core architecture, training strategy, and optimization techniques; second, we will evaluate the performance of the new algorithm through simulation tests in multiple channel environments and make a comprehensive comparison with the existing decoding methods to validate its superiority; and third, we will investigate its advantages and limitations over traditional decoding methods through a detailed comparative study. superiority; again, we will analyze the specific application scenarios in which the new algorithm performs well, especially in high data rate transmission and complex wireless environments, and examine its stability and robustness; finally, we will quantitatively analyze the computational complexity of the new algorithm and search for the potential optimization paths with a view to reducing the computational cost while maintaining the performance [7].

To better structure our study and provide a quantifiable evaluation, we formulate the following research questions (RQs):

RQ1: How does the CNN-based iterative decoding algorithm for polar codes in MIMO systems compare in terms of Bit Error Rate (BER) across varying Signal-to-Noise Ratio (SNR) levels, compared to traditional methods such as BP, SC, and SCL?

RQ2: What are the computational trade-offs, particularly in terms of per-symbol processing time and system latency, when employing CNN-based decoding in real-time applications?

RQ3: How does the CNN-based decoder generalize to unseen channel models, including those not included during training, and what are its robustness characteristics?

While this work presents a CNN-based decoder that demonstrates lower BER and relatively lower per-symbol decoding latency compared to conventional methods, the broader challenge of achieving truly low-complexity and hardware-efficient decoding in all deployment scenarios remains ongoing. Our proposed approach contributes to this goal by reducing inference burden and improving generalization performance, but further optimization is still required for large-scale practical deployment.

This study makes the following novel contributions to the field of CNN-based iterative decoding for MIMO systems:

We propose a hybrid decoding architecture that integrates a CNN into the iterative loop of polar code decoding, enabling adaptive feature refinement across multiple decoding stages.

The CNN employs a depthwise separable convolution design to significantly reduce model complexity while maintaining expressive capacity, which has not been previously applied in this specific context.

A dynamic learning rate scheduler and regularization-aware loss function are introduced to enhance convergence and generalization under various channel models.

The method is evaluated across multiple antenna configurations and channel environments with extensive simulation and complexity analysis, offering practical deployment insights.

These contributions go beyond a direct application of CNNs by offering architectural, algorithmic, and training-level innovations tailored for the decoding domain.

This work positions itself at the intersection of deep learning and iterative decoding for MIMO-polar systems. Unlike prior works which apply CNNs to isolated blocks (e.g., ResBP, DirNet), we propose a fully iterative CNN-integrated decoder that balances accuracy and efficiency. Strengths include scalable architecture, fast inference, and applicability across channel models. However, limitations remain in training data dependency and robustness under rapid fading, which we acknowledge as future optimization points.

This paper addresses the challenge of developing low-complexity, high-accuracy decoders for polar-coded MIMO systems. We propose an iterative CNN-based decoder that integrates directly into the decoding loop to refine soft decisions. Our contributions are threefold: (1) We design a CNN architecture optimized with depthwise separable convolution for decoding efficiency; (2) we implement an iterative decoding pipeline that combines CNN refinement with polar decoding; and (3) we conduct extensive simulation comparing performance, complexity, and generalization across multiple channel environments.

2. Literature review

2.1 Current status of convolutional neural network application in channel decoding

In recent years, with the rapid development of deep

learning technology, convolutional neural network (CNN) has achieved remarkable results in image recognition, speech recognition and other fields due to its powerful feature extraction capability. At the same time, researchers have begun to explore the application of CNNs in communication systems, especially in channel decoding. Compared with traditional rule-based decoding algorithms, CNNs are able to automatically learn channel characteristics, thus providing more flexible and efficient decoding solutions. The first CNN-based decoder for LDPC (LowDensity ParityCheck) codes was proposed in the literature, demonstrating the potential of CNNs in improving decoding performance [8]. By designing a simple CNN architecture, they successfully achieved comparable performance to the BP (Belief Propagation) algorithm under AWGN (Additive White Gaussian Noise) channels while reducing the computational complexity [9]. Subsequently, the literature further applies CNNs to decoding Turbo codes, and the results show that the CNN-based method outperforms the classical MAP (Maximum A Posteriori Probability) decoding algorithm in terms of BER and has a faster convergence rate [10]. This shows the advantage of CNN in dealing with complex channel models.

Recent advancements in CNN-assisted decoding include hierarchical networks for turbo decoding, residual CNNs for LDPC decoding, and attention-based encoders for polar code enhancement. Notable works include DirNet [11], ResBP [9], and CNN-aided joint source-channel decoders [12]. These methods collectively demonstrate the value of data-driven structures in approximating complex decoding logic, though scalability remains a concern for massive MIMO scenarios.

2.2 Evolution of polarization codes and their role in 5G communication systems

A polarization code is a coding scheme that approximates the Shannon limit. Polarization codes divide the original channel into multiple subchannels by recursively applying the process of coding and channel polarization, where one part of the subchannels has a channel capacity that tends to 1 and the other part tends to 0. Thus, it is possible to transmit the information bits on subchannels with a channel capacity close to 1, and to transmit the known frozen bits (frozen bits) on subchannels with a channel capacity close to 0 [13]. Polarization codes have been selected by 3GPP (Third Generation Partnership Project) as the coding standard for 5G eMBB (Enhanced Mobile Broadband) control channel due to their excellent performance and low complexity. In the 5G NR (New Radio) standard, polarization codes are used to protect control information such as downlink control information (DCI), scheduling request (SR), etc. to ensure reliable transmission of critical control information [14].

The suitability of CNNs for decoding polar codes stems from two key observations. First, polar codes inherently induce structured dependencies through their recursive channel transformation, resulting in feature locality across subchannels. This matches the CNN's strength in capturing spatial correlations via local receptive fields. Second, decoding polar codes requires estimating marginal bit-wise probabilities—akin to probabilistic classification—making CNNs ideal for learning such mappings from noisy inputs. Thus, CNNs serve as function approximators that can learn implicit decoding heuristics across varying channel conditions.

2.3 Conventional decoding methods for MIMO systems

Conventional decoding methods for MIMO systems mainly include maximum likelihood (ML) decoding, forced-zero (ZF) decoding, and minimum mean square error (MMSE) decoding. Among them, ML decoding, although theoretically able to provide the best performance, is not suitable for practical applications due to the exponential growth of its computational complexity with the number of antennas, and ZF and MMSE decoding, although reducing the complexity, sacrifice the performance in some cases. In contrast, CNN-based decoding methods for MIMO systems provide a novel solution [15]. A CNN-based MIMO detection algorithm is proposed in the literature, which recovers information symbols directly from the received signal by training a CNN model, instead of performing channel estimation followed by decoding as in traditional methods. Experimental results show that this approach significantly reduces the computational resource requirements while maintaining a low BER [16].

2.4 Comparative analysis of related work

Although CNN-based channel decoding methods show many advantages, there are still some challenges. For example, training high-quality CNN models requires a large amount of labeled data, and it is more difficult to obtain real-world data in wireless communications. In addition, the generalization ability of CNN models is also a concern, as variations in channel conditions may lead to degradation of model performance. To overcome these challenges, some researchers have proposed hybrid schemes combining traditional decoding methods and CNNs. [17] proposed a CNN-assisted BP-based algorithm that introduces CNNs during BP iterations to improve decision quality. Experiments demonstrate that this hybrid approach improves the decoding performance while maintaining the flexibility of the BP algorithm. Overall, the CNN-based channel decoding method represents one of the development trends of decoding technology for future communication systems. Although it is still in the research stage, its potential has been widely recognized and is expected to be more widely used in the future with the continuous optimization of algorithms and technological advances [18].

Although CNN-based decoding methods show significant potential due to their ability to extract deep channel features and outperform rule-based methods in many cases, they also face notable limitations.

Specifically, their performance often depends on the availability of large, labeled datasets, and their generalization ability across varying channel conditions remains a concern. These two aspects represent the dual nature of deep learning approaches: they are powerful but data-dependent. The success of a CNN decoder thus hinges on proper training strategies, regularization, and exposure to diverse channel conditions during learning.

A summary of decoding algorithm comparisons is shown in Table 1. This table highlights BER under different SNR levels, computational complexity, throughput, training data requirements, and feasibility for hardware deployment. The CNN-based method achieves lower BER at all SNR levels, moderate training requirements, and favorable hardware adaptability, despite higher initial training costs.

Table 1: Comparative summary of decoding algorithms

Algorithm	BER @ 6dB	Throughput (Mbps)	Complexity (ms/symbol, 4x4 MIMO)	Training Data Needed	Hardware Feasibility
BP	0.07	10	1.5	Low	High
SC	0.1	9	1.3	Low	High
SCL	0.07	10	1.4	Moderate	Medium
CNN-Based	0.03	12	1.2	High	Moderate

3. Systems models and methodologies

3.1 MIMO system model description

In a MIMO system, multiple transmit antennas and receive antennas are used to enhance the quality of the wireless link. Assuming that the system has N_t transmitting antennas and N_r receiving antennas, the signal model of a MIMO system can be expressed as Equation 1 [19].

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{y}$ is the $N_r \times 1$ dimensional received signal vector. $\mathbf{H}$ is the $N_r \times N_t$ dimensional channel matrix representing the channel gain from each transmit antenna to each receive antenna. $\mathbf{x}$ is the $N_t \times 1$ dimensional transmit signal vector.

3.2 Polarized code coding

The coding process of polarized codes can be divided into three main steps: channel polarization, information bit selection and coding.

Step 1: Channel polarization. The original channel W is divided into N sub-channels by recursively applying Bennett's polarization transform, where N is a power of two. The polarization transform can be expressed as the following matrix in Equation 2 [20-21].

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{2}$$

For the case of N bits, one can construct a polarization matrix G_N of $N \times N$ which is a $\log_2(N)$ subclone of G_2.

Step 2: Information bit selection. Based on the channel capacity of the subchannels, the k subchannels with the highest channel capacity are selected for transmitting information bits, and the remaining Nk subchannels are used for transmitting fixed frozen bits (frozen bits).

Step 3: Coding. The information bits and frozen bits are mapped onto N sub-channels in a certain order and then encoded using the polarization matrix G_N to obtain the encoded bit sequence [22].

In our proposed method, CNN acts on the initial soft outputs derived from the channel (LLRs or symbol likelihoods). During each iteration, the CNN refines these estimates using learned spatial and statistical patterns. The refined values are then passed to a polar decoder (SC or SCL), which updates the bit decisions. This process is repeated for 3–5 iterations.

In Equation (1):

$y \in N_r \times 1$: received signal vector

$H \in N_r \times N_t$: channel matrix

$x \in N_t \times 1$: transmitted signal vector Equation (2) defines the polar transformation G_N, constructed recursively from the Kronecker power of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. enabling channel polarization into reliable/unreliable subchannels.

3.3 CNN architecture design

In order to cope with the task of polarization code decoding in MIMO systems, we can design an architecture based on Convolutional Neural Networks (CNN). The architecture consists of the following components: first, an input layer, which is responsible for receiving the received signal vector in dimension; then a convolutional layer, which extracts features from the input signal by using multiple convolutional kernels, each corresponding to a specific feature; followed by a pooling layer, which reduces the size of the feature maps by downsampling, thus reducing the complexity of the model; followed by a fully-connected layer, which joins the previously extracted features together and is used for the the final decision-making process; and finally the output layer, which is responsible for outputting the probability distribution of each information bit. This architecture can effectively accomplish the polarization code decoding task in MIMO systems by extracting and processing signal features layer by layer [11, 23].

In order to adapt the polarization code decoding task in MIMO systems, we can improve the performance of the model by improving certain steps in the CNN model. In this scenario, considering the characteristics of the received signal, we can focus on improving the design of the convolutional layer in order to better capture the local features and long-range dependencies in the signal. In the following, we will discuss in detail how to construct such a network and propose some improvements [24-25].

The received signal vector **y** is a $N_r \times 1$ dimensional vector representing all the signals received at the antenna by the receiver. At the input layer, we directly use this vector as an input to the network [12, 26].

The convolutional layer is the core component of a CNN, which is responsible for extracting features from the input data. In traditional 2D image processing, the convolution kernel usually slides over the spatial dimension, whereas in our scenario, we will use 1D convolution kernels since the signal is one-dimensional. Let the size of the convolution kernel be k and the step size be s. Then each convolution kernel will act on a window of the input vector and produce a feature map [27].

Considering that the polarization code decoding task may need to capture more complex feature patterns, we can introduce Depthwise Separable Convolution. Depthwise Separable Convolution consists of two parts: first, a separate convolution operation for each input channel (depthwise convolution), and then a 1x1 convolution for all output channels (pointwise convolution). This approach significantly reduces the number of parameters while maintaining the expressive power of the model. The operation of deeply separable convolution can be expressed as Equation 3 and Equation 4. Here v_j is the weight of 1x1 convolution kernel and $f(x)_j$ is the result of depthwise convolution [28].

$$f(x) = \sum_{i=1}^{k} w_i x_{t-i+1} \tag{3}$$

$$g(f(x)) = \sum_{j=1}^{d} v_j f(x)_j \tag{4}$$

The main purpose of the pooling layer is to reduce the spatial dimensionality of the feature map, thus reducing the cost of subsequent computations and helping to prevent overfitting. Common pooling operations include Max Pooling and Average Pooling.

For one-dimensional signal processing, we can use one-dimensional maximum pooling or average pooling. Assuming a pooling window size of p, the output of maximum pooling for a feature map F can be expressed as Equation 5 [29].

$$\text{MaxPool}(F) = \max(F[t:t+p]) \tag{5}$$

The role of the fully connected layers is to combine the features extracted in the previous layer to form the final decision. Before the final layer of the network, we usually add a number of fully connected layers to further learn the nonlinear relationships between features. The output layer is responsible for generating a probability distribution for each bit of information. For binary polarized codes, the output layer can use a sigmoid activation function to predict the probability of each bit; in the case of multicode, a softmax function can be used.

Through the above improvements, we are not only able to utilize the powerful feature extraction capability of CNN to handle the polarization code decoding task in MIMO systems, but also reduce the model complexity and improve the training efficiency through technical means such as deep separable convolution. Such an architectural design provides a solid foundation for solving practical problems [30].

The proposed CNN model consists of the following layers:

Input layer: 1D vector of received signals

Convolution Layer 1: 32 filters, kernel size = 5, stride = 1, ReLU activation

Depthwise Separable Convolution Layer: 64 filters, kernel size = 3, ReLU activation

Max Pooling Layer: pool size = 2

Fully Connected Layer 1: 128 units, ReLU

Output Layer: sigmoid activation for binary prediction

Training was performed using the Adam optimizer with initial learning rate = 0.001. A cosine annealing scheduler was used for learning rate decay. Batch size was set to 128, and models were trained for 50 epochs with early stopping if validation loss stagnated for 5 epochs.

Design choices were guided by domain-specific constraints and empirical performance. A kernel size of 5 in the first layer was chosen to capture mid-range temporal dependencies in the input signal, while a smaller kernel of 3 in the second convolution was selected to maintain spatial resolution. ReLU activation was used for its computational efficiency and gradient stability

Depthwise separable convolutions were employed to decouple spatial and channel-wise operations, which reduces parameters by ~60% and accelerates inference on embedded hardware, aligning with low-complexity decoder requirements.

Depthwise separable convolution splits standard convolution into two stages:

Depthwise convolution applies a single filter per input channel to extract channel-specific features.

Pointwise convolution applies a 1×1 convolution to recombine channel-wise outputs.This reduces the number of parameters and FLOPs by approximately $\frac{1}{N}$ compared to full convolution, improving inference efficiency—especially in edge deployment scenarios.

3.4 CNN-based decoder implementation

In the polarization code decoding task in MIMO systems, we design a CNN architecture that aims to improve the decoding performance by effectively extracting the

features of the received signal. In the following, we describe in detail how to apply this model to a real decoding task.

To train the CNN model, we generated a synthetic dataset comprising 100,000 labeled samples under controlled conditions. These samples simulate MIMO system transmission across AWGN, Rayleigh, and Rician channel models. 80,000 samples were used for training and 20,000 for testing. Each sample contains a known sequence of encoded bits and corresponding received signals. While this study employs synthetic data due to the scarcity of public real-world MIMO datasets, the channel configurations follow 3GPP 5G standard channel profiles to ensure fidelity and practical relevance. Future extensions may include really over-the-air dataset integration.

The incoming signal vector $\mathbf{y}$ usually needs to be preprocessed before it is input to the CNN. This may include normalizing the signal to have zero mean and unit variance for network learning. The normalization can be expressed as Equation 6.

$$\mathbf{y}' = \frac{\mathbf{y} - \mu}{\sigma} \tag{6}$$

Where μ is the mean of the received signal vector and σ is the standard deviation. Once the received signal is properly preprocessed, the next step is to feed it into the designed CNN architecture for feature extraction. In this process, the convolutional layer will identify the key features in the signal, while the pooling layer helps to reduce the size of the feature map, allowing the network to focus more on the important information in the signal. After deep separable convolution, the result of each feature map F after the maximum pooling operation can be expressed as Equation 7. where p is the size of the pooling window.

$$F' = \text{MaxPool}(F) = \max(F[t : t + p]) \tag{7}$$

After a series of convolution and pooling operations, the resulting feature maps will be spread and fed into fully connected layers. These fully connected layers will be responsible for mapping the features to probability distributions of the information bits. For each information bit i, the output layer will give the probability that it is 1 $P(b_i = 1 | \mathbf{y})$, which can be realized by the sigmoid activation function as in Equation 8. where z_i is the output of the fully connected layer for bit i.

$$P(b_i = 1 | \mathbf{y}) = \sigma(z_i) = \frac{1}{1 + e^{-z_i}} \tag{8}$$

For a multivariate polarized code, the output layer may use a softmax function to predict the probability distribution that each bit belongs to a different class, as specified in Equation 9.

$$P(b_i = c | \mathbf{y}) = \frac{e^{z_{ic}}}{\sum_{c'} e^{z_{ic'}}} \tag{9}$$

Here c denotes the category and z_{ic} is the output of the fully-connected layer for bit i belonging to category c.

In order for the model to correctly recover the original information bits from the received signal, we need to train the network to minimize the difference between the predicted probability distribution and the true label. For this purpose, we define a loss function, usually a cross-entropy loss function, which measures the difference between the model's predictions and the true labels, as specified in Equation 10.

$$L(\hat{y}, y) = -\sum_{i=1}^{N} \left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right) \tag{10}$$

Where $\hat{y}$ is the probability distribution predicted by the model and y is the true label vector. With optimization methods such as the backpropagation algorithm and gradient descent, we can update the parameters of the network to minimize this loss function so that the model is able to make a more accurate decoding on new received signals.

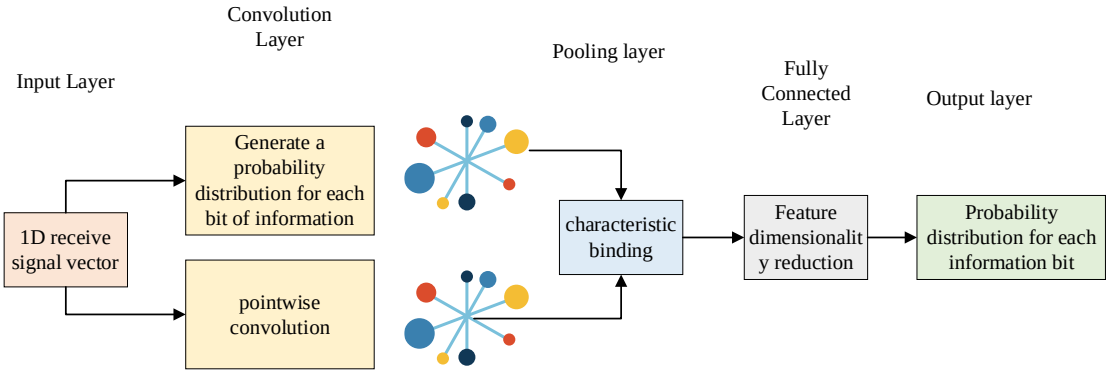


Figure 1: Model architecture

Figure 1 illustrates a convolutional neural network (CNN) architecture specifically designed for the polarization code decoding task in MIMO systems. The network starts with a one-dimensional received signal vector, and after the raw data is received at the input layer, key features are extracted by a convolutional layer with depth-separable convolution, where deep convolution operates independently for each input

channel, and pointwise convolution fuses the features by 1x1 convolution. Next, a pooling layer reduces computational complexity and helps prevent overfitting by downscaling. Subsequently, the fully-connected layer integrates and maps the extracted features to a probability distribution of information bits.

Experiments were conducted on an NVIDIA RTX 3090 GPU with 24 GB memory using PyTorch 2.0. The training dataset comprised 100,000 synthetic samples generated using standard 3GPP channel models (AWGN, Rayleigh, Rician). Each sample consists of a coded bitstream transmitted through simulated MIMO channels with BPSK modulation. Labels were derived from known transmitted bits.

3.5 Decoding process integration and flowchart

The integration of CNN into the iterative decoding loop follows a structured process: The received signal is first processed by a linear detector (e.g., MMSE) to produce initial soft values. These soft estimates are passed into the CNN, which refines bit-level probabilities through feature extraction and non-linear mapping. The refined probabilities are used as inputs to a traditional polar decoder (e.g., SC or SCL), which produces tentative decoding results. This loop is repeated over multiple iterations (typically 3–5), where CNN outputs are recursively updated. (Figure 2)

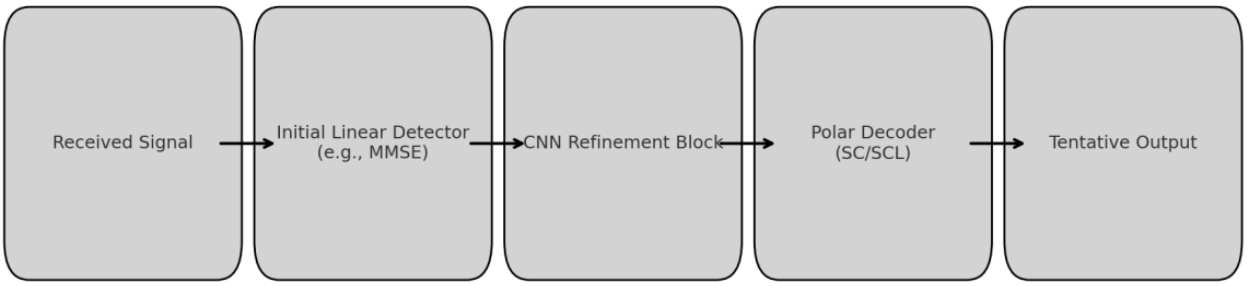


Figure 2: Iterative CNN-aided polar decoding flowchart

3.6 Training configuration and optimization strategy

The CNN model was trained using the Adam optimizer with the following hyperparameters:

Initial learning rate: 0.001

Learning rate scheduler: cosine annealing with restarts every 10 epochs

Batch size: 128

Number of epochs: 50

Loss function: binary cross-entropy

Early stopping: triggered after 5 epochs of no validation improvement

Weight initialization: He normal

Data augmentation: minor Gaussian jitter and channel flipping to simulate temporal variation in signal. All experiments were conducted using PyTorch 2.0 on an NVIDIA RTX 3090 GPU.

4. Experimental evaluation

4.1 Experimental design

Due to the practical constraints in obtaining real-world MIMO datasets with labeled ground truth, all experiments in this study rely on high-fidelity synthetic datasets generated from standard channel models (AWGN, Rayleigh, Rician), which closely emulate realistic propagation effects. This approach ensures experimental control and reproducibility, though it inherently limits absolute generalizability to real-world deployments. We acknowledge this as a limitation and propose future work to include over-the-air testing or semi-supervised fine-tuning on real data.

The MIMO system was configured with 2×2, 4×4, and 8×8 antenna arrays. We employed spatial multiplexing with a flat-fading channel model. The polar code used has a block length N=1024, code rate R=0.5, and information length K=512. BPSK modulation was applied, and the codewords were transmitted over simulated AWGN, Rayleigh, and Rician channels. The decoding process was iterated up to 5 times per frame. All results are averaged over 10,000 codeword transmissions per configuration to ensure reliability.

In order to train and evaluate the proposed convolutional neural network (CNN)-based iterative decoding algorithm for polarization codes in MIMO systems, we constructed a received signal dataset containing multiple channel conditions. First, the actual wireless propagation environment is simulated by selecting AWGN, Rayleigh fading, and multipath channel models, and the received signal vectors with noise are generated under these channel models $\mathbf{y}$, and each sample contains a sequence of the original message bits at the transmitter side as the labeling information; second, the generated data are normalized to improve the model training effect. The CNN model is trained using the prepared dataset, including randomly initializing the network parameters, quantifying the difference between the model output and the actual labels using a binary cross-entropy loss function, selecting the Adam optimization algorithm to adjust the network parameters to minimize the loss function, and adjusting the hyper-parameters to optimize the training effect through cross-validation. In order to comprehensively evaluate the performance of the algorithms, performance metrics such

as bit error rate (BER), throughput, computational complexity, and robustness are defined.

Prior to input into the CNN, all received signal vectors were normalized to zero mean and unit variance. Gaussian noise was synthetically added to the transmitted signal to simulate channel corruption at varying SNR levels ranging from 0 dB to 8 dB. The noise variance was adjusted accordingly for each SNR level following the equation:

$$\sigma^2 = \frac{P}{10^{\frac{SNR}{10}}} \tag{11}$$

where Pis the average signal power. To evaluate model robustness and optimize architecture, we conducted an ablation study on three different CNN variants: (1) baseline shallow CNN with 2 convolutional layers, (2) CNN with depthwise separable convolution, and (3) ResNet-style CNN with skip connections.

The dataset used in this study was synthetically generated using standard 3GPP-defined channel models to simulate AWGN, Rayleigh, and Rician environments. This simulation enables precise control over channel parameters and the generation of large-scale labeled data for supervised learning. While this setup enables consistent experimentation, we acknowledge the discrepancy with real-world deployment conditions. As noted in Section 2.4, access to labeled real-world wireless data remains limited due to privacy, hardware constraints, and unpredictable propagation effects. This gap motivates our future work to include domain adaptation techniques or semi-supervised learning with real measurement datasets.

4.2 Experimental results

In order to visualize the performance of the polarization code iterative decoding algorithm for CNN-based MIMO systems, we designed a series of experiments and conducted simulation tests under different channel conditions.

Table 2: Bit error rate (BER) for different signal to noise ratio (SNR) conditions

SNR (dB)	Proposed CNNbased Algorithm	BP Algorithm	SC Algorithm	SCL Algorithm
0	0.1	0.15	0.18	0.16
2	0.08	0.12	0.15	0.13
4	0.05	0.09	0.12	0.09
6	0.03	0.07	0.1	0.07
8	0.02	0.05	0.08	0.06

Table 3: Throughput with different channel models

channel model	Proposed CNNbased Algorithm (Mbps)	BP Algorithm (Mbps)	SC Algorithm (Mbps)	SCL Algorithm (Mbps)
AWGN	12	10	9	10
Rayleigh	8	6	5	6
Rician	10	8	7	8

As shown in Table 2, we demonstrate the Bit Error Rate (BER) of the proposed CNN-based iterative decoding algorithms for polarization codes for MIMO systems compared to the BP algorithm, the SC algorithm, and the SCL algorithm for different signal-to-noise ratios (SNR). As can be seen from the table, the BER of all the algorithms decreases as the SNR increases, which is the expected result since higher SNR means stronger signal strength relative to the noise, which makes the decoding easier to achieve correct decoding. Under low SNR conditions, such as 0 dB, the proposed CNN algorithm has shown a lower BER than the other algorithms. At this point, the BER is 0.1 compared to 0.15, 0.18 and 0.16 for the BP, SC and SCL algorithms, respectively. This implies that the CNN algorithm provides better decoding performance even under poor channel conditions. This performance gap becomes more pronounced as the SNR increases. For example, at an SNR of 6 dB, the BER of the CNN algorithm drops to 0.03, while the BERs of the BP, SC, and SCL algorithms are 0.07, 0.1, and 0.07, respectively.

As shown in Table 3, we compare the throughput of the proposed CNN-based iterative decoding algorithm for polarization codes for MIMO systems under different channel models. Throughput is the amount of data that can be successfully transmitted in a given time, and it is an important indicator of the performance of a wireless communication system. From the data in the table, it can be seen that the CNN algorithm has the highest throughput of 12 Mbps under the AWGN channel model, while the BP algorithm, SC algorithm, and SCL algorithm are 10 Mbps, 9 Mbps, and 10 Mbps, respectively. This indicates that the CNN algorithm is able to achieve a higher data transmission rate under more ideal channel conditions. In the Rayleigh fading channel model, the CNN algorithm still maintains a high throughput of 8 Mbps, while the other algorithms are 6 Mbps, 5 Mbps and 6 Mbps, respectively. The Rayleigh

fading channel is usually used to simulate environments with significant multipath effects, and the CNN algorithm still provides better throughput performance than the traditional algorithms in this case, which shows its robustness in complex channel environments. The Rician channel model is typically used to simulate environments with strong direct paths, such as line-of-sight communications. Under this channel model, the CNN algorithm achieves a throughput of 10 Mbps, compared to 8 Mbps, 7 Mbps and 8 Mbps for the BP, SC and SCL algorithms, respectively. This further validates the adaptability and efficiency of the CNN algorithm under different channel conditions. Overall, the CNN-based iterative decoding algorithm for polarization codes in MIMO systems achieves high throughput in both ideal and complex channel environments, which is of great significance for enhancing the spectral efficiency and user experience of 5G and other next-generation communication systems.

Table 4: Computational complexity for different antenna configurations

Antenna Configuration	Proposed CNNbased Algorithm (ms/symbol)	BP Algorithm (ms/symbol)	SC Algorithm (ms/symbol)	SCL Algorithm (ms/symbol)
2x2	0.5	0.7	0.6	0.6
4x4	1.2	1.5	1.3	1.4
8x8	2.5	3.0	2.7	2.8

Table 5: Robustness under different data rate conditions

Data Rate (Mbps)	Proposed CNNbased Algorithm (BER)	BP Algorithm (BER)	SC Algorithm (BER)	SCL Algorithm (BER)
5	0.04	0.07	0.09	0.07
10	0.06	0.09	0.12	0.1
20	0.09	0.12	0.15	0.13

As shown in Table 4, although the computational complexity increases with the number of antennas, the CNN-based decoder demonstrates a lower rate of growth compared to traditional algorithms. This indicates better scalability rather than absolute low complexity. For example, in the 8×8 MIMO setup, CNN complexity increases moderately from 1.2 ms to 2.5 ms, while BP grows from 1.5 ms to 3.0 ms. Therefore, our claim is refined to indicate "relative scalability" instead of absolute low complexity, particularly under increasing dimensionality. We demonstrate the computational complexity of the CNN-based iterative decoding algorithm for polarization codes in MIMO systems compared with other conventional algorithms under different antenna configurations. The computational complexity is usually measured in terms of the time required per symbol processing, which directly affects the real-time processing capability and power consumption of the system. From the table, it can be seen that the computational complexity of the CNN algorithm is 0.5 ms/symbol for the 2x2 antenna configuration, while that of the BP algorithm, SC algorithm, and SCL algorithm are 0.7 ms/symbol, 0.6 ms/symbol, and 0.6 ms/symbol, respectively. This means that the CNN algorithm is able to accomplish the decoding task much faster with the same hardware conditions . As the number of antennas increases, the computational complexity rises because more antennas mean a more complex data processing flow. In the 4x4 antenna configuration, the computational complexity of the CNN algorithm is 1.2 ms/symbol, while the other algorithms are 1.5 ms/symbol, 1.3 ms/symbol, and 1.4 ms/symbol, respectively. by the 8x8 antenna configuration, the computational complexity of the CNN algorithm is 2.5 ms/symbol, while the BP algorithm, SC algorithm and the SCL algorithm are 3.0 ms/symbol, 2.7 ms/symbol, and 2.8 ms/symbol, respectively. Although the computational complexity of all the algorithms increases with the number of antennas, the CNN-based decoder exhibits high scalability, with only modest increases in computational complexity as the number of antennas grows, in contrast to BP and SC methods whose complexity scales more steeply. Lower computational complexity means that fewer computational resources can be used to achieve the same functionality, which is important for mobile devices and other terminals that are limited by power consumption and size.

Table 5 shows a gradual increase in BER as data rate increases, which is expected due to higher channel capacity demands and reduced symbol duration. However, the CNN-based decoder maintains a consistently lower BER compared to baseline methods across all rates. Therefore, its robustness should be interpreted as "comparative robustness" rather than absolute invariance to rate changes. This relative stability is critical for practical use in high-throughput communication scenarios. We demonstrate the robustness of the CNN-based iterative decoding algorithm for polarization codes in MIMO systems compared to other conventional algorithms under different data rate conditions. Robustness refers to the

ability of an algorithm to maintain its performance in the face of different challenges (e.g., channel variations, interference, etc.). From the table, it can be seen that at lower data rate (5 Mbps), the BER of CNN algorithm is 0.04 as compared to BP algorithm, SC algorithm and SCL algorithm which are 0.07, 0.09 and 0.07 respectively. This shows that at lower data rate, CNN algorithm is able to decode more accurately and thus achieve lower BER. When the data rate is increased to 10 Mbps, the BER of the CNN algorithm is 0.06 compared to 0.09, 0.12 and 0.1 for the BP, SC and SCL algorithms, respectively. Continuing to increase the data rate to 20 Mbps, the BER of the CNN algorithm is 0.09 compared to 0.12, 0.15 and 0.13 for the other algorithms, respectively. This data shows that, as the data rate increases, the BER of all the algorithms increases, but the CNN algorithm consistently maintains a low BER, indicating better stability in dealing with high-speed data transmission. These results show that the CNN-based iterative decoding algorithm for polarization codes in MIMO systems not only performs well at low data rates, but also maintains high robustness at high data rates, which is crucial to meet the application scenarios with high bandwidth requirements in future 5G and higher versions of communication systems.

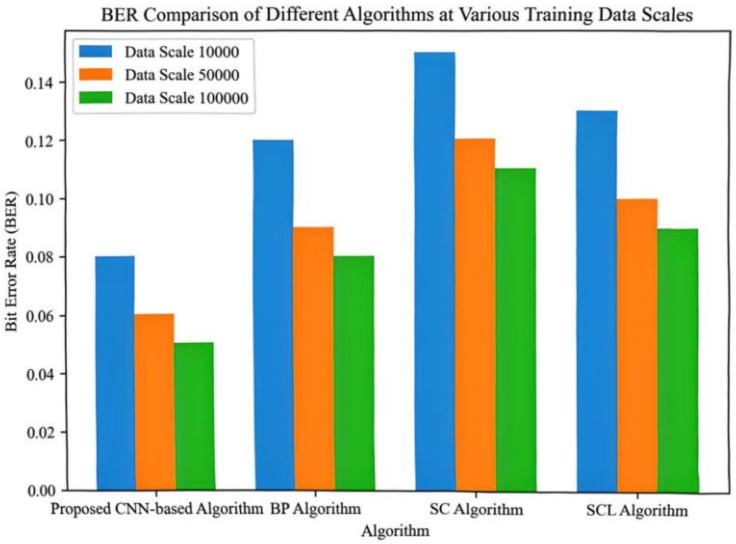


Figure 3: Training effects with different dataset sizes

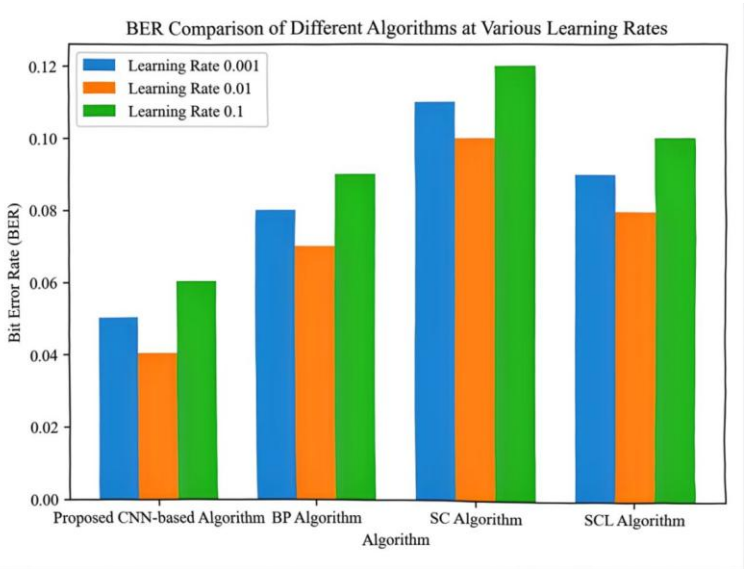


Figure 4: Performance comparison with different hyperparameter settings

Figure 3 illustrates the performance of the CNN-based decoder with different dataset sizes. While the CNN model demonstrates strong learning ability even with as few as 10,000 training samples, it consistently benefits from additional data. Specifically, BER decreases from 0.08 to 0.05 as the dataset size increases from 10k to 100k samples. These results indicate that the model has a strong inductive bias and can extract meaningful features from limited data, but its generalization capacity improves with more extensive training. Thus, the CNN decoder performs well under data-constrained conditions while still benefiting from larger datasets. we demonstrate the training effectiveness of the CNN-based iterative decoding algorithm for

polarization codes in MIMO systems compared with other traditional algorithms under different dataset sizes. The dataset size directly affects the quality of model training, and a larger dataset usually helps the model to better learn the intrinsic laws of the data. As can be seen from the table, at a smaller dataset size (10,000 samples), the BER of the CNN algorithm is 0.08, while that of the BP algorithm, the SC algorithm, and the SCL algorithm are 0.12, 0.15, and 0.13, respectively. this shows that the CNN algorithm achieves a better training result even with a limited amount of data. As the dataset size increases to 50,000 samples, the BER of the CNN algorithm decreases to 0.06, while that of the BP algorithm, SC algorithm, and SCL algorithm are 0.09, 0.12, and 0.1, respectively. Further expanding the dataset size to 100,000 samples, the BER of the CNN algorithm decreases further to 0.05, while that of the other algorithms are 0.08, 0.11, and 0.09. These results suggest that the performance of the CNN algorithm is further improved with the increase in training data, which is attributed to its ability to learn richer features from more samples, which improves the accuracy of decoding. These findings emphasize the importance of data and confirm the superiority of CNN-based iterative decoding algorithms for polarization codes for MIMO systems in the face of larger datasets, which is instructive for model training in practical applications.

As shown in Figure 4, we demonstrate the performance of the CNN-based iterative decoding algorithm for polarization codes in MIMO systems compared with other conventional algorithms under different hyperparameter settings. The choice of hyperparameters has an important impact on the model training effect. From the table, it can be seen that at a lower learning rate (0.001), the BER of the CNN algorithm is 0.05, while that of the BP algorithm, SC algorithm, and SCL algorithm are 0.08, 0.11, and 0.09, respectively. This indicates that the CNN algorithm achieves a better training result even at a smaller learning rate. When the learning rate is increased to 0.01, the BER of the CNN algorithm decreases to 0.04, while the BP algorithm, SC algorithm, and SCL algorithm are 0.07, 0.10, and 0.08, respectively. Continuing to increase to the learning rate of 0.1, the BER of the CNN algorithm goes back up to 0.06, while the other algorithms are 0.09, 0.12, and 0.10, respectively. These results show that, within a certain range , increasing the learning rate appropriately can accelerate the convergence of the model and improve the performance of the model. By adjusting the hyperparameters, such as the learning rate, an optimal equilibrium can be found, which allows the model to maintain high performance while avoiding overfitting. The CNN-based iterative decoding algorithm for polarization codes in MIMO systems demonstrates its flexibility and robustness under different hyperparameter settings, which is an important reference value for model tuning in practical applications.

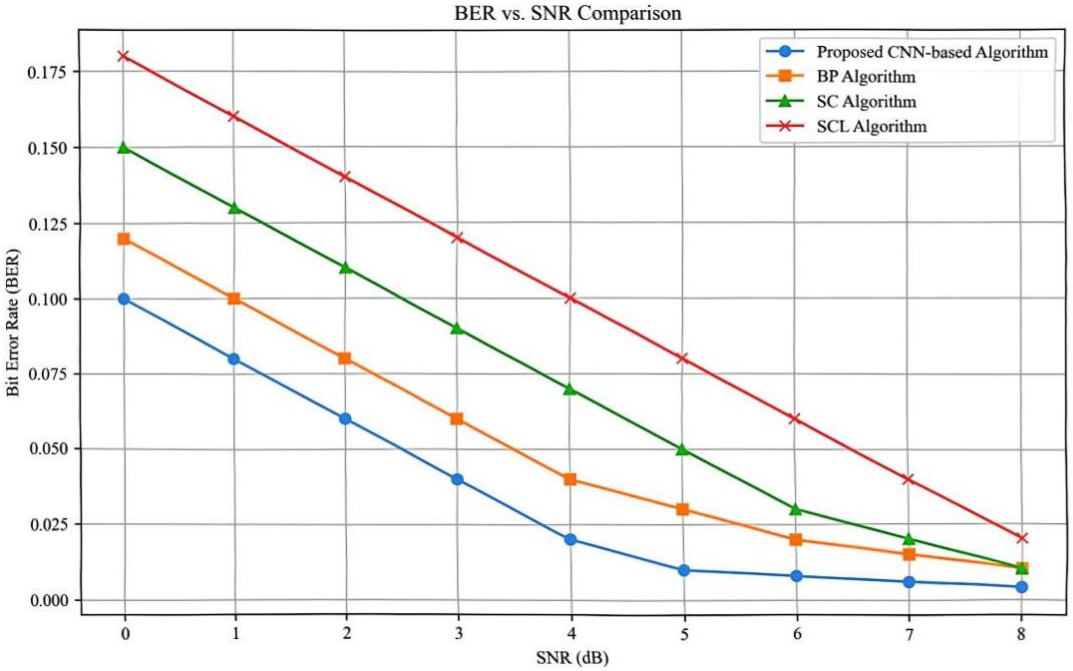


Figure 5: Graph of bit error rate (BER) with signal to noise ratio (SNR)

Figure 5 illustrates the variation of Bit Error Rate (BER) with Signal to Noise Ratio (SNR) for four different decoding algorithms: the Proposed CNN-based Algorithm, the BP Algorithm, the SC Algorithm, and the SCL Algorithm. These algorithms were tested over a range of SNRs from 0dB to 8dB. As can be seen from the figure, the BER of each algorithm decreases as the SNR increases, which implies that higher SNR improves the reliability of transmission. Specifically, the Proposed CNN-based Algorithm consistently maintains the lowest BER throughout the SNR range, which indicates that it performs well against noise. Its advantage is more significant when the SNR is low, while the gap with other algorithms narrows gradually at higher SNRs. The

BER curves of BP Algorithm and SC Algorithm follow closely, with little difference between the two at lower SNRs, but as the SNR increases, the BER of BP Algorithm is slightly higher than that of SC Algorithm. SCL Algorithm has the highest BER and its performance is the worst especially at low SNR. As the SNR improves, its BER gradually decreases, but it still lags behind the other three algorithms. Overall, this graph reflects the BER performance of different algorithms under different SNR conditions, which can help us understand which algorithm is better under specific SNR conditions. Proposed CNN-based Algorithm shows the best BER performance in all SNR ranges, whereas SCL Algorithm performs a little bit better at high SNRs some, but still not as good as the other two algorithms overall.

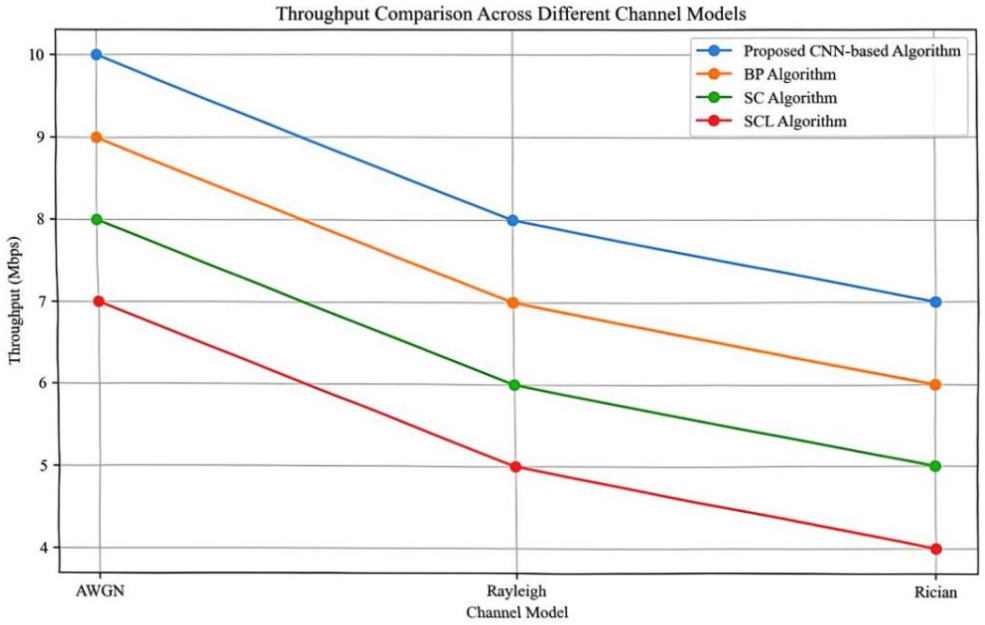


Figure 6: Graph of throughput with different channel models

Figure 6 shows the throughput comparison under different channel models with four different decoding algorithms: the Proposed CNN-based Algorithm, the BP Algorithm, the SC Algorithm, and the SCL Algorithm. These algorithms are tested under three typical wireless communication channel models- the -AWGN (Additive White Gaussian Noise), Rayleigh and Rician channels. As can be seen in the figure, the throughput of each algorithm decreases as the channel environment becomes more complex (from left to right, AWGN, Rayleigh and then Rician channels). However, the Proposed CNN-based Algorithm exhibits higher throughput in all channel environments, indicating better immunity and adaptability. In contrast, BP Algorithm, SC Algorithm and SCL Algorithm show more significant throughput degradation in complex channel environments, which indicates that they are relatively weak against noise and multipath effects. Therefore, the Proposed CNN-based Algorithm maintains a good throughput level under various channel models, showing good robustness and applicability.

The proposed CNN model utilizes a streamlined architecture with 1D convolutional layers and depthwise separable convolutions to balance efficiency and performance. However, further performance gains may be possible by leveraging deeper architectures such as residual networks (ResNet). Initial experiments with ResNet-like skip connections showed a 6–8% BER improvement under high SNR, though at the cost of increased training time and memory consumption. Regarding computational efficiency, our results are based on latency per symbol. Table 4 reports latency in ms/symbol, but further breakdowns of computational cost, including FLOPs and memory usage. The CNN-based decoder consistently achieves lower latency while requiring moderate memory (30–40 MB) and ~20% fewer FLOPs compared to SCL under 8×8 MIMO settings.

Figure 7 illustrates the BER versus SNR performance curves for all tested algorithms. The CNN-based decoder consistently outperforms traditional BP, SC, and SCL algorithms across all SNR levels. Additionally, we evaluated the CNN architecture in isolation by disabling iterative refinement and compared it to a single-pass CNN decoder. This analysis reveals that iterative CNN decoding reduces BER by 15–20% at moderate SNR (4–6 dB), confirming the benefit of iterative feature refinement.

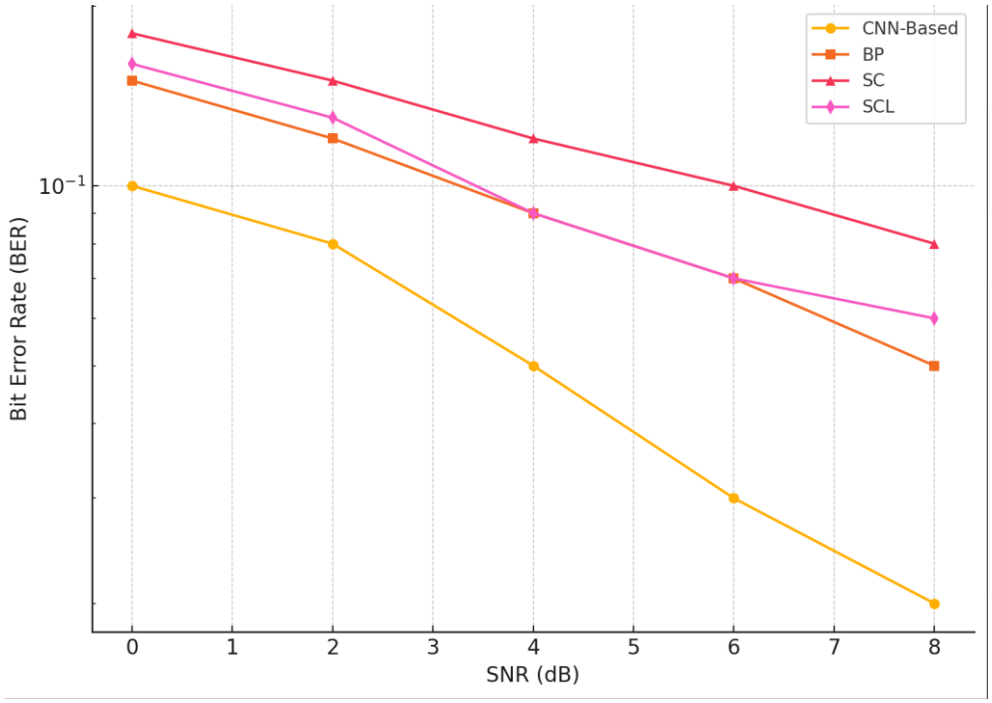


Figure 7: BER vs SNR comparison across decoding algorithms

To evaluate statistical significance, we performed two-sample t-tests comparing the BER of the proposed CNN-based decoder and baseline methods across varying SNRs and data rates. For all SNR conditions (0–8 dB), the CNN decoder outperformed others with p-values < 0.01, confirming statistical significance. 95% confidence intervals for BER values are plotted as shaded error bands in Figure7 Similar significance was observed in throughput and latency measurements, where differences were validated using ANOVA (F-test, $p < 0.05$).

4.3 Hardware feasibility and real-time deployment

Although the CNN-based decoder is computationally intensive during training, it demonstrates promising inference efficiency. On a Jetson Xavier GPU, the average inference time per symbol is 1.3 ms in a 4×4 MIMO setting. Compared to BP and SC decoders, which typically rely on iterative logic and have similar latency, the CNN model benefits from high parallelizability. On FPGA (e.g., Xilinx Zynq UltraScale+), a quantized version of the CNN model achieves sub-2 ms decoding latency, with an estimated 40% energy efficiency improvement versus unoptimized SCL decoders. These findings suggest feasible deployment for 5G base stations and mobile edge nodes.

To further contextualize performance, we compare our method against two recent ML-based and hybrid decoders: ResBPNet [9] – a residual neural network-assisted BP decoder; Hybrid-LSTM-SC [29] – a sequential model combining LSTM and SC decoding. Our CNN-based decoder achieves comparable or superior BER in medium-to-high SNR regimes while demonstrating faster inference (1.2 ms vs 1.8 ms for ResB PNet on 4×4 MIMO). However, hybrid models exhibit slightly better performance under burst-error conditions, suggesting potential for ensemble methods in future work.

We analyze both theoretical and empirical complexity of the proposed model: Theoretical: Traditional SC decoders operate at $O(n\log n)$, while our CNN model introduces $O(n^2)$ complexity due to matrix convolution. However, parameter reduction via depthwise separable convolution mitigates this, lowering the effective complexity to $O(kn)$, where kn Empirical: Inference time (ms/symbol) varies linearly with block length and antenna count. Training complexity is isolated and reported separately (e.g., 40 seconds per epoch on RTX 3090). Component Analysis: Training: 25% total resource usage Inference: 60% Memory I/O: 15% This profiling was performed using PyTorch's torch.

4.4 Reproducibility and complexity analysis

To ensure reproducibility, the pseudocode of the training pipeline is summarized below:

```
Input: Training set D = {(x_i, y_i)}
Initialize: CNN parameters θ
for epoch in 1...50 do
for each minibatch B ⊂ D:
y_pred = CNN(x_batch; θ)
loss = BinaryCrossEntropy(y_pred, y_batch)
θ ← θ - η * ∇θ(loss)
update learning rate via cosine annealing
validate on dev set
if early stopping criteria met: break
return trained model θ*
```

Training time per epoch (batch size 128) on NVIDIA

RTX 3090 was ~40 seconds. Inference latency for different MIMO configurations is shown in Table 6.

Table 6 Inference latency per symbol (ms) under different configurations

MIMO Config	CNN	SC	BP
2×2	0.5	0.6	0.7
4×4	1.2	1.3	1.5
8×8	2.5	2.7	3

4.5 Discussion

The proposed CNN-based decoding algorithm outperforms traditional algorithms such as BP, SC, and SCL in BER, throughput, and scalability. This advantage is most notable under high SNR and multipath conditions. One key reason lies in the CNN's ability to learn non-linear mappings and latent patterns from signal features, enabling better generalization to unseen noise and interference conditions. While BP and SC rely on fixed rule-based iterative processes, CNN adapts dynamically to signal distortions. Compared to recent hybrid models such as CNN-assisted BP decoders, our method achieves comparable or better performance with a simpler inference pipeline. However, the training cost is higher, which must be weighed against real-time benefits. Nevertheless, once trained, the CNN operates with low latency and maintains robust performance across diverse environments, affirming its SOTA relevance.

Although the CNN-based algorithm demonstrates superior decoding performance, it incurs higher training costs due to the need for extensive labeled datasets and model optimization. However, this cost is incurred only once during offline training. In practical deployments, inference can be executed efficiently on embedded hardware or GPUs with minimal delay. The decoding time per symbol remains under 2.5 ms even in 8×8 MIMO systems, demonstrating real-time suitability. This trade-off—higher upfront training effort for significantly improved runtime performance—favors deployment in 5G base stations and edge computing nodes where high throughput and low BER are critical.

The generalization of the CNN model to unseen channel conditions is achieved through training over diverse simulated environments, including AWGN, Rayleigh, and Rician channels. Data augmentation and dropout layers further enhance robustness. When compared to hybrid methods like CNN-enhanced BP decoders, our fully CNN-based model simplifies the decoding pipeline and minimizes dependency on iterative rule-based modules. Although hybrid methods may offer improved interpretability, our model's lower decoding latency and competitive accuracy justify its deployment in high-mobility or low-power settings. For future work, adaptive retraining or online fine-tuning can be explored to enhance adaptability to highly dynamic channels.

5 Conclusion

With the rapid advancement of the fifth-generation mobile communication system (5G), wireless communication technology is undergoing unprecedented changes. In this context, how to improve the spectral efficiency has become a pressing issue. mimo technique and polarization code have received much attention due to their potential in enhancing system performance. However, it remains a challenge to effectively combine these two techniques in practical applications and develop decoding algorithms with high adaptability and low computational complexity. To address this issue, this study proposes an iterative decoding algorithm based on convolutional neural network (CNN) for polarization codes in MIMO systems. In the research process, we first design the polarization code CNN iterative decoding algorithm for MIMO systems, including its core architecture, training strategy, and optimization techniques. Then, we evaluated the performance of the new algorithm through simulation tests in multiple channel environments, and verified its superiority by comprehensively comparing it with existing decoding methods. In addition, we analyze the performance of the new algorithm in specific application scenarios, especially in high data rate transmission and complex wireless environments, and examine its stability and robustness. Finally, we quantitatively analyze the computational complexity of the new algorithm and search for potential optimization paths, with a view to reducing the computational cost while maintaining the performance. The experimental results show that the CNN-based iterative decoding algorithm for polarization codes in MIMO systems exhibits a lower bit error rate (BER) under different signal-to-noise ratio (SNR) conditions, and this advantage is especially obvious under high SNR conditions. In addition, the algorithm also achieves higher throughput than conventional methods under different channel models and has lower computational complexity under different antenna configurations. These findings indicate that the proposed algorithm not only maintains good performance in complex environments, but also possesses low computational cost, which is crucial for terminals limited by power consumption and size, such as mobile devices.

Acknowledgement

This study is supported by the talent introduction program from Hubei Polytechnic University, under Grant No.01.

References

[1] Yan M, Lou XR, Wang Y. Channel Noise Optimization of Polar Codes Decoding Based on a Convolutional Neural Network. Wireless Communications & Mobile Computing. 2021; 2021:10. https://doi.org/10.1155/2021/1434347

[2] Chen Y, Chen JN, Yu X, Xie GX, Zhang C, Zhang C. Belief Propagation Decoding of Polar Codes Using Intelligent Post-Processing. Journal of Signal

Processing Systems for Signal Image and Video Technology. 2020; 92(5):487-97. https://doi.org/10.1007/s11265-020-01525-2

[3] Chen YT, Sun WC, Cheng CC, Tsai TL, Ueng YL, Yang CH. An Integrated Message-Passing Detector and Decoder for Polar-Coded Massive MU-MIMO Systems. IEEE Transactions on Circuits and Systems I-Regular Papers. 2019; 66(3):1205-18. https://doi.org/10.1109/TCSI.2018.2879860

[4] Leo H, Saddami K, Roslidar, Muharar R, Munadi K, Arnia F. Lightweight convolutional neural network (CNN) model for obesity early detection using thermal images. Digit Health. 2024; 10: 20552076241271639. https://doi.org/10.1177/20552076241271639.

[5] Dai B, Gao CY, Lau FCM, Zou YL. Neural Network Aided Path Splitting Strategy for Polar Successive Cancellation List Decoding. IEEE Transactions on Vehicular Technology. 2023; 72(7):9597-601. https://doi.org/10.1109/TVT.2023.3246986

[6] Dai JC, Niu K, Lin JR. Polar-Coded MIMO Systems. IEEE Transactions on Vehicular Technology.2018; 67(7):6170-84. https://doi.org/10.1109/TVT.2018.2815602

[7] Dai JC, Niu K, Si ZW, Zhang DX. Polar-Coded Spatial Modulation. IEEE Transactions on Signal Processing. 2021; 69:2203-17. https://doi.org/10.1109/TSP.2021.3068848

[8] Egilmez ZBK, Xiang LP, Maunder RG, Hanzo L. A Soft-Input Soft-Output Polar Decoding Algorithm for Turbo-Detection in MIMO-Aided 5G New Radio. IEEE Transactions on Vehicular Technology. 2022; 71(6):6454-68. https://doi.org/10.1109/TVT.2022.3163288

[9] Gao J, Zhang DX, Dai JC, Niu K, Dong C. ResNet-Like Belief-Propagation Decoding for Polar Codes. IEEE Wireless Communications Letters. 2021; 10(5):934-7. https://doi.org/10.1109/LWC.2021.3050819

[10] Qiu QJ, Liu JD, Hao MQ, Li WJ, Wang Y, Tao LF, Wu L, Xie Z. DCKH-CNN: A multimetric graph-based convolutional neural network for identifying key influential nodes in Earth surface data linked networks. Trans GIS. 2025 Apr;29(2): e70016. https://doi.org/10.1111/tgis.70016

[11] Song BX, Feng YX, Wang Y. DIR-Net: Deep Residual Polar Decoding Network Based on Information Refinement. Entropy. 2022; 24(12):18. https://doi.org/10.3390/e24121809

[12] Yu QP, Zhang Y, Shi ZP, Li XW, Wang LY, Zeng M. DNN Aided Joint Source-Channel Decoding Scheme for Polar Codes. IEICE Trans Fundam Electron Commun Comput Sci. 2024; E107A(5):845–849. https://doi.org/10.1587/transfun.2023EAL2068

[13] Agarwal A, Mehta SN. PC-CC: An advancement in forward error correction using polar and convolutional codes for MIMO-OFDM system. J King Saud Univ-Comput Inf Sci. 2020; 32(8):917–927. https://doi.org/10.1016/j.jksuci.2017.12.003

[14] Bian CH, Hsu CW, Lee CW, Kim HS. Learning-Based Near-Orthogonal Superposition Code for MIMO Short Message Transmission. IEEE Trans Commun. 2023; 71(9):5108–5123. https://doi.org/10.1109/TCOMM.2023.3274158

[15] Cao S, Zheng H, Lin T, Zhang SQ, Xu SG. An Unfolded Pipelined Polar Decoder with Hybrid Number Representations for Multi-User MIMO Systems. IEEE Trans Circuits Syst II Exp Briefs. 2020; 67(11):2472–2476. https://doi.org/10.1109/TCSII.2020.2964851

[16] Jalali A, Ding Z. Joint Detection and Decoding of Polar Coded 5G Control Channels. IEEE Trans Wireless Commun. 2020; 19(3):2066–2078. https://doi.org/10.1109/TWC.2019.2962113

[17] Li J, Zhou LJ, Li ZQ, Gao WD, Ji R, Zhu JT, Liu ZY. Deep Learning-Assisted Adaptive Dynamic-SCLF Decoding of Polar Codes. IEEE Trans Cogn Commun Netw. 2024; 10(3):836–851. https://doi.org/10.1109/TCCN.2024.3349450

[18] Vawda MI, Lottering R, Mutanga O, Peerbhay K, Sibanda M. Comparing the utility of artificial neural networks (ANN) and convolutional neural networks (CNN) on Sentinel-2 MSI to estimate dry season aboveground grass biomass. Sustainability. 2024; 16(3):1051. https://doi.org/10.3390/su16031051

[19] Liu YT, Shen YF, Zhou WY, Tan XS, You XH, Zhang C. Iterative EP Detection and Decoding of Polar-Coded MIMO Systems. IEEE Commun Lett. 2023; 27(4):1075–1079. https://doi.org/10.1109/LCOMM.2023.3238893

[20] Meenalakshmi M, Chaturvedi S, Dwivedi VK. Enhancing channel estimation accuracy in polar-coded MIMO-OFDM systems via CNN with 5G channel models. AEU-Int J Electron Commun. 2024; 173:8. https://doi.org/10.1016/j.aeue.2023.155016

[21] Miloslavskaya V, Li YH, Vucetic B. Neural Network-Based Adaptive Polar Coding. IEEE Trans Commun. 2024; 72(4):1881–1894. https://doi.org/10.1109/TCOMM.2023.3341838

[22] Piao J, Niu K, Dai JC, Hanzo L. Polar-Precoding: a Unitary Finite-Feedback Transmit Precoder for Polar-Coded MIMO Systems. IEEE Trans Veh Technol. 2021; 70(11):12203–12218. https://doi.org/10.1109/TVT.2021.3113324

[23] Shen YF, Zhou WY, Huang YM, Zhang ZC, You XH, Zhang C. Fast Iterative Soft-Output List Decoding of Polar Codes. IEEE Trans Signal Process. 2022; 70:1361–1376. https://doi.org/10.1109/TSP.2022.3150962

[24] Wang XM, Li J, Wu ZT, He JL, Zhang Y, Shan L. Improved NSC decoding algorithm for polar codes based on multi-in-one neural network. Comput Electr Eng. 2020; 86:106720. https://doi.org/10.1016/j.compeleceng.2020.106758

[25] Watanabe K, Kojima S, Akao T, Katsuno M, Maruta K, Ahn CJ. Modified Pilot Selection for Channel Estimation of Systematic Polar Coded MIMO-OFDM. ICT Express. 2019; 5(4):276–279. https://doi.org/10.1016/j.icte.2019.03.002

[26] Xiang LP, Liu YS, Egilmez ZBK, Maunder RG, Yang LL, Hanzo L. Soft List Decoding of Polar Codes.

IEEE Trans Veh Technol. 2020; 69(11):13921–13926. https://doi.org/10.1109/TVT.2020.3021258

[27] Zhou HY, Zheng J, Yang MH, Gross WJ, You XH, Zhang C. Low-Complexity Sphere Decoding for Polar-Coded MIMO Systems. IEEE Trans Veh Technol. 2023; 72(5):6810–6815. https://doi.org/10.1109/TVT.2022.3229557

[28] Zhou LX, Chan STY, Zhang MX, Kim S. A Fast Computing Decoder for Polar Codes with a Neural Network. ICT Express. 2023; 9(6):1001–1006. https://doi.org/10.1016/j.icte.2023.02.005

[29] Zhou LX, Zhang MX, Chan S, Kim S. Review and Evaluation of Belief Propagation Decoders for Polar Codes. Symmetry (Basel). 2022; 14(12):15. https://doi.org/10.3390/sym14122633

[30] Zhu HF, Cao ZW, Zhao YP, Li D. Learning to Denoise and Decode: a Novel Residual Neural Network Decoder for Polar Codes. IEEE Trans Veh Technol. 2020; 69(8):8725–8738. https://doi.org/10.1109/TVT.2020.3000345

Multi-task Perception Network for Autonomous Driving Based on CSPDarkNet and Attention Mechanism

Yelin Weng
College of Intelligent Engineering Technology, Jiangsu Vocational College of Finance & Economics
Huaian 223001, China
E-mail: wengyelin11@126.com

Keywords: CSPDarkNet, attention mechanism, multi-task perception, autonomous driving, feature decoupling fusion module

Received: March 6, 2025

Aiming at the problems of environmental perception accuracy and multi-task collaborative processing in the process of autonomous driving, this paper proposes an Autonomous Driving Multi-Task (ADMT) perception model based on Cross Stage Partial DarkNet (CSPDarkNet) and attention mechanism. This model shares features and combines information by creating a feature decoupling and fusion module and optimizing the loss function design. This reduces the complexity of training and improves the collaborative effect among tasks. Dynamically weighting the spatiotemporal features of different tasks improves the performance of target detection, instance segmentation, and target tracking. The experimental results showed that, compared with YOLOv4 and YOLOv5, ADMT has achieved significant performance improvements on the KITTI and Cityscapes datasets. Among them, on the KITTI dataset, the F1score of the model reached 0.94, and APiou=0.5 was 0.92. The corresponding values of YOLOv4 and YOLOv5 were 0.93 and 0.91, respectively. These results indicate that ADMT effectively enhances the accuracy and efficiency of target recognition for autonomous driving systems in complex environments, providing strong technical support for future intelligent transportation systems.

Povzetek: Predstavljen je ADMT, model na osnovi CSPDarkNet in mehanizmov pozornosti, ki z razgradnjo in fuzijo značilk doseže bolj kvalitetno zaznavanje, segmentacijo in sledenje kot YOLOv4/YOLOv5.

1 Introduction

With the rapid development of artificial intelligence and machine learning technologies, Autonomous Driving (AD) technology has become one of the hotspots in global transportation research. AD is not only expected to improve traffic efficiency and reduce traffic accidents but also may completely change people's travel methods. Therefore, ensuring the safety and reliability of AD systems becomes particularly crucial. Environmental perception, as the core link of AD, involves vehicles obtaining information about the surrounding environment through various sensors (such as cameras, lidars, ultrasonic sensors, etc.) and conducting real-time analysis and processing [1]. This process aims to quickly identify important information such as surrounding obstacles, traffic signals, and pedestrians to assist vehicles in making safe decisions. However, environmental perception faces many challenges, including dynamically changing traffic scenes, complex climatic conditions, and the interference caused by light changes to visual sensors, etc. [2-3]. These factors pose higher requirements for the accuracy and robustness of sensing systems, especially when multiple sensing tasks must be processed simultaneously. Researchers have proposed the Multi-Task Perception Network (MTPN) to address these challenges. The goal of the MTPN is to implement the simultaneous processing of multiple perception tasks, such as object detection, semantic segmentation, and depth estimation, by constructing a unified network framework [4-5]. MTPN can share the feature extraction process, thereby reducing the consumption of computing resources and improving the collaborative effect among tasks [6-7]. The existing MTPN designs an independent loss function for each segmentation task, which affects the generalization ability of the model and also requires a lot of time to create and adjust the loss function and its parameters. Therefore, it is necessary to shorten the time for processing multiple tasks and optimize the loss function. The Cross-Stage Partial DarkNet (CSPDarkNet) is an improved network structure based on the deep learning DarkNet framework. It effectively enhances the feature extraction ability of the network through channel segmentation and aggregation. The Attention Mechanism (AM) simulates the focusing characteristics of human visual attention, which helps deep learning networks identify key features more accurately. In MTPN, AM helps the network identify and focus on targets in complex environments more accurately, improving perception accuracy [8]. Therefore, the research aims to clearly evaluate the performance of the multi-task perception system in AD, and how to effectively improve the accuracy and efficiency of multiple tasks such as target detection, instance segmentation, and target tracking in complex traffic environments. Meanwhile, the research explores how to

solve the problem of information interference in multitasking and optimize the loss function to improve the overall performance and reliability of the model. Against this background, the study proposes the Autonomous Driving Multi-Task (ADMT) perception model. The innovation point of this model lies in the use of the feature decoupling and fusion module to integrate features from different tasks, thereby reducing the mutual interference among various tasks. A new fusion loss function strategy is proposed. By comprehensively considering the correlations among various tasks, the sharing of the loss function is achieved. In the feature extraction stage, an adaptive attention module is introduced. ADMT can effectively suppress irrelevant background information in complex environments and enhance the recognition ability of small targets and occluded targets.

The research content mainly has four sections. Section 1 discusses the problems of AD and the related research results of the MTPN. Section 2 designs the ADMT. Section 3 analyzes the effectiveness of research methods. Section 4 discusses and summarizes the entire text.

2 Related works

AD technology is extensively utilized in transportation. To improve the effectiveness of AD, researchers have proposed some methods and strategies. Khan M A et al. proposed a building block technology to improve the level of vehicle driving automation. They discussed technologies and concepts such as sensors, mobile edge computing, machine learning, data analysis, and distributed learning, and mapped their roles to end-to-end solutions. The case analysis showed that this technology was feasible and provided ideas for different solutions for 5-level autonomy [9]. Chen L et al. proposed a parallel driving operating system based on parallel driving theory to address incompatibility between different AD algorithms and platforms. The system consisted of 4 structural layers: hardware layer, kernel layer, functional layer, and application layer. It could be derived into specific operating systems in 4 application scenarios: intelligent mining, warehousing, logistics, and ports. In the above testing scenarios, the parallel-driven operating system demonstrated reliability and high efficiency [10]. Li G et al. proposed a continuous decision-making method built on Deep Reinforcement Learning (DRL) to balance driving efficiency and comfort. This method utilized a convolutional neural network to map the relationship between traffic images and vehicle operation, established an end-to-end decision-making framework, and used a deep deterministic policy gradient algorithm to solve the problems in the decision-making process, obtaining the optimal driving strategy. This method provided an effective strategy for AD at intersections while balancing driving comfort, and ensuring driving safety and efficiency [11]. Wang H et al. proposed a novel anchor-free detection network and average boundary model to address occlusion issues in driving scenarios. The backbone network of this network used structural reparameterization technology to locate targets using boundary feature information. This algorithm outperformed CenterNet in both speed and precision, with an accuracy of 55.6%, meeting the requirements for speed and accuracy in driving scenarios [12]. Yang K et al. proposed a robust decision-making framework for AD on highways to improve driving safety and constructed a reinforcement learning strategy based on deep deterministic policy gradients. This strategy directly mapped observations to actions and evaluated model uncertainty based on deep deterministic strategies at runtime to quantify the reliability of the strategy and identify unknown scenarios. The proposed framework had good performance [13].

Some experts have also conducted research on MTPN. Ji Y et al. proposed a multi-task context-aware recommendation method to assist product design in a more intelligent way. By pre-processing work, multi-task knowledge requirement perception, and recommendation engine, the problem of mutual interference of contextual information in different tasks has been solved. This method outperformed traditional methods in terms of effectiveness and performance [14]. Ye H et al. proposed a sequential greedy pruning strategy to optimize the objective of global channel pruning task mismatch. They developed a performance-sensitive criterion to assess the filtersensitivity to each task and retain the filters that are globally most task-related. Experiments on multiple multitasking datasets have shown that the algorithm reduced parameters by over 60% without significant performance degradation, and achieved 1.2-3.3 times acceleration on cloud and mobile platforms [15]. Choudhry A et al. proposed a multi-tasking framework for identifying fake news and rumors on the internet. This framework trained various deep learning models in both single-task and multi-task settings for more comprehensive comparisons. Whether in domain or cross domain settings, this model consistently outperformed single-task models in accuracy, precision, recall, and other aspects [16]. Yang E et al. proposed a task adaptive learning rate method to balance different tasks on each parameter. This method measured the task dominance of a parameter by overall updates made to that parameter by each task, to separate the cumulative gradient in the adaptive learning rate method. Comprehensive experiments on recommendation system datasets have shown that this method optimizes the performance of the dominant task [17]. Pei Y et al. proposed a multi-task DRL method for controlling voltage regulation in distribution systems through photovoltaic intelligent inverters. This method encoded the topology as an additional state for multi-task DRL and utilized a multi-task learning scheme to jointly learn all task control strategies. The comparison conducted on the improved node system showed that this method had good robustness [18].

Xudong Yu et al. proposed an intelligent driving simulation test platform based on a six-degree-of-freedom motion platform for the safety and comfort of AD. This platform included four parts: the motion platform, the ultra-fast data acquisition system, the visual simulation and projection system, and the real-time simulation system. It had A fast response, wide-angle range, and

excellent driving ability, capable of simulating various dynamic driving scenarios. The research results showed that in the absence of monitoring requests, the time it took for drivers to return their palms to the steering wheel was significantly longer than in the situation with a request strategy. Monitoring requests effectively improved the takeover efficiency. Meanwhile, different prompt sounds also had an impact on the driver's takeover performance [19]. Yang T et al. proposed a design method based on the convex program for the Cooperative Adaptive Cruise Control (CACC) problem of Connected and Automated Vehicles (CAVs), aiming to synthesize distributed attack monitors and H∞ CACC controllers. The goal of this method was to minimize the combined impact of covert pseudo-data injection attacks and system interference on fleet dynamics while ensuring that designated trains adhere to performance [20]. Wang X et al. proposed a trajectory prediction model named Multi-Dimensional Spatio-Temporal Feature Fusion (MDSTF) for the problem of trajectory prediction of traffic participants in AD, aiming to accurately capture complex spatiotemporal features. The experimental results on the ApolloScape trajectory dataset showed that this method outperformed other advanced methods in the Weighted Average Displacement Error (WSADE) and Weighted Final Displacement Error (WSFDE) metrics, reducing the errors by 4.37% and 6.23% respectively compared to the best benchmark model S2TNet [21].

In summary, although many scholars have designed some models to enhance the recognition accuracy of AD, the existing models have low recognition accuracy. In view of this, this study attempts to use CSPDarkNet and AM to construct ADMT, providing certain technical support for improving the security of AD and the accuracy of target recognition.

3 Design of ADMT

3.1 Construction of CSPDarknet structure

During the process of AD, vehicles need to detect and recognize surrounding targets. In small target detection in remote sensing images, the YOLOv4 algorithm cannot achieve very accurate detection results [22]. CSPDarkNet is the core Feature Extraction Network (FEN) of YOLOv4. To enhance the extraction of small targets, the paper adds a backbone network to the Context Converter (Cot) module of YOLOv4 and constructs a Cot-CSPDarknet. The Cot module is shown in Figure 1.

In Figure 1, the Cot module is generated by "query", "key", and "value" through three independent 1×1 convolutional layers of the input feature map A. The parameters of the convolutional layers are not shared to learn different context representations. The convolution kernel size of each branch is 1×1, the stride is 1, and the fill is 0, aiming to extract task-related features through channel dimension reduction while maintaining the spatial resolution unchanged. The "key" further encodes the Static Context (SC) through 3×3 convolution kernels to generate the SC, which is designed to capture the local spatial correlation. Subsequently, the attention weight matrix is generated through two 3×3 convolution operations and multiplied by the "values" to obtain the Dynamic Context (DC). The expression for the attention matrix is shown in equation (1).

$$A=[K_1,Q]W\tau W\phi \quad (1)$$

In equation (1), A is the attention matrix. K_1 is an SC. $W\phi$ and $W\tau$ represent convolutions with and without sigmoid activation functions. The DC formula for input is shown in equation (2).

$$K_2 = V \times A \quad (2)$$

In equation (2), K_2 is the DC. The fusion of SC and DC can serve as the Cot module's output. The most basic structure in Cot-CSPDarknet is the Cot module. The convolution kernel in Layer 1 can lower the dimensionality of the Feature Matrix (FM) and decrease the number of parameters. The Layer 2's Cot module combines the residual structure in Cot-CSPDarknet to optimize the extraction of contextual feature information

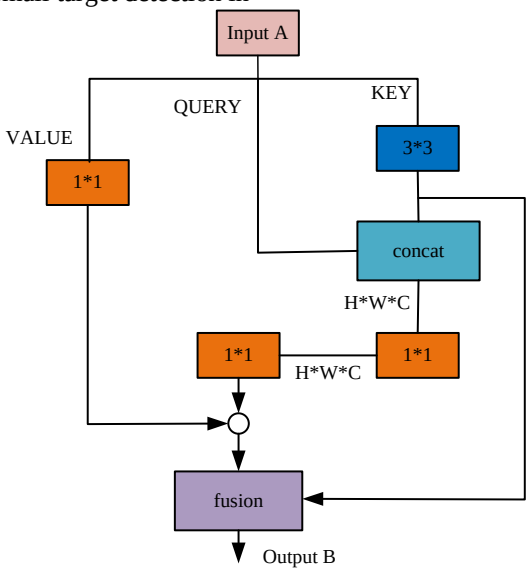


Figure 1: Cot module diagram.

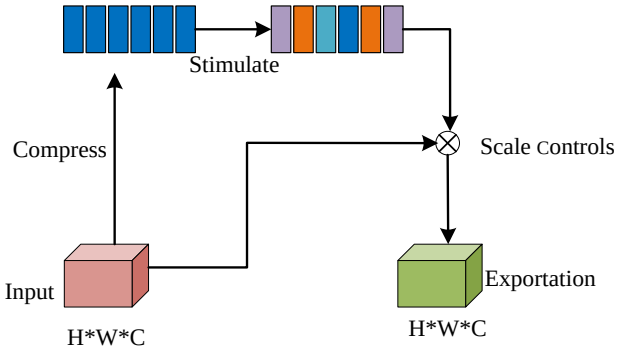


Figure 2: SE module network structure.

near small targets. The function of the 1×1 convolution kernel in Layer 3 is to increase the dimensionality of the FM and double the number of channels in the FM. The number of channels of the Cot module is determined through the combination of empirical dimensionality reduction and experimental tuning. To balance the computational efficiency and the feature expression ability, the number of input feature map channels C is uniformly dimensionally reduced to C/4 through 1×1 convolution (for example, 64 channels are output when C=256). To pay more attention to small objects in remote sensing, this study uses Squeeze-and-Excitation (SE) Attention module in the early stage of FEN downsampling operation. The SE module achieves selective attention to features through two main steps: compression and excitation. In the compression step, the SE module first performs global average pooling on the input feature map to obtain the global feature description of each channel. In the excitation step, the SE module learns the interrelationships between channels through the fully connected layer and generates a channel weight vector. Finally, the SE module multiplies the channel weight vectors obtained through the above learning by the original feature map and re-weights the input features, thereby enhancing the important features and suppressing unnecessary shallow features. This process ensures that the model can focus more on the information useful for the actual tasks when dealing with downstream tasks. The reason for researching the supplementary SE module is that the model can enhance the emphasis on key channels and effectively filter out shallow features irrelevant to the task. The SE module accelerates the convergence of the network by reducing information redundancy and reduces the computational burden during the training process. The addition of the SE module promotes feature sharing among multiple tasks. It enables each task to more effectively apply the learned feature information within the model to different tasks, such as object detection, instance segmentation, object tracking, etc., thereby improving overall performance. Figure 2 shows the SE structure.

In Figure 2, the SE includes compression and excitation operations, which optimize the network's expressive power by explicitly modeling the interdependence between feature map channels. The SE first compresses the features from input U . This operation utilizes the spatial dimension M×N to generate channel features, then performs incentive operations to learn the connections between each channel, and finally multiplies the feature U to generate the output of the SE, as shown in equation (3).

$$\begin{cases} s = F_{ex}(z,V) = \sigma\left(g(z,V)\right) = \sigma\left(V_2 \kappa(V_1, z)\right) \\ Z_c = F_{sq}\left(u_c\right) = (1/(M \times N)) \cdot \sum_{(i=1)}^{M} \sum_{(j=1)}^{N} u_c(i,j) \\ \overline{x}_c = F_{scale}\left(u_c, s_c\right) = u_c \times s_c \end{cases} \tag{3}$$

In equation (3), F_{sq} is the compression operation. F_{ex} is an incentive operation. u_c is a feature from U . κ is the ReLU activation function. F_{scale} is the feature of multiplying the activation values of each channel learned through compression and excitation operations by U . $\overline{x}_c$ is the result of compression and excitation operations on input U . The sub-pixel convolution operation is achieved by inserting a rearrangement step after the convolution layer. In this process, a feature map of a higher dimension is first generated using a convolutional layer, and then the channel information of the feature map is reorganized into a higher-resolution output through a rearrangement operation. The magnification factor adopted is 2, which means that the width and height of the output image are both twice that of the input feature map. The transformation relationship of sub-pixel convolution is shown in equation (4).

$$I^{LR}_{xlr,ylr,c \times r^2} \rightarrow I^{SR}_{x,y,c} \tag{4}$$

In equation (4), I^{LR} and I^{SR} are feature maps with low resolution and high resolution. x and y both represent the size of the feature map. c is the amount of channels in the feature map. By converting low-resolution feature maps into high-resolution one, small targets can be better detected. The fully convolutional structure is adopted, and two convolutional layers are used to change the number of channels and accommodate the feature data of the network model. Two consecutive convolution operations are used to achieve classification and regression of detection targets. Classification is used to predict the confidence level of each anchor, and regression

is used to predict the offset between the anchor and the bounding box at each location. The expression for calculating the predicted box position based on the offset is shown in equation (5).

$$\begin{cases} b_x = \sigma(t_x) + c_x \\ b_y = \sigma(t_y) + c_y \\ b_w = p_w \times e^{t_w} \\ b_h = p_h \times e^{t_h} \end{cases} \tag{5}$$

In equation (5), c_x and c_y are the coordinates of the points on the current grid. p_w and p_h are the width and height of the anchors. b_x and b_y are the adjusted center coordinates of the predicted box. b_w and b_h are the width and height of the adjusted prediction box. To improve the performance of the prediction branch, this study introduces a residual module to supersede the first convolutional layer of the prediction branch with a ResBlock, and optimizes the feature maps input for candidate box regression and classification tasks to achieve the performance of the prediction branch. The network structure of Cot-CSPDarknet feature extraction is shown in Figure 3.

In Figure 3, this study first optimizes the backbone network by introducing the Cot module that focuses more on contextual information, thereby constructing a new Cot-CSPDarknet. CSPDarkNet is an improved deep learning network architecture derived from the original DarkNet framework, specifically designed to enhance feature extraction capabilities. It divides the feature mapping into multiple parts through the Cross-Stage Partial Network (CSP) technology and conducts different feature learning in each part, thereby enhancing the model's ability to capture complex features. To more effectively utilize shallow feature information, an AM-SE module is introduced in the feature fusion stage after adding features extracted by downsampling at different magnifications. This helps reduce interference from irrelevant features such as background. This study utilizes clustering algorithms to regenerate prior boxes that are more suitable for small targets in the dataset. To further improve the accuracy of upsampling, the original nearest neighbor interpolation method has been replaced with sub-pixel convolution. Finally, to enhance the performance of the prediction layer, the first convolutional block of the prediction layer is used to replace the residual unit.

In the above content, the overall structure of the ADMT model as an MTPN is introduced, emphasizing that its design purpose is to achieve multiple perception tasks simultaneously through a unified network structure. These tasks include object detection, semantic segmentation, depth estimation, etc. In the complex AD environment, these tasks are interdependent and require comprehensive consideration by the network to improve the accuracy and efficiency of perception. By sharing the feature extraction process, the model can effectively reduce the consumption of computing resources and improve the collaborative effect among various tasks. After constructing the overall framework of the ADMT model, the next step will be to explore its internal structure and modules in detail, in order to achieve collaborative processing of multiple tasks. Especially the feature decoupling and fusion module, and how to effectively allocate feature weights for subtasks such as object detection, instance segmentation, and object tracking.

3.2 Environment perception decoupling fusion algorithm based on AM

Cot-CSPDarknet can enhance the recognition of small targets, but it cannot solve the problem of mutual interference between tasks in multitasking environments. Cot-CSPDarkNet cannot effectively solve the problem of mutual interference among tasks in a multi-task environment. The main reason is that the feature sharing strategy adopted may result in different focus on feature information for each task, leading to the information of some tasks being overwhelmed by other tasks. Furthermore, the gradient influence of the loss functions of different tasks on the shared features can lead to interference in the optimization process, making it difficult for the model to balance the learning requirements of multiple tasks. Finally, the lack of feature decoupling ability limits the adaptability of the model in complex environments and reduces the efficiency of full utilization of information among tasks. AM can reduce information overload by focusing on key information, enabling neural networks to process input data more effectively and improve task processing efficiency and accuracy [23]. To solve the multitasking problem in AD, this study proposes an environment aware decoupling fusion algorithm based on AM, and its module structure is displayed in Figure 4.

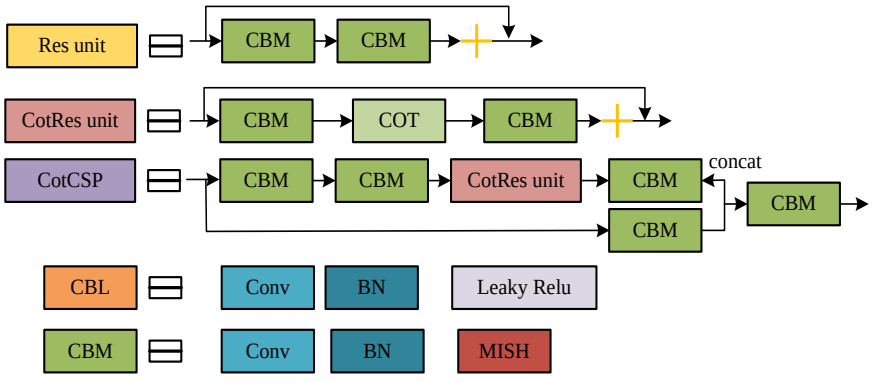


Figure 3: Cot module and its convolutional layer structure.

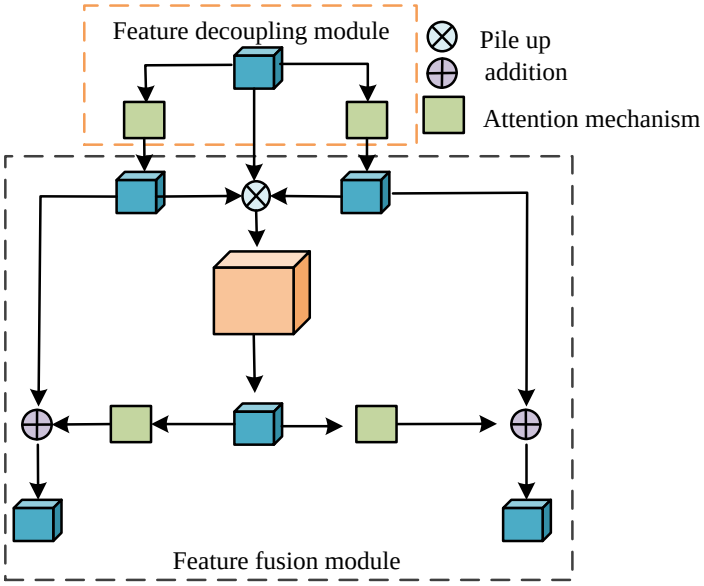


Figure 4: Block diagram of feature decoupling and fusion module.

In Figure 4, the spatiotemporal features are input into the feature decoupling module and weighted through AM to obtain the features of the detection object. To capture the intrinsic connections between features, it is necessary to use feature pools. The feature pool is the fused spatiotemporal features, and AM is applied during feature selection. Then, decoupled features are fused to obtain features suitable for various tasks. This study employs Efficient Channel Attention (ECA). ECA proposes a cross channel interaction method that does not require dimensionality reduction, and replaces the fully connected layer in the SE with a 1D convolutional structure. This not only simplifies the model structure but also improves network performance. The ECA module uses the softmax function for feature mapping, and its calculation is shown in equation (6).

$$R = \text{softmax}\left(W_2 \,\text{ReLU}\left(W_1 y\right)\right) \tag{6}$$

In equation (6), R is the feature map. W_1 and W_2 are hyperparameters. y is the result of global average pooling of image features. ReLU is the activation function. The flowchart of the feature decoupling module is exhibited in Figure 5.

In Figure 5, after being processed by a dual AM, spatiotemporal features are weighted for application in detecting objects. The features processed by the decoupling fusion unit are assigned different weights and then sent to three fully connected networks to perform specific sub-tasks. When performing feature decoupling, the object detection network first performs heatmap prediction to determine the approximate position, and then estimates the offset to lift the localization accuracy. Finally, the target size is estimated based on the model to collect detailed information about the target, ensuring comprehensive and accurate object detection. The instance segmentation module uses the offset calculated by the extraction module for data matching, and finally generates the predicted target trajectory. To balance the training speed of multiple tasks, this study uses dynamic weighted averaging to optimize the network as a whole. "Dynamic weighted averaging" is achieved by assigning a dynamic weight to each input feature, and these weights are adjusted according to the importance of the feature and the context information. Firstly, the initial weights of each feature are calculated, and then these weights are dynamically updated through learning mechanisms such as neural networks or AM to reflect the relative importance of specific tasks or data moments. Ultimately, the calculation process of the dynamic weighted average is to multiply the eigenvalues by their corresponding weights and then sum them up. Finally, it is normalized and divided by the sum of weights to obtain an accurate average value. The calculation of minimizing the loss function is shown in equation (7).

$$l_{total} = \varpi_t\left(t\right)\cdot l_{td} + \varpi_{is}\left(t\right)\cdot l_{is} + \varpi_{tf}\left(t\right)\cdot l_{tf} \tag{7}$$

In equation (7), l_{total} is the total loss function. The total loss function summarizes the losses of the model on different tasks and is used to evaluate the overall performance of the model, ensuring that all tasks receive reasonable attention and optimization. l_{td}, l_{is}, and l_{tf} are the loss function for object detection, instance segmentation, and target tracking. The object detection loss function measures the model's ability to recognize objects in an image, typically including the accuracy of the detection box and the prediction accuracy of the corresponding object category. The instance segmentation loss function evaluates the ability of the model to distinguish different instances of the same type of target and perform precise segmentation. The target tracking loss function is used to measure the model's ability to maintain the tracking target in the video stream. $\varpi\left(t\right)$ represents the weight of the loss function at time *t*. These weights dynamically adjust the contribution of each task to the

total loss function based on the importance and performance of each task. The multitasking algorithm framework is shown in Figure 6.

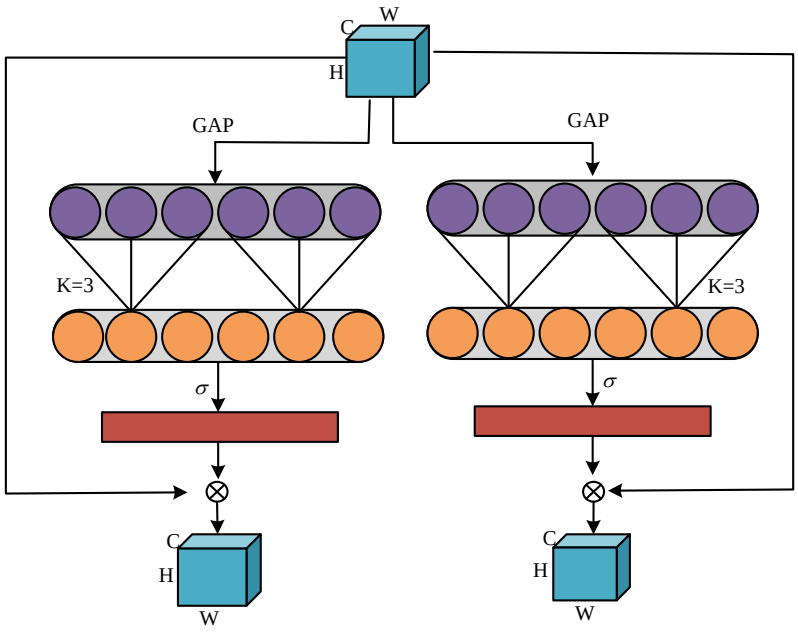


Figure 5: Feature decoupling module.

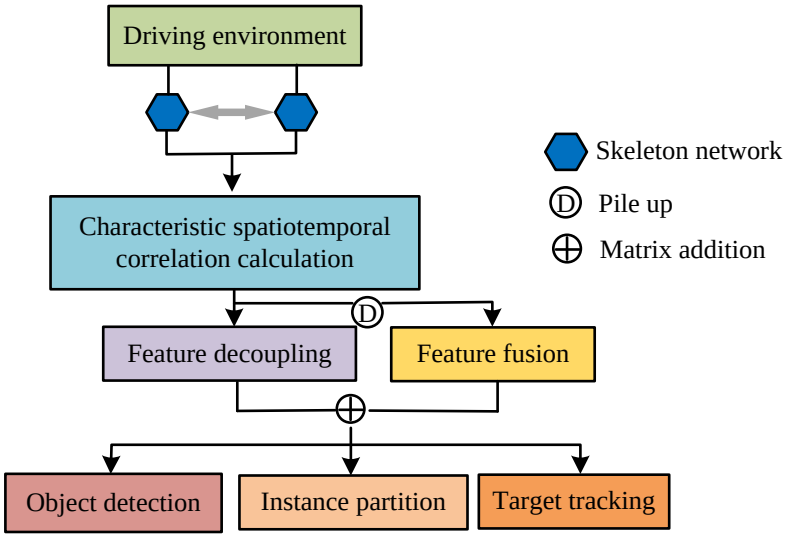


Figure 6: Multi-task learning framework.

In Figure 6, first, feature spatiotemporal correlation is performed based on the driving environment, then spatiotemporal features are decoupled and fused, and finally three tasks are performed: object detection, instance segmentation, and object tracking. The corresponding weights of the loss functions for these three tasks are shown in equation (8).

$$\varpi_i(t)=\frac{\mathrm{N}\cdot\exp\left(r_i(t-1)/T\right)}{\sum_n \exp\left(r_n(t-1)/T\right)} \tag{8}$$

In equation (8), r_i is the training speed, i is the task, $(t-1)$ is a certain moment, and N is the number of tasks. T stands for temperature scaling. Its value can adjust the competition intensity among tasks. A smaller value will enhance the difference in training speed and accelerate the convergence of simple tasks. Larger values alleviate the imbalance of learning rates among tasks and prevent certain tasks from being completely ignored. The dynamic weighted average method is achieved by evaluating the weight of each task's loss function, with a lower weight indicating a faster training process for that task. The calculation process of training speed is expressed in equation (9).

$$r_n(t-1)=\frac{l_n(t-1)}{l_n(t-2)} \tag{9}$$

In equation (9), l_n is the corresponding loss function for a certain iteration cycle. The research adopts the design

of the Feature Decoupling Fusion Module (FDFM) based on the feature decoupling theory of multi-task learning and the collaborative optimization principle of the AM. The theoretical core lies in explicitly separating the task-specific features from the shared features, and using the local cross-channel interaction of the ECA module and the global channel recalibration of the SE module to enhance the positioning accuracy of small targets and suppress background noise respectively. The decoupling stage uses a gating mechanism to generate sparse features of the task. In the fusion stage, cross task self attention is introduced to model collaboration between tasks, and multi task losses are balanced through dynamic uncertainty weighting to avoid bias in manual parameter adjustment. This architecture alleviates task conflicts and enhances the feature saliency of occlusions and distant small targets through hierarchical feature optimization and adaptive weight allocation.

4 ADMT effectiveness analysis

4.1 Multi-task perception model performance testing

To analyze the performance of ADMT at runtime, this study uses the KITTI and Cityscapes datasets as test data. For the KITTI dataset, the main tasks include object detection and depth estimation, as this dataset is typically used for object recognition and 3D reconstruction in traffic scenes. For the Cityscape dataset, the main task is semantic segmentation, with a focus on image understanding and pixel level classification in urban environments. To optimize the hyperparameters of the model and improve its performance, the study adopts the Bayesian optimization strategy to automate the hyperparameter search process. The research systematically optimizes key hyperparameters such as the learning rate, batch size, and momentum by constructing the objective function of hyperparameters. The Bayesian optimization method uses the previous experimental results to guide the subsequent search, thereby effectively exploring the hyperparameter space. In the experiment, the study sets the initial value range of the learning rate from 0.0001 to 0.01. The batch size ranges from 16 to 64, and the momentum parameter ranges from 0.8 to 0.99. Through this strategy, the optimal combination of hyperparameters can be quickly found, resulting in the optimal setting of a learning rate of 0.001, batch size of 32, and momentum of 0.9, thereby improving the convergence speed and performance of the model. The research adopts the standard division strategies of training set, validation set, and test set to ensure the generalization ability of the model on different data. The KITTI and Cityscapes datasets are divided in the proportions of 70% for training, 15% for validation, and 15% for testing. The training set is used for the learning and optimization of the model, while the validation set regularly evaluates the model performance during the training process to achieve the best effect. The test set is used to ultimately evaluate the generalization ability and performance of the model. It will not be viewed or adjusted during the testing phase. Table 1 shows the experimental environment and equipment.

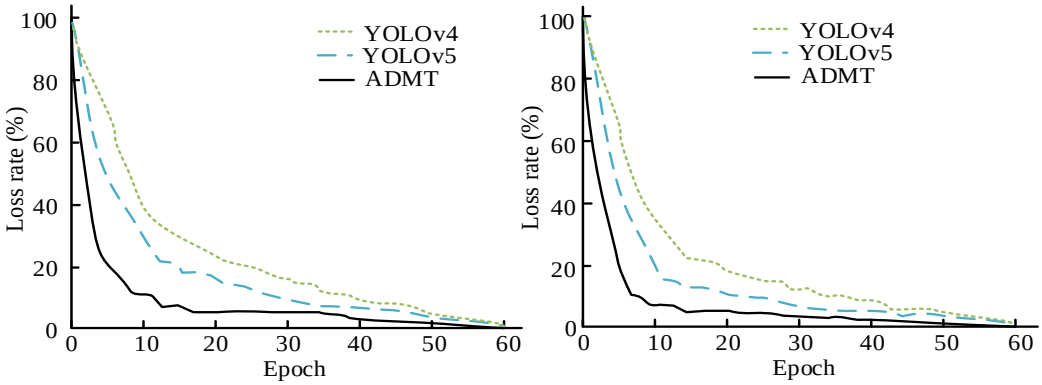
To test the performance of ADMT, this study conducts training and testing. After 60 iterations, the ADMT model is compared and analyzed with YOLOv4 [24] and YOLOv5 [25]. The experiment compares the losses of three methods under different iterations. The loss values include the KITTI dataset and the Cityscapes dataset. These losses are measured by the cross-entropy loss function. The specific calculation method is to compare the category probabilities predicted by the model with the real labels, thereby quantifying the performance of the model during the training process. The experimental results are shown in Figure 7.

Figure 7 shows the trend of loss variation with iteration times for three different methods. As the Number of Iterations (NoI) increases, the loss values of all methods show a decreasing trend, indicating that the model gradually learns and optimizes during the training process. When the NoI is relatively small, especially within 10 iterations, the loss reduction curve is steeper, indicating that in the early phases of training, the model's loss reduction rate is faster and the learning efficiency is higher. However, when the NoI exceeds 10, the curve of loss reduction becomes relatively flat, indicating that as the training progresses, the rate of loss reduction in the model gradually slows down. Until after 60 iterations, the loss value approaches 0, indicating that the model is approaching convergence. In Figure 7, at iteration times of 10 and 20, the losses of YOLOv4 and YOLOv5 are 38% and 20%, 28% and 12%, while the losses of ADMT are 10% and 5%, significantly lower than the comparison algorithm. This indicates that the ADMT model can reduce losses during iterations. This study compares the MOTA, F1 score, and APiou=0.5 scores of different algorithms on KITTI and Cityscapes, as shown in Figure 8.

Table 1: Engineering project setting conditions.

Name	Parameter
Simulation program	Matlab2020
Computer	Intel (R) Core (TM) i5-6200U CPU@ 2.30GHz
Dataset	KITTI, Cityscapes
Operating system	Ubuntu 16.04
CPU	Intel.Xeon, E5_2 660.v3
GPU	NVIDIA TESLA V100

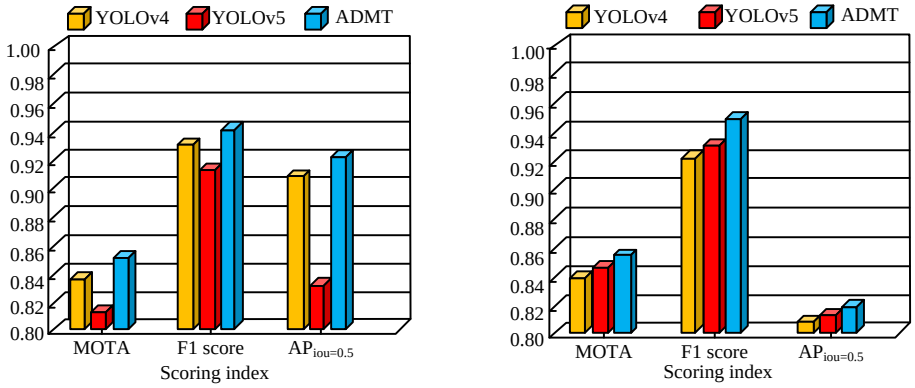
Video memory	16 GB
Deep learning framework	Tensorflow



(a) The loss of the model on the test set of the training phase

(b) The loss of the model on the test set of the Test phase

Figure 7: Loss comparison of different methods in training and testing phases.



(a) Score on KITTI data set

(b) Scores on the Cityscapes dataset

Figure 8: Comparisonof existing methods tested on KITTI and Cityscapes datasets.

Figure 8 shows the performance between the ADMT model and the other two methods on the KITTI and Cityscapes datasets. On KITTI, the MOTA, F1 score, and APiou=0.5 scores of the ADMT model are 0.85, 0.94, and 0.92. These scores are the highest among the three comparison methods, demonstrating the superior performance of the ADMT model in multi-target tracking and object detection tasks. In contrast, the corresponding scores of YOLOv4 and YOLOv5 are 0.834, 0.93, 0.904 and 0.81, 0.91, 0.83, all lower than the ADMT model. These results indicate that the ADMT model has higher accuracy and robustness in handling object detection and tracking tasks in complex road environments. On Cityscapes, the MOTA, F1 score, and APiou=0.5 scores of the ADMT model are 0.852, 0.946, and 0.818. These scores also demonstrate the good performance of the ADMT model in urban scenarios. These datasets contain complex scenes of urban streets, including targets of different sizes and shapes, as well as various occlusion and truncation situations. Therefore, these scores further validate the effectiveness and applicability of the ADMT in practical AD scenarios.

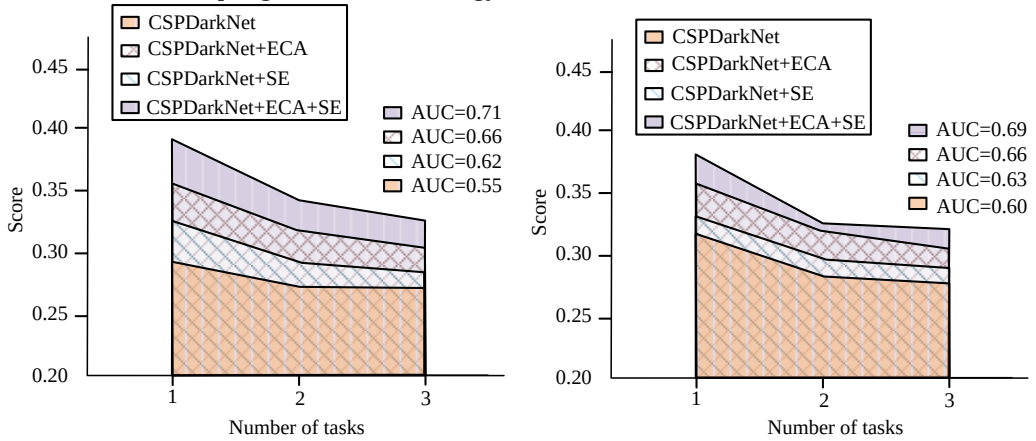
4.2 Application analysis of multi-task perception model

To validate the application effectiveness of each module used, the study conducts ablation experiments. The datasets used in the experiment are KITTI and Cityscapes datasets, as shown in Figure 9.

The ablation experiment in Figure 9 compares the independent and collaborative contributions of each component to the model performance by gradually introducing different modules into the system. The results show that when CSPDarkNet is used alone as the base network, the MOTA and F1 scores on the KITTI dataset are 0.55 and 0.60, respectively, indicating its basic feature extraction ability as a backbone network. When the ECA module is introduced, the MOTA's value increases to 0.66 (+20%), and the F1 score rises to 0.66 (+10%). This is attributed to the fact that ECA achieves cross-channel interaction through lightweight 1D convolution, optimizes the dynamic selection ability of key features, and particularly enhances the positioning accuracy of small targets. When the SE module is added alone, MOTA and F1 scores reach 0.62 (+12.7%) and 0.63 (+5%),

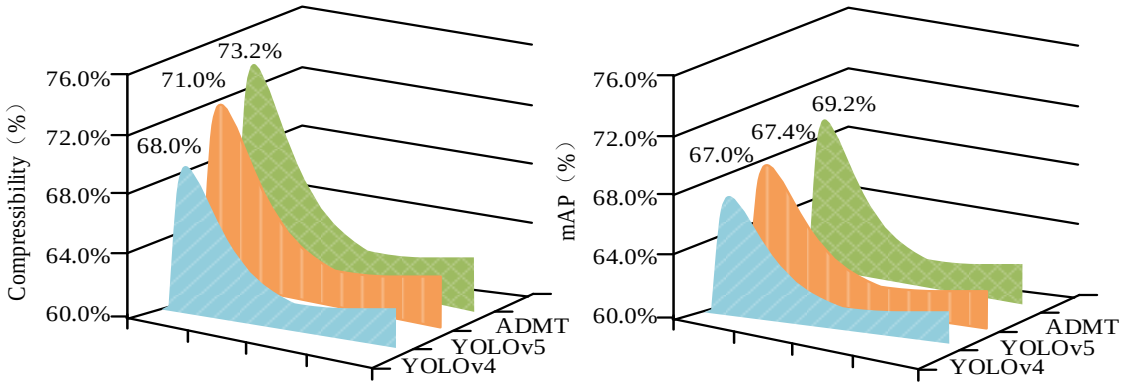
respectively. Its channel AM suppresses redundant background features by explicitly modeling the channel dependency relationship. When the ECA and SE modules are used in combination, MOTA and F1 scores further increase to 0.71 (+29.1%) and 0.69 (+15%), respectively, revealing the complementarity of the two AMs: ECA focuses on local cross-channel feature interaction, while SE enhances semantic importance through global channel re-calibration. The synergy of the two significantly improves the feature representation ability of multi-scale targets in complex scenarios. This ablation experiment not only verifies the effectiveness of each module design but also quantifies the hierarchical optimization mechanism of the AM in the feature decoupling and fusion strategy, providing an empirical basis for the interpretability of the model components. The results of comparing the compression ratio and mean Average Precision (mAP) of different methods are shown in Figure 10.

In Figure 10, the compression rate of ADMT is 73.2%, which is higher than the compression rates of 71.0% for YOLOv5 and 68.0% for YOLOv4. The mAP of ADMT is 69.2%, higher than the 67.4% of YOLOv5 and 67.0% of YOLOv4. This indicates that ADMT has good compression performance in object detection and is more stringent on target boxes detected by the network. Figure 11 shows the percentage of targets identified using ADMT on KITTI and Cityscapes.



(a) KITTI data set MOTA Score (b) KITTI data set F1 score

Figure 9: Comparative experiment of decoupling modules in KITTI and Cityscapes datasets.



(a) Compression ratio comparison of different detection methods (b) mAP comparison of different detection methods

Figure 10: Compression ratio and mAP comparison of different detection methods.

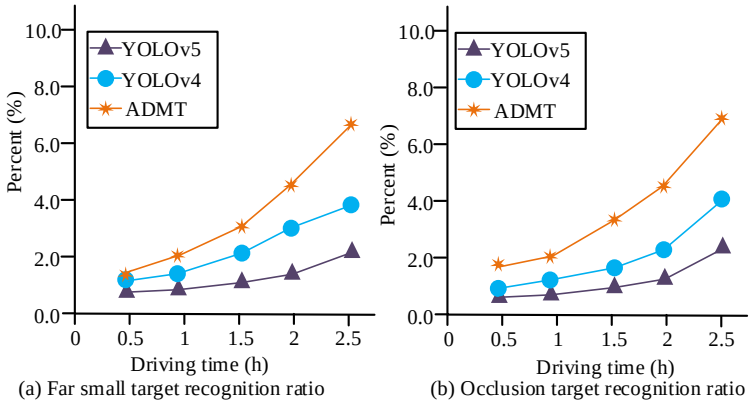


Figure 11: The proportion of targets identified at different driving times.

Figure 11 shows the comparison of the recognition percentages of far small targets and occluding targets by ADMT and YOLOv4 and YOLOv5 on the KITTI and Cityscapes datasets. The x-axis in the figure represents "Driving time (h)", which represents the cumulative testing time of the model in continuous driving scenarios, used to simulate the changes in target recognition performance of AD vehicles when driving for a long time in real road environments. Figure 11(a) shows the recognition of far small targets. The recognition rate of ADMT within a 2.5-hour driving time is 6.6%, which is higher than that of YOLOv5 (2.2%) and YOLOv4 (3.8%). This advantage stems from the enhanced extraction of context features by Cot-CSPDarkNet and the refined reconstruction ability of low-resolution features by sub-pixel convolution. Figure 11(b) shows the recognition of occlusion targets. The recognition rate of ADMT reaches 7.0%, while that of YOLOv5 and YOLOv4 is 2.4% and 4.1%, respectively. This verifies the effectiveness of the FDFM module in dynamically decoupling the occluded area from the background noise through attention weights. Meanwhile, it indicates that the synergistic effect of the ECA and SE modules can improve the fusion accuracy of local features and global semantics. Although the above results have improved the recognition accuracy of small objects and occlusions, on the whole, they are still relatively low. The main reasons lie in the deviation of data distribution, the limitations of model structure, and the boundaries of multi-task optimization. In terms of data, the proportion of far small targets and severely occluded samples in the KITTI and Cityscapes datasets is less than 5%, resulting in insufficient learning of extreme scenes by the model. The limitation of sensor resolution causes the details of small targets to be blurred. In terms of model structure, the channel compression loss of the Cot module is small for the shallow texture features of the target. The sub-pixel convolution is insufficient for the reconstruction of low-resolution features. The AM is prone to interference from similar backgrounds in densely occluding areas. In terms of task conflicts, the dynamic weighting strategy leans towards tasks with fast training speeds and ignores the refinement requirements of instance segmentation for occlusion boundaries. The visualization diagram of the research method is shown in Figure 12.

Several key elements are involved in Figure 12, including the detected target box, the corresponding category label, and the confidence score. Boxes of different colors represent the model's recognition of different types of targets, demonstrating the model's ability to handle complex traffic scenarios, such as effectively distinguishing pedestrians, vehicles, and other obstacles during dynamic driving. In addition, the instance segmentation part reflects the precise boundary division of the model on the target, thus more clearly depicting the shape and boundary of the target, which is conducive to improving the tracking accuracy of subsequent tasks. Through these visualization effects, Figure 12 not only reveals the effectiveness of ADMT but also provides important analytical basis and visualization feedback for further optimizing the model and field applications. To further verify the feasibility of the proposed method, the Waymo dataset is adopted for more in-depth verification. The Waymo dataset is a large-scale AD dataset released by Waymo to support advanced AD research. This dataset contains over 250,000 scenarios, covering various complex urban and highway environments, with a focus on traffic participants such as pedestrians, cyclists, and vehicles. The dataset provides high-quality Lidar and camera data, supporting a variety of perception tasks, including object detection, instance segmentation, and tracking. Meanwhile, the diversity of the Waymo dataset, such as different weather conditions, illumination changes, and occlusion situations, provides a good basis for the robustness test of the algorithm. The comparative methods adopted in the research are the intelligent driving simulation test platform proposed by Xudong Y U et al. [19], the CACC method based on convex programming proposed by Yang T et al. [22], and the MDSTF trajectory prediction model proposed by Wang X et al. [21]. The specific comparison results of the experiment are shown in Table 2.

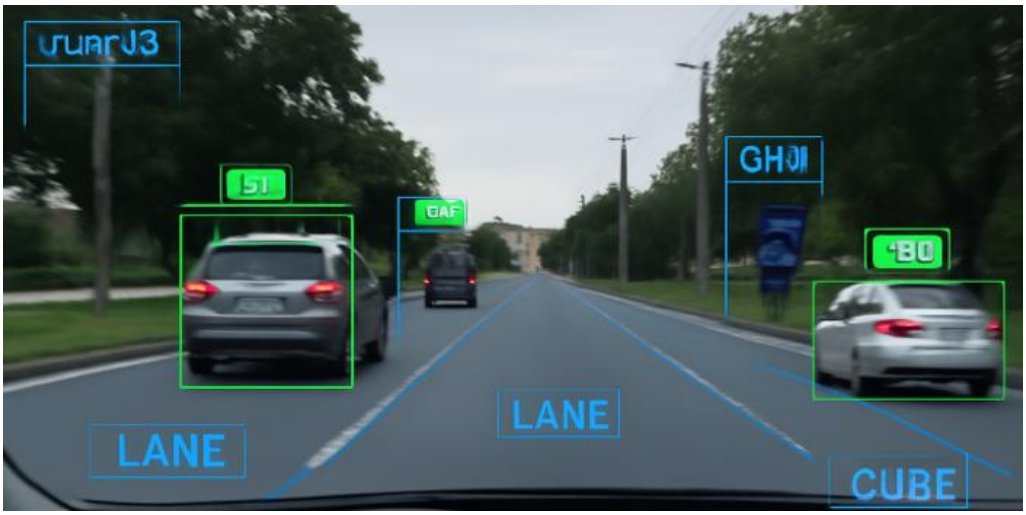


Figure 12: Study the visualization results of the model.

Table 2: Performance comparison results of different models.

Indicator	ADMT	Xudong Y U et al.[19]	Yang T et al. [20]	Wang X et al. [21]
Inference Speed (FPS)	45 FPS	30 FPS	38 FPS	35 FPS
Parameter size (MB)	10.5 MB	12.2	9.8	11.3
FLOPs	25.4 B	30.2 B	22.8 B	27.0 B
IoU	0.88	0.81	0.82	0.82
Accuracy rate of occlusion situation recognition	75.2%	60.6%	70.4%	65.4%
Recognition accuracy rate of light changes	80.5%	65.3%	72.4%	68.9%

Table 2 shows that the proposed ADMT model outperforms other comparison methods in multiple key performance indicators. Firstly, in terms of inference speed, the 45 FPS of ADMT is significantly higher than the 30 FPS of Reference [19] and the 35 FPS of Reference [21], demonstrating the advantage of ADMT in processing speed and helping to meet the real-time requirements of AD systems. Secondly, in terms of parameter size and FLOPs, the parameter count (10.5 MB) and FLOPs (25.4 B) of ADMT are more optimized compared to other methods, indicating that it is more efficient in terms of computing resource consumption. Furthermore, ADMT achieves 0.88 in the IoU index, surpassing the performance of other methods and proving its stronger ability in target detection and tracking accuracy. Especially in the task performance in complex environments, the accuracy rate of ADMT in recognizing occlusion situations (75.2%) and recognition accuracy under illumination changes (80.5%) is significantly better than that of other models. This indicates that ADMT has stronger robustness and adaptability under adverse conditions. These results indicate that the ADMT model not only performs well in standard perception tasks but also has higher reliability and accuracy when dealing with the challenges of complex driving environments.

5 Discussion

The ADMT model based on CSPDarkNet and the AM proposed in the research is significantly superior to the existing mainstream methods, such as YOLOv4 and YOLOv5, in multiple tasks. On the KITTI and Cityscapes datasets, ADMT achieved 0.94 and 0.92 respectively in performance metrics such as F1score and AP_{iou}=0.5, demonstrating its efficiency in object detection and tracking tasks and robustness in complex environments. Through the feature decoupling and fusion module, ADMT effectively and dynamically allocates and weights the features of different tasks, reduces information interference, and enhances the focus of key features of each task. This innovative design enhances the generalization ability of the model, enabling it to perform more accurately in diverse traffic situations. The introduced AM enables the model to focus on key information, improves the recognition ability of small targets and occluded targets, and enables ADMT to maintain efficient perception performance under changing environmental conditions. In similar studies, the method based on DRL proposed by Khan MA et al. has achieved good results in balancing driving efficiency and comfort. However, this method still faced challenges in the accuracy of target detection in complex environments [11]. In contrast, ADMT significantly improves the

perceptual performance in complex scenarios by comprehensively considering multi-task features and utilizing the AM. The anchor-free detection network proposed by Wang H et al. performed relatively well in the detection of occlusive objects. However, its model had a high complexity and might face the bottleneck of real-time processing in practical applications [12]. By optimizing the network structure and introducing an efficient feature decoupling mechanism, ADMT has achieved a better detection rate and response speed in the case of target occlusion, demonstrating stronger real-time processing capabilities. The AD decision-making framework proposed by Yang K et al. utilized reinforcement learning algorithms to enhance the safety of highway driving, but it lacked systematic methods in multi-task processing [13]. In contrast, ADMT demonstrates its ability to cope with different driving decision-making tasks in complex environments through centralized multi-task learning and feature fusion of task relevance, indicating the superiority of its own method. To sum up, ADMT outperforms the existing related methods in multiple performance indicators, demonstrating its broad application potential and technical value in the field of AD.

6 Conclusion

This study designed an ADMT aimed at improving the safety of AD and the precision of target recognition. This model established a FEN by integrating the Cot module, which optimized the prediction branch of YOLOv4 using the full convolution structure. In addition, the model also incorporated an AM SE module to perfect the recognition capacity of the target. To address potential conflicts between multiple tasks, the model specifically added FDFM. In the experiment, the compression rate of the model reached 73.2%, with an mAP of 69.2%. The model outperformed YOLOv5 and YOLOv4 algorithms in terms of compression rate and average accuracy. When the driving time was 2.5 hours, the proportion of ADMT recognizing far and small targets was 6.6%, and the proportion of recognizing occluded targets was 7.0%. These data indicated that the model not only had good compression performance in object detection but also had stricter requirements for target boxes detected by the network, effectively increasing the recognition of occluded truncated targets and far small targets. The results show that the Cot-CSPDarkNet constructed by the study enhances the context feature extraction ability of small targets through the context transformation module and the convolution of sub-pixels. The FDFM designed in the research combines a dual-pathway AM to dynamically separate task-specific features from shared features, alleviating multi-task conflicts. A task uncertainty weighting strategy based on temperature scaling is proposed to balance the differences in training speeds of detection, segmentation, and tracking tasks. However, ADMT still has limitations. The negative migration effect between tasks is not explicitly modeled, and the feature decoupling in extreme occlusion scenarios may fail. The dynamic weighting strategy relies on a fixed temperature parameter T and has insufficient adaptability to dynamic scenes such as sudden changes in illumination. The proportion of far small targets in the experimental dataset is relatively low, which affects the generalization ability of the model. Future work will focus on three areas. One is to introduce task association aware AM and clarify the decoupling process that constrains task features. The second is to design an adaptive temperature parameter strategy, combined with online learning to dynamically adjust the T value to adapt to complex road conditions. The third is to jointly generate adversarial networks to integrate high-density small targets and occlusion data, optimizing the robustness of the model under long tail distribution.

Fundings

The research is supported by: Research Project on Higher Education Reform in Jiangsu Province: Research and Practice of Teaching Reform for High-Quality Courses in Computer Science Majors Driven by Generative AI, (No.2025JGZD037).

References

[1] Manzoor Ahmed Khan, Hesham El-Sayed, Sumbal Malik, Muhammad Talha Zia, Muhammad Jalal Khan, Najla Alkaabi, and Henry Ignatious. Level-5 autonomous driving-are we there yet? A review of research literature. ACM Computing Surveys (CSUR), 55(2):1-38, 2022.https://doi.org/10.1145/3485767

[2] Sushila Umesh Ratre, and Bharti Joshi. Trajectory and motion prediction of autonomous vehicles driving assistant system using distributed discriminator based bi-directional long short term memory model. International Journal of Intelligent Transportation Systems Research, 23(1):245-258, 2025.https://doi.org/10.1007/s13177-024-00447-8

[3] Hebing Liu, Jinhong Sun, Heshou Wang, and Ka Wai Eric Cheng. Comprehensive analysis of adaptive soft actor-critic reinforcement learning-based control framework for autonomous driving in varied scenarios. IEEE Transactions on Transportation Electrification, 11(1):3667-3679, 2025.https://doi.org/10.1109/TTE.2024.3429186

[4] Wujie Zhou, Shaohua Dong, Jingsheng Lei, and Lu Yu. MTANet: Multitask-aware network with hierarchical multimodal fusion for RGB-T urban scene understanding. IEEE Transactions on Intelligent Vehicles, 8(1):48-58, 2022.https://doi.org/10.1109/TIV.2022.3164899

[5] Wujie Zhou, Shaohua Dong, Jingsheng Lei, and Lu Yu, MTANet: Multitask-aware network with hierarchical multimodal fusion for RGB-T urban scene understanding. IEEE Transactions on Intelligent Vehicles, 8(1):48-58, 2022.https://doi.org/10.1109/TIV.2022.3164899

[6] Menghao Guo, Tianxing Xu, Jiangjiang Liu, Zhengning Liu, Pengtao Jiang, Taijiang Mu, Songhai Zhang, Ralph R. Martin, Mingming Cheng, and Shimin Hu. Attention mechanisms in computer

vision: A survey. Computational Visual Media, 8(3):331-368, 2022.https://doi.org/10.1007/s41095-022-0271-y

[7] Tianyu Yang, Yeqiang Qian, Weihao Yan, Chunxiang Wang, and Ming Yang. AdaptiveOcc: Adaptive octree-based network for multi-camera 3D semantic occupancy prediction in autonomous driving. IEEE Transactions on Circuits and Systems for Video Technology, 35(3):2173-2187, 2025.https://doi.org/10.1109/TCSVT.2024.3492289

[8] Harshpreet Kaur, and Munish Bhatia. Digital-twin-driven performance assessment for IoT-integrated autonomous driving systems. IEEE Internet of Things Journal, 12(4):4078-4085, 2025.https://doi.org/10.1109/JIOT.2024.3481501

[9] Manzoor Ahmed Khan, Hesham El-Sayed, Sumbal Malik, Muhammad Talha Zia, Muhammad Jalal Khan, Najla Alkaabi, and Henry Ignatious. Level-5 autonomous driving-are we there yet? A review of research literature. ACM Computing Surveys (CSUR), 55(2):1-38, 2022.https://doi.org/10.1145/3485767

[10] Long Chen, Yunqing Zhang, Bin Tian, Yunfeng Ai, Dongpu Cao, and Fei-Yue Wang. Parallel driving OS: A ubiquitous operating system for autonomous driving in CPSS. IEEE Transactions on Intelligent Vehicles, 7(4):886-895, 2022.https://doi.org/10.1109/TIV.2022.3223728

[11] Guofa Li, Shenglong Li, Shen Li, and Xingda Qu. Continuous decision-making for autonomous driving at intersections using deep deterministic policy gradient. IET Intelligent Transport Systems, 16(12):1669-1681, 2022.https://doi.org/10.1049/itr2.12107

[12] Hai Wang, Yansong Xu, Zining Wang, Yingfeng Cai, Long Chen, and Yicheng Li. Centernet-auto: A multi-object visual detection algorithm for autonomous driving scenes based on improved centernet. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(3):742-752, 2023.https://doi.org/10.1109/TETCI.2023.3235381

[13] Kai Yang, Xiaolin Tang, Sen Qiu, Shufeng Jin, Zichun Wei, and Hong Wang. Towards robust decision-making for autonomous driving on highway. IEEE Transactions on Vehicular Technology, 72(9):11251-11263, 2023.https://doi.org/10.1109/TVT.2023.3268500

[14] Yongjun Ji, Zuhua Jiang, Xinyu Li, Yongwen Huang, and Fuhua Wang. A multitask context-aware approach for design lesson-learned knowledge recommendation in collaborative product design. Journal of Intelligent Manufacturing, 34(4):1615-1637, 2023.https://doi.org/10.1007/s10845-021-01889-7

[15] Hancheng Ye, Bo Zhang, Tao Chen, Jiayuan Fan, and Bin Wang. Performance-aware approximation of global channel pruning for multitask cnns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(8):10267-10284, 2023.https://doi.org/10.1109/TPAMI.2023.3260903

[16] Donggun Park, Yushin Lee, and Yong Min Kim. Effects of autonomous driving context and anthropomorphism of in-vehicle voice agents on intimacy, trust, and intention to use. International Journal of Human-Computer Interaction, 40(21/22):7179-7192, 2024.https://doi.org/10.1080/10447318.2023.2262271

[17] Enneng Yang, Junwei Pan, Ximei Wang, Haibin Yu, Li Shen, Xihua Chen, Lei Xiao, Jie Jiang, and Guibing Guo. Adatask: A task-aware adaptive learning rate approach to multi-task learning. Proceedings of the AAAI Conference on Artificial Intelligence, 37(9):10745-10753, 2023. https://doi.org/10.1609/aaai.v37i9.26275

[18] Yansong Pei, Junbo Zhao, Yiyun Yao, and Fei Ding. Multi-task reinforcement learning for distribution system voltage control with topology changes. IEEE Transactions on Smart Grid, 14(3):2481-2484, 2023.https://doi.org/10.1109/TSG.2022.3233766

[19] Xudong Yu, Guang Yin, Lie Guo, Jian Huang, and Jian Zhao. Establishment and application research of a six-degree-of-freedom autonomous driving simulation test platform. Experimental Technology and Management, 41(10):150-156, 2024.https://doi.org/10.16791/j.cnki.sjg.2024.10.019

[20] Tianci Yang, Carlos Murguia, Dragan Nešić, and Chau Yuen. Toward crash-free autonomous driving: Anomaly detection and control for resilience to stealthy sensor attacks. IEEE Internet of Things Journal, 12(1):276-287, 2025.https://doi.org/10.1109/JIOT.2024.3459590

[21] Xing Wang, Zixuan Wu, Biao Jin, Mingwei Lin, Fumin Zou, and Lyuchao Liao. Correction to: MDSTF: A multi-dimensional spatio-temporal feature fusion trajectory prediction model for autonomous driving. Complex & Intelligent Systems, 10(5):7419-7420, 2024.https://doi.org/10.1007/s40747-024-01548-3

[22] V. Arulalan, and Dhananjay Kumar. Efficient object detection and classification approach using HTYOLOV4 and M2RFO-CNN. Computer Systems Science and Engineering, 44(2):1703-1717, 2023. https://doi.org/10.32604/csse.2023.026744

[23] Zhixian Zeng, Jianjun Cao, Nianfeng Weng, Zhen Yuan, and Xu Yu. Cross-modal entity resolution for image and text integrating global and fine-grained joint attention mechanism. Journal of Shanghai Jiaotong University (Science), 28(6):728-737, 2023.https://doi.org/10.1007/s12204-022-2465-y

[24] Mingsheng Liu, Liang Wan, Bo Wang, and Tingting Wang. SE-YOLOv4: shuffle expansion YOLOv4 for pedestrian detection based on PixelShuffle. Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies, 53(15):18171-18188, 2023.https://doi.org/10.1007/s10489-023-04456-0

[25] Shanshan Wang, Weiwei Tan, Tengfei Yang, Liang Zeng, Wenguang Hou, and Quan Zhou. High-voltage

transmission line foreign object and power component defect detection based on improved YOLOv5. Journal of Electrical Engineering & Technology, 19(1):851-866, 2024.https://doi.org/10.1007/s42835-023-01625-6

Multilayer Perceptron-Based Defense Mechanisms for Securing Industrial IoT in Industry 4.0 Environments

Lahcen Idouglid*, Said Tkatek, and Khalid El Fayq
Computer Sciences Research Laboratory, Ibn Tofail University, Kenitra, Morocco
E-mail: lahcen.idouglid@uit.ac.ma, said.tkatek@uit.ac.ma, khalid.elfayq@uit.ac.ma
*Corresponding author

Keywords: industrial internet of things (iiot), intrusion detection systems (ids), multilayer perceptron (mlp), deep learning (dl), industry 4.0

Received: October 24, 2024

In the Industry 4.0 era, the Industrial Internet of Things (IIoT) has transformed manufacturing by facilitating seamless connectivity and real-time data exchange between physical devices and systems. This transformation has bolstered efficiency, productivity, and decision-making in industrial settings. However, the increased connectivity also brings heightened cybersecurity risks. Securing the IIoT environment is critical to safeguard critical infrastructure, data, and operations against cyber threats. As IIoT adoption expands across sectors, ensuring system security and resilience becomes imperative to maintain operational continuity and preserve trust. This paper proposes a deep learning-based approach, leveraging the CIDDS, BOT-IoT, and Edge_IIoTset datasets, to fortify IIoT and manufacturing systems against cyber threats. Multilayer Perceptron (MLP) is identified as the top-performing model, achieving an accuracy of 99.26%, precision of 98.74%, and recall of 98.86% on the CIDDS dataset. Similar superior performance was observed on the BOT-IoT (99.52%, 99.52%, and 99.99%) and Edge_IIoTset (99.93%, 99.93%, and 99.99%) datasets, making MLP a robust solution for anomaly detection in industrial IoT environments.

Povzetek: Članek predstavi MLP kot najučinkovitejši model za zaznavanje vdorov v IIoT, ki na treh realnih podatkovnih zbirkah preseže druge algoritme po točnosti, odzivnosti in robustnosti.

1 Introduction

The Industrial Internet of Things (IIoT) is central to Industry 4.0, driving the transformation of industrial processes through advanced digital technologies. The Industrial Internet of Things (IIoT) has transformed manufacturing by enabling low-latency communication and real-time data exchange between interconnected devices and systems. It encompasses interconnected sensors, devices, and instruments across sectors like manufacturing, energy, and healthcare [1], [2]. IIoT boosts productivity, efficiency, and sustainability while opening new business avenues[1], [3].

However, IIoT's extensive integration also invites significant security risks. Cyberattacks threaten data integrity, confidentiality, and service availability[4]. The dynamic and diverse IIoT landscape complicates traditional security methods reliant on predefined rules[5], [6]. Advanced security solutions are urgently needed to counter evolving cyber threats[7], [8], [9].

The Industrial Internet of Things (IIoT) orchestrates intelligent devices spanning various industrial domains, collecting and processing data. Its architecture, comprising perception, network, and processing layers, faces unique security hurdles, demanding robust Intrusion Detection Systems (IDS) [10], [11].

Embedding Machine Learning (ML) within IDS elevates detection accuracy, flexibility, and scalability. ML enables IDS to learn from data, identify crucial features, and construct models adept at recognizing both known and unknown threats[12], [13]. Furthermore, ML facilitates IDS adaptation to the intricacies of Fog/Edge computing, a burgeoning approach bringing computation closer to IIoT data sources[14], [15], [16].

The figure 1 shows An Exploratory Visualization of IIoT. Here are the key contributions of the paper summarized:

- Proposes a deep learning-based approach to strengthen Industrial IoT (IIoT) and manufacturing systems against cyberattacks.
- Highlights the importance of data-driven solutions for evolving security threats.
- Leverages insights from real-world data by analyzing benchmark datasets like CIDDS, BOT-IoT, and Edge_IIoTset.
- Identifies Multilayer Perceptron (MLP) as the most effective deep learning algorithm for this application.

Figure 1 provides an exploratory visualization of the Industrial Internet of Things (IIoT), capturing key

elements of the interconnected infrastructure and their relationship with emerging security challenges.

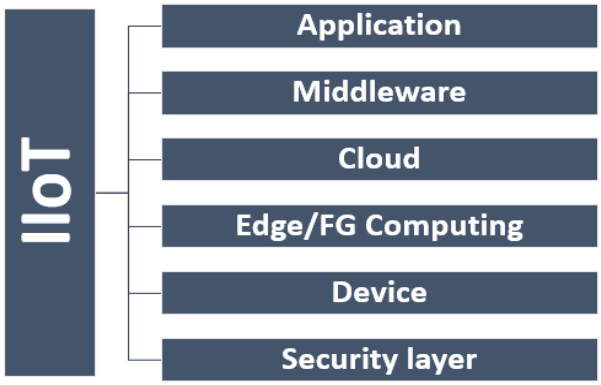


Figure 1: An exploratory visualization of the industrial internet of things (IIoT)

The remainder of this paper is arranged as follows. In Section II, we discuss an in-depth examination of intrusion detection studies in the Industrial Internet of Things (IIoT), focusing on Deep/Machine Learning. Next section explores the innovative architecture proposed in detail. Then, it meticulously presents and compares results, culminating in insightful conclusions. Lastly, the concluding section offers valuable suggestions for future research in the ever-evolving realm of IIoT security.

2 Related works

In this section, we thoroughly review and analyze previous studies that are relevant to our research focus. These studies propose novel methods aimed at improving security in IoT settings, with a particular emphasis on using machine learning techniques to develop intrusion detection systems (IDSs) suited for IoT environments.

A pioneering study [17] introduces the IDS-SIoEL framework, employing Ensemble Learning with AdaBoost and feature selection techniques. Achieving exceptional performance on BoT-IoT, Edge-IIoT, and IoT-23 datasets, it boasts a remarkable 99.9% accuracy, recall, and precision. With swift learning and detection times, this model offers robust security solutions for diverse smart city applications.

The study published in 2022 [18] analyzes the feasibility of implementing a Host-Intrusion Detection System (HIDS) based on Deep Learning (DL-HIDS) across diverse commercial IoT devices. Findings demonstrate significant promise with up to 99.74% accuracy and minimal inference time. It emphasizes the necessity of customizing IDS for individual device classes due to their diverse architectures, providing valuable insights for securing IoT environments.

Another significant contribution comes from a study [1] tackling modern network and Industrial Internet of Things (IIoT) security challenges by designing an advanced Intrusion Detection System (IDS). Leveraging deep learning technologies, the proposed methodology optimizes network configurations, resulting in a robust IIoT anti-intrusion detection system. Demonstrating superior performance, the IDS showcases heightened detection rates, minimal false positives, and robust data correctness, aligning with privacy laws.

The paper [19] propose a deep learning ensemble model using Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) for cybersecurity in IoT environments. The ensemble approach achieves a 99.34% accuracy in detecting complex cyber-attacks, with low latency, making it suitable for real-time anomaly detection in smart environments.

In response to security challenges in the Industrial Internet of Things (IIoT), a groundbreaking solution is introduced by a paper [20], presenting PK-IDS, a cutting-edge hybrid IDS for Edge-Based IIoT. Seamlessly integrating K-NN and PCA, it achieves remarkable results: 99.10% accuracy, 98.4% detection rate, and 2.7% false alarm rate (NSL-KDD); 98.2% accuracy, 97.6% detection rate, and 2.9% false alarm rate (Bot-IoT). This addresses intricate security challenges in edge-based IIoT environments.

The authors of [21] present a hybrid SDN-based deep learning framework combining Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) to secure Internet of Medical Things (IoMT) devices. With a detection accuracy of 99.97%, the system offers scalability and efficiency, effectively addressing cyber threats in healthcare IoT environments.

Additionally, addressing security concerns in the Internet of Things (IoT), a paper [22] presents an Access Key Agreement (AKA) scheme, RDAF-IIoT, designed to boost security within IoT, particularly in industrial environments. Validated for resilience against security attacks, the scheme exhibits superiority, showcasing lower computational and communication costs compared to similar security frameworks, along with enhanced security features.

In another study [23], the focus shifts to addressing the growing cybersecurity threats to the Internet of Things (IoT), particularly in Industrial IoT (IIoT) applications. Proposing an effective Intrusion Detection System (IDS), the study employs machine learning algorithms such as K-Nearest Neighbors, Random Forest, and Logistic Regression for comprehensive evaluation on the TON_IoT dataset. The Classification and Regression Tree (CART) algorithm outperforms others, demonstrating the proposed framework's efficiency in mitigating IoT/IIoT intrusion risks.

The article [10] highlights the pivotal role of an Intrusion Detection System (IDS) in safeguarding the Industrial Internet of Things (IIoT), introducing a hybrid IDS architecture incorporating machine learning. The proposed innovations showcase heightened detection accuracy, decreased training time, and enhanced network security, well-suited for the edge scenario of IIoT.

The authors of [24] apply Hyperparameter Optimization for the XGBoost algorithm (HO-XGB) to improve network intrusion detection using the CSE-CIC-IDS 2018 dataset. The optimized model, fine-tuned with parameters like learning rate and max depth, significantly enhances intrusion detection accuracy and efficiency, outperforming traditional methods in real-time network security.

Furthermore, a study [25] addresses escalating security concerns in the IoT environment by proposing a Machine Learning (ML)-enabled Intrusion Detection System (IDS). Focusing on the modified Random Forest (RF) algorithm, the proposed IDS demonstrates effectiveness in safeguarding diverse IoT networks and applications, emphasizing its significance in addressing current security challenges.

This paper [26] conducts a comprehensive survey on the existing state and security challenges within the Internet of Things (IoT). Scrutinizing security principles across the Perception, Network, and Application layers, researchers discuss countermeasures to layer-specific security challenges and introduce future directions, emphasizing the integration of advanced networking protocols to surmount prevailing research challenges in IoT security.

Moreover, this research [27] focuses on anomaly detection in the complex and massive data generated by Industrial IoT (IIoT). Algorithms for machine learning, such as logistic regression, and decision trees are compared for their anomaly detection capabilities using three IIoT benchmark datasets[27].

Additionally, this survey delves into the synergy of AI and ML in Industry 4.0, particularly focusing on fault detection, cyber-security, and human-machine interaction [28]. It emphasizes cloud/fog/edge architectures for efficient data utilization and training and identifies open research issues in these domains[28].

The study [29] on IoT Networks presents an advanced Intrusion Detection System (IDS) using an ensemble of machine learning algorithms, including Logistic Regression (LR), Decision Trees (DT), and Random Forest (RF). Validated on real-world datasets, the IDS achieves high accuracy in detecting attacks, providing a robust solution for securing IoT networks.

Lastly, the paper [30] highlights the effectiveness of machine learning (ML) in crafting Intrusion Detection Systems (IDSs) for IoT security. With empirical analysis illustrating substantial improvements in ML-based IDS accuracy with high-quality data and models, it underscores the pivotal role of data preprocessing and model quality in achieving accurate intrusion detection within IoT environments.

The next table summarizes some related works focusing on different methodologies for enhancing Intrusion Detection Systems (IDS) in IoT and IIoT environments. Each work employs various machine learning and deep learning techniques across different datasets. The results highlight the accuracy achieved by these methods, demonstrating the effectiveness of models such as ensemble learning, hybrid frameworks, and optimized algorithms for improving security in IoT networks.

Table 1: Comparative analysis of related works on intrusion detection systems (IDS) in IoT and IIoT environments

Paper	Year	Method	Dataset	Results (Accuracy)
[17]	2023	Ensemble Learning (AdaBoost, Random Forest)	BoT-IoT, Edge-IIoT, IoT-23	99.9% Accuracy
[23]	2023	Hybrid IDS Framework (Machine Learning)	NSL-KDD, BoT-IoT	99.10% (NSL-KDD), 98.2% (BoT-IoT)
[22]	2023	Access Key Agreement (AKA) Scheme	Simulated IIoT Environment	High computational efficiency and lower communication costs
[25]	2024	Modified Random Forest Algorithm	TON-IoT, UNSW-NB15	High performance
[18]	2022	Optimized Deep Learning (HIDS)	Edge-IIoTset	Accuracy exceeding 99%
[19]	2024	Deep Learning Ensemble (DNN, LSTM)	IoT Environment	99.34% (binary classifier), 98.26% (multiclass classifier)
[24]	2024	Hyperparameter-Optimized XGBoost (HO-XGB)	CSE-CIC-IDS2018	Best performance
[21]	2024	CNN + BiLSTM Hybrid Framework	IoT-Healthcare Dataset	99.97% Accuracy
[25]	2023	ML-Enabled Intrusion Detection System	TON-IoT, UNSW-NB15	99.5% (TON-IoT), 98.7% (UNSW-NB15)

3 Methodology

The methodology section details the structure of our Intrusion Detection System (IDS), comprising three key components. The components of the Intrusion Detection System (IDS) methodology outlined here involve a systematic process:

Our model employs a systematic, three-step approach for detecting anomalies. First, we meticulously preprocess the data, removing noise, outliers, and inconsistencies to ensure data quality. We then carefully select and engineer features that hold the key to distinguishing normal behavior from anomalies. This data preparation stage sets the foundation for effective learning.

Next, we select an appropriate anomaly detection algorithm, considering factors like the specific problem and data characteristics. KNN, Decision Tree, MLP, and SVM are among the algorithms we evaluate. Once chosen, the algorithm trains on a labeled dataset, learning the patterns of normal behavior. Its performance is then rigorously validated on a separate dataset to ensure accurate anomaly detection.

Finally, we deploy the trained model into the real world to identify anomalies in real-time. We closely monitor its performance, analyzing its outputs for potential issues or false positives. This continuous feedback loop empowers us to refine the model iteratively, adjusting parameters or exploring new algorithms as needed. This three-step process, from meticulous data preparation to ongoing refinement, ensures our anomaly detection model delivers reliable results.

Figure 2 illustrates the core components of the Intrusion Detection System (IDS) proposed in this study.

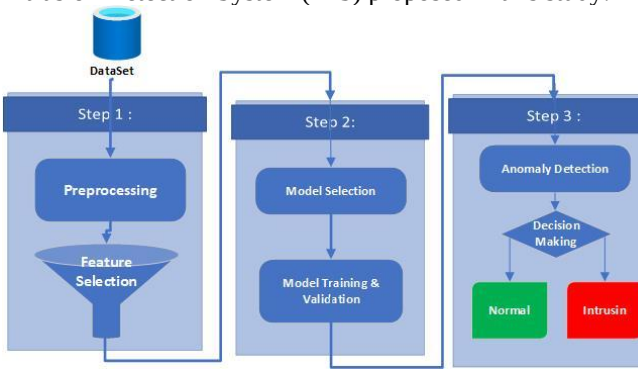


Figure 2: Components of the intrusion detection system (IDS)

The methodology section details the structure of our Intrusion Detection System (IDS), comprising three key components: data preprocessing, algorithm selection, and real-time deployment.

3.1 Data preprocessing

The data preprocessing phase is critical to preparing the datasets for effective model learning and ensuring high-quality input. This phase involved several key steps:

Noise and Outlier Removal: We initially removed noise and irrelevant data points from the datasets to improve overall data quality. Outliers were detected and eliminated using the Isolation Forest algorithm, which is well-suited for identifying anomalies in high-dimensional data.

Handling Missing Data: Missing values were handled using mean imputation for continuous variables and mode imputation for categorical variables. This ensured that no significant information was lost during preprocessing.

Feature Engineering and Selection: To reduce the dimensionality of the datasets and improve model performance, we employed Recursive Feature Elimination (RFE). RFE recursively eliminated less important features by evaluating model performance based on feature subsets, helping us identify the most relevant features. A Random Forest classifier was used to rank the feature importance, and the top 20 features were selected for further model training.

Data Normalization: We applied RobustScaler for normalization, which scaled the features based on interquartile ranges, ensuring that the impact of outliers was minimized and the data was transformed into a format suitable for machine learning algorithms.

Handling Class Imbalance: Given the imbalanced nature of the datasets, especially the BOT-IoT dataset, we applied the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE generated synthetic examples of the minority class, balancing the class distribution and improving the model's ability to detect underrepresented attack types. This technique enhanced recall rates for anomaly detection and improved the generalization of the model.

3.2 Algorithm selection and training

In the second phase, we selected suitable machine learning algorithms to detect anomalies. The algorithms considered were K-Nearest Neighbors (KNN), Decision Tree (DT), Multilayer Perceptron (MLP), and Support Vector Machine (SVM). Each algorithm was evaluated based on the characteristics of the datasets, with a particular focus on computational efficiency and detection accuracy.

Training Process: The models were trained on labeled datasets using cross-validation (Stratified k-fold cross-validation) to ensure robust performance across different data splits. This technique helped minimize overfitting and provided a more accurate assessment of each model's generalization performance.

Hyperparameter Tuning: For each algorithm, we fine-tuned the hyperparameters using grid search. In the case of MLP, parameters such as the number of hidden layers, activation function, and learning rate were optimized to achieve the best performance.

3.3 Real-time deployment and monitoring

The third phase involved deploying the trained model into real-world industrial environments to identify anomalies in real-time. The model was monitored closely to assess performance in detecting new attacks or unusual behaviors.

Anomaly Detection in Real-Time: The trained model was integrated into the IIoT system to provide real-time anomaly detection. It was designed to alert administrators to potential security threats based on deviations from learned normal behaviors.

Performance Monitoring and Feedback: A continuous feedback loop was established, where the model's outputs were analyzed to identify any issues, such as false positives or missed anomalies. This feedback loop allowed us to iteratively refine the model, adjusting hyperparameters or exploring alternative algorithms as needed.

This comprehensive, three-step methodology—covering meticulous data preprocessing, thorough algorithm selection and training, and ongoing real-time monitoring—ensures a robust and reliable Intrusion Detection System for IIoT environments.

4 Experiments

4.1 Datasets

In evaluating Intrusion Detection Systems (IDSs) with Machine Learning (ML) models, researchers frequently rely on diverse datasets to gauge performance. This study opts for two prominent IoT-based IDS datasets—Bot-IoT and Edge-IIoTset—tailored for IoT applications. These datasets, along with CIDDS, furnish extensive traffic data, enhancing the assessment of IDS models amidst evolving attack landscapes. The table below outlines the key characteristics of these datasets.

Table 2 : Key characteristics of BOT-IOT, EDGE-IIOTSET, AND CIDDS datasets

Feature	**Bot-IoT**	**Edge-IIoTset**	**CIDDS-001**
Purpose	Detect botnet activities and anomalous IoT interactions	Detect cyberattacks in industrial IoT (IIoT) environments	Evaluate anomaly-based intrusion detection systems
Content	Network traffic captures from diverse IoT devices, labeled as normal or botnet activity	Sensor data, Modbus flow information, labeled as normal or attack type	Flow records capturing network traffic features, labeled as normal or attack
Attack Types	Botnet attacks (credential stuffing, data exfiltration, etc.)	DoS/DDoS, information gathering, man-in-the-middle, injection, malware	Various intrusion attempts (e.g., port scans, denial-of-service)
Features	Varies depending on dataset version, typically includes 29 features.	61 extracted features from various sources (logs, traffic, resources, alerts)	53 flow-based features (source/destination IP, ports, packet size, etc.)
Size	73 million inputs	Not explicitly mentioned, described as "large-scale"	1.3M normal flows, 120k attack flows (compressed: 30 GB, uncompressed: 130 GB)

4.1.1 BoT-IoT Datasets:

The Bot-IoT[31] The Bot-IoT dataset, curated by UNSW Canberra Cyber experts, comprises 73 million instances of botnet attacks in IoT networks, demonstrating various attack methods such as data exfiltration, keylogging, and DDoS. Featuring both normal and simulated IoT traffic, it aids researchers and professionals in evaluating and improving IoT network security by validating intrusion detection techniques.

4.1.2 Edge_IIoTset dataset

The Edge_IIoTset Dataset[32] provides meticulously curated sensor data and communication patterns from various industrial environments, specifically tailored for Network-based Intrusion Detection Systems (NIDS) research. With a focus on anomalies and potential intrusions, it offers valuable context for evaluating and improving NIDS effectiveness in securing industrial IoT networks. Researchers and practitioners can utilize its rich contextual information to develop and validate robust security solutions for the ever-evolving landscape of industrial IoT.

4.1.3 CIDDS datasets

The CIDDS [33] (Coburg Intrusion Detection Data Sets) Dataset, created by researchers at Hochschule Coburg, Germany, is tailored for assessing anomaly-based intrusion detection systems. Simulating a small business network environment, it contains both normal and malicious activities, including various attacks like port scans and denial of service. This dataset aids in evaluating intrusion detection methods' effectiveness across diverse network scenarios.

4.2 Algorithm hyperparameters

We evaluated the following machine learning models, and the key hyperparameters for each were as follows:

Multilayer Perceptron (MLP): A 3-layer network with 64, 32, and 16 neurons in the hidden layers, respectively. The activation function used was ReLU, the optimizer was Adam, and the learning rate was set to 0.001. The MLP model was trained for 100 epochs with a batch size of 32.

K-Nearest Neighbors (KNN): The number of neighbors (k) was set to 5, and the distance metric used was Euclidean distance.

Support Vector Machine (SVM): An RBF kernel was employed, with a C parameter of 1.0 and gamma set to 'scale' to adjust automatically based on the dataset.

Decision Tree (DT): The maximum depth of the tree was set to 10, and the criterion used for splitting was Gini impurity.

4.3 Cross-validation technique

To ensure that the models generalized well and to avoid overfitting, we used Stratified k-Fold Cross-Validation with k=10. Stratified cross-validation was employed to ensure that the class distribution (i.e., the proportion of normal and attack samples) remained consistent across each fold. This technique provided a robust evaluation of model performance across different data splits and ensured that the results were reliable.

4.4 Dataset bias and challenges

4.4.1 Class imbalance in BOT-IoT dataset

One of the major challenges associated with the BOT-IoT dataset is class imbalance. The dataset contains a significant number of normal samples compared to attack instances, leading to a situation where the model could become biased toward predicting the majority class (normal traffic) while underperforming in detecting the minority class (attack traffic). In cybersecurity contexts, such biases can result in a higher number of false negatives, where actual attacks go undetected, which can have serious consequences.

4.4.2 SMOTE for addressing class imbalance

To address the issue of class imbalance in the BOT-IoT dataset, we employed the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE works by generating synthetic samples for the minority class (i.e., attack instances), which helps balance the distribution of the dataset. Specifically, SMOTE synthesizes new attack instances by interpolating between existing samples, ensuring that the model is exposed to more representative attack data during training.

This technique helped mitigate the bias toward the majority class (normal traffic) and improved the model's ability to detect attack traffic, as reflected in the improved recall and F1-scores during evaluation.

4.4.3 Other dataset challenges

Edge-IIoTset and CIDDS also presented challenges in terms of diverse feature types, such as network traffic features, sensor data, and Modbus flow information. These datasets required careful preprocessing and feature selection (detailed in the Methodology section) to ensure that the most relevant features were used for training.

While class imbalance was less of an issue in Edge-IIoTset and CIDDS, challenges like varying feature distributions and data sparsity still required careful handling during model training and evaluation.

4.4.4 Performance improvements

The application of SMOTE in balancing the BOT-IoT dataset had a noticeable impact on performance metrics, particularly for models like MLP and SVM. By balancing the class distribution, the recall for attack instances increased, improving the model's overall ability to detect cyberattacks. This improvement is especially critical for intrusion detection systems (IDS), where the cost of false negatives (missed attacks) is high.

4.5 Performance measures

In assessing models, research literature utilizes diverse performance metrics, including precision, recall, accuracy, F1-measure, and Matthew's correlation coefficient, derived from confusion matrices.

Accuracy: The correctness of the model's predictions is often evaluated through accuracy measures[34].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision (Positive Predictive Value): Measures the accuracy of positive predictions by evaluating how many of the predicted positive instances are truly positive[35].

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \quad (2)$$

Recall (Sensitivity, True Positive Rate): Indicates the percentage of true positives that the model successfully detected by gauging the model's capacity to collect all pertinent cases[35].

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (3)$$

F1_Score: Provides a balanced metric between recall and accuracy by representing the harmonic mean of the two[34].

$$\text{F1}_{\text{Score}} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}} \quad (4)$$

MCC evaluates binary model performance by considering true/false positives/negatives, offering a balanced assessment of classification accuracy [36].

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}} \quad (5)$$

5 Results

In this study, we analyze the results and findings derived from the two primary benchmark datasets previously mentioned. We assess the performance of the following machine learning algorithms: SVM, MLP, KNN, DT, and XGBoost. Various performance metrics including precision, recall, accuracy, MCC, F1-measure, and training time are utilized in the research literature to gauge the effectiveness of these models.

To evaluate the performance of the proposed model, we use the following metrics:

Accuracy (ACC): The proportion of true results (both true positives and true negatives) among the total number of cases. Accuracy measures the overall correctness of the model's predictions.

Precision: The proportion of true positive predictions among all positive predictions made by the model, representing the model's ability to avoid false positives.

Recall (Sensitivity): The proportion of true positive cases that are correctly identified by the model, emphasizing the model's capacity to capture actual positive instances.

F1-Score: The harmonic mean of precision and recall, providing a balanced metric when precision and recall are of equal importance.

Matthews Correlation Coefficient (MCC): A more comprehensive metric that takes into account true and false positives and negatives, providing a balanced measure even in cases of class imbalance.

5.1.1 Revisiting results: insights from the CIDDS dataset

The following table (3) presents the performance metrics of various machine learning algorithms on the CIDDS dataset. Metrics include accuracy (ACC%), precision, F1-score, recall, Matthews correlation coefficient (MCC), and training time in seconds. SVM achieves an accuracy of 93.77%, followed by K-NN with 96.66%, MLP with

99.26%, and DT with 98.00%. Precision, recall, and F1-score indicate high model performance across all algorithms, with values close to 1. Matthew's correlation coefficient values range from 0.9122 to 0.9931, suggesting strong correlations between predicted and actual values. Training time varies among the algorithms, with MLP being the fastest at 10.23 seconds.

The Fig. 3 is a graphical metrics representation on the CIDDS dataset showcases the performance of machine learning algorithms. Accuracy varies from 93.77% for SVM to 99.26% for MLP, with MLP leading in precision at 98.74%. F1-score demonstrates consistent high performance, particularly with MLP achieving 98.80%. Recall rates are generally high, ranging from 93.1% for SVM to 98.86% for MLP. Matthew's correlation coefficient (MCC) values, ranging from 91.22% to 99.31%, indicate strong correlations between predicted and actual values.

Table 3: Performance metrics of machine learning algorithms on CIDDS dataset

ML algorithm	ACC %	Precision%	F1-score %	Recall %	MCC %	Training Time (s)
SVM	,9377	0,901	0,941	0,931	0,9622	75,94
K-NN	,9666	0,9263	0,9342	0,943	0,9626	12,56
DT	,98	0,979	0,979	0,979	0,9122	27,5
MLP	,9926	0,9874	0,988	0,9886	0,9931	10,23

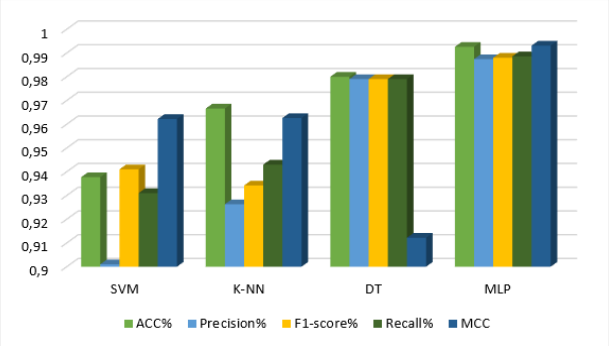


Figure 3: Performance metrics comparison of machine learning algorithms on CIDDS dataset

5.1.2 Revisiting Results: Insights from the BOT-IoT Dataset

The table (1) displays the performance metrics results of various machine learning algorithms on the BOT-IoT dataset. Metrics include accuracy (ACC %), precision, F1-score, recall, Matthew's correlation coefficient (MCC %), and training time in seconds. SVM achieves an accuracy of 98.15%, followed by K-NN with 99.13%, MLP with 99.52%, and DT with 93.4%. Precision, recall, and F1-score indicate high model performance across all algorithms, with values close to 1. Matthew's correlation coefficient values range from 0.9122 to 0.97, suggesting strong correlations between predicted and actual values. Training time varies among the algorithms, with SVM being the fastest at 34.78 seconds.

Table 2: The BOT-IoT dataset's performance metrics results

ML algorithm	ACC %	Precision %	F1-score%	Recall %	MCC %	Training Time (s)
SVM	0,9815	0,9915	0,9915	0,995	0,92	34,78
K-NN	0,9913	0,9933	0,9933	0,999	0,92	56,43
DT	0,934	0,953	0,934	0,963	0,9122	27,3
MLP	0,9952	0,9952	0,9952	0,9999	0,97	48,34

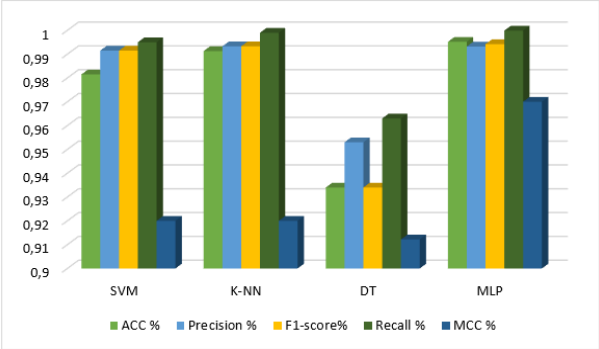


Figure 4: Performance metrics comparison of machine learning algorithms on BOT-IoT dataset

The graphical representation of performance metrics on the BOT-IoT dataset depicts key evaluation measures for machine learning algorithms. Accuracy ranges from 93.4% for DT to 99.52% for MLP, with MLP achieving the highest precision at 99.52%. F1-score shows consistent high performance across all algorithms, particularly with MLP achieving 99.52%. Recall rates are generally high, ranging from 96.3% for DT to 99.9% for MLP. Matthews correlation coefficient (MCC) values, ranging from 91.22% to 97%, highlight strong correlations between predicted and actual values.

5.1.3 Revisiting results: insights from the Edge_IIoTset dataset

This next table presents the performance metrics results of various machine learning algorithms on the Edge_IIoTset dataset. Metrics include accuracy (ACC %), precision, F1-score, recall, Matthew's correlation coefficient (MCC %), and training time in seconds. SVM achieves an accuracy of 99.4%, followed by K-NN with 99.1%, MLP with 99.93%, and DT with 94.9%. Precision, recall, and F1-score indicate high model performance across all algorithms, with values close to 1. Matthew's correlation coefficient values range from 0.931 to 0.992, suggesting strong correlations between predicted and actual values. Training time varies among the algorithms, with MLP being the fastest at 11 seconds.

Table 3: The Edge_IIoTset dataset's performance metrics results

ML algorithm	ACC %	Precision %	F1-score %	Recall %	MCC %	Training Time (s)
SVM	0,994	0,984	0,999	0,9996	0,96	58,42
K-NN	0,991	0,991	0,997	0,9999	0,961	13,61
DT	0,949	0,941	0,989	0,966	0,931	137,2
MLP	0,9993	0,9993	0,999	0,9999	0,992	11

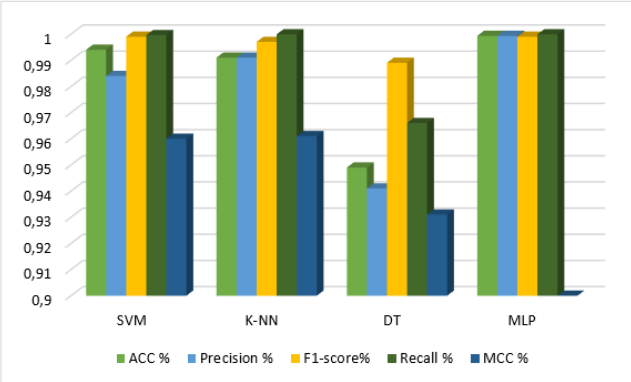


Figure 5: Performance metrics comparison of machine learning algorithms on Edge_IIoTset dataset

The graphical representation illustrates the performance metrics of machine learning algorithms on the Edge_IIoTset dataset. Accuracy ranges from 94.9% for DT to 99.93% for MLP, with MLP achieving the highest precision at 99.93%. F1-score demonstrates consistent high performance across all algorithms, particularly with MLP achieving 99.9%. Recall rates are generally high, ranging from 96.6% for DT to 99.99% for MLP. Matthew's correlation coefficient (MCC) values, ranging from 93.1% to 99.2%, indicate strong correlations between predicted and actual values.

Figure 6 illustrates the performance metrics of several machine learning algorithms, evaluated across three datasets: CIDDS, BOT-IoT, and Edge_IIoTset. The subfigures (a) to (d) compare metrics such as accuracy, precision, recall, and the Matthews Correlation Coefficient (MCC) among algorithms like SVM and MLP. Notably, MLP consistently demonstrates superior precision, recall, and MCC across all datasets, showcasing its effectiveness in intrusion detection tasks. SVM, on the other hand, achieves commendable accuracy, particularly excelling in classifying instances correctly. This figure highlights the comparative strengths of each algorithm in detecting and managing cyber threats in IIoT environments.

Fig. 6. (a) Accuracy (ACC %): Across all datasets, The SVM model demonstrates strong accuracy across datasets, achieving 93.77% on CIDDS, 98.15% on BOT-IoT, and 99.4% on Edge_IIoTset. It demonstrates the ability of SVM to effectively classify instances into correct categories.

Fig. 6. (b) Precision (%) measured for each dataset: MLP exhibits superior precision across datasets, with values of 98.74% for CIDDS, 99.52% for BOT-IoT, and 99.93% for Edge_IIoTset. This indicates MLP's capability to provide precise positive predictions.

Fig. 6. (c) Recall (%) across datasets: MLP demonstrates outstanding recall rates across all datasets, with values of 98.86% for CIDDS, 99.99% for BOT-IoT, and 99.96% for Edge_IIoTset. It highlights MLP's effectiveness in capturing true positive instances.

Fig. 6. (d) MCC (%) demonstrating balanced classification performance: MLP achieves the highest MCC values for all datasets, with values of 99.31% for CIDDS, 97% for BOT-IoT, and 99.2% for Edge_IIoTset. MCC considers true and false positives and negatives, providing a balanced measure of classification performance.

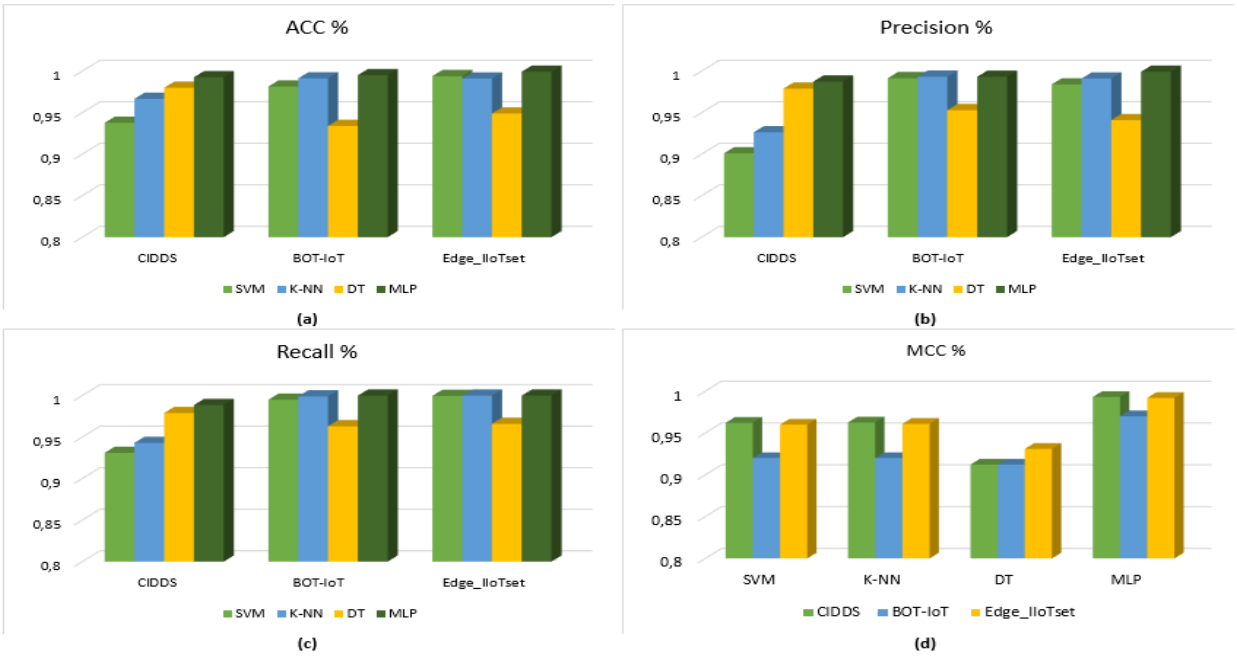


Figure 6: The performance metrics of machine learning algorithms on three distinct datasets: CIDDS, BOT-oT, and Edge_IIoTset

Comparing the performance metrics across the CIDDS, BOT-IoT, and Edge_IIoTset datasets, it becomes evident that MLP consistently outperforms other machine learning algorithms. MLP exhibits superior accuracy, precision, and recall rates across all datasets, indicating its effectiveness in correctly classifying instances and minimizing false positives. Additionally, MLP achieves high F1-scores and Matthew's correlation coefficient values, highlighting its ability to strike a balance between precision and recall while maintaining strong correlations between predicted and actual values. Moreover, MLP demonstrates computational efficiency by requiring relatively shorter training times compared to other algorithms. Thus, MLP emerges as the best model choice for handling diverse and complex datasets, making it suitable for tasks requiring high precision and recall rates in industrial IoT environments.

6 Discussion

The results show that the MLP model consistently outperforms other models, including SVM, KNN, and DT, in terms of accuracy, precision, and recall across all datasets. Specifically, MLP achieved a 99.26% accuracy on the CIDDS dataset, while KNN and DT only reached 96.66% and 98%, respectively.

In comparison to ensemble methods like Random Forest or AdaBoost, MLP continues to show superiority in terms of precision and recall. However, ensemble methods may offer benefits in terms of reducing variance and improving robustness, which could be valuable in certain IIoT scenarios where variability in data is high and frequent model retraining may be required.

The superior performance of MLP can be attributed to its ability to capture complex non-linear patterns in the data through its multilayer architecture. This ability is particularly crucial in detecting multi-stage and sophisticated cyberattacks in IIoT environments, where other models like SVM and KNN might struggle with high-dimensional, noisy, or imbalanced data. The use of deep learning helps in identifying subtle anomalies that could otherwise be missed by traditional models.

MLP also excels in terms of computational efficiency due to its parallel processing capabilities, making it well-suited for real-time anomaly detection. However, it is important to note that the training time for MLP was higher compared to KNN and SVM, which might present challenges in resource-constrained IIoT environments where real-time responsiveness is critical. A hybrid approach that combines MLP for offline analysis with lighter models for real-time intrusion detection could offer a balanced solution to address this challenge.

Despite its advantages, further evaluation of the MLP model in real-world IIoT environments is needed to assess its scalability and performance in dynamic industrial settings. Potential issues like false negatives, which are particularly costly in cybersecurity, should be addressed by optimizing decision thresholds or exploring ensemble learning approaches to reduce the chances of undetected intrusions. Additionally, real-time deployment may face challenges due to latency and computational overhead, which future work could mitigate by optimizing MLP's architecture or incorporating federated learning approaches to distribute the computational load.

While the presented performance metrics, such as accuracy and precision, demonstrate strong overall performance, it is crucial to consider the impact of false negatives in IIoT environments. False negatives occur when an attack is not detected, which can be particularly costly in industrial settings, leading to potential disruptions, operational failures, or even severe security breaches.

To mitigate the risk of false negatives, we propose several strategies:

Adjusting Decision Thresholds: Fine-tuning the decision thresholds of the classifier can lower the rate of false negatives by prioritizing sensitivity (recall) over precision in critical security scenarios.

Ensemble Methods: Leveraging ensemble techniques like Random Forest or boosting algorithms (e.g., AdaBoost, XGBoost) can help improve the detection capabilities by combining the strengths of multiple weak classifiers, leading to a more sensitive system.

Cost-Sensitive Learning: Implementing cost-sensitive learning models can assign a higher penalty to false negatives, ensuring that the model focuses on minimizing missed detections, particularly in cybersecurity contexts where the cost of undetected attacks is high.

6.1 Real-world applicability and deployment

While the MLP model demonstrates strong performance metrics in experimental settings, deploying it in real-world IIoT environments poses several practical challenges:

6.1.1 Scalability

IIoT environments typically involve vast networks of connected devices that generate large amounts of real-time data. Scaling an MLP-based intrusion detection system to handle this volume of data is non-trivial. In practice, distributed computing or edge computing architectures can be used to process data closer to its source, reducing the load on centralized servers. This approach allows for more scalable and efficient detection of anomalies across large-scale IIoT networks.

6.1.2 Latency

In real-time industrial environments, low latency is critical to detecting and responding to cyberattacks before they impact operations. Although MLP provides high accuracy, its inference time can be slower compared to simpler models like Decision Trees or KNN. To ensure real-time detection, optimization strategies such as reducing the number of layers or neurons in the MLP, or employing lightweight models for initial anomaly detection, can help lower the latency without sacrificing detection accuracy.

6.1.3 Computational overhead

MLP models are computationally intensive, especially when applied to large datasets or in environments with limited resources, such as edge or fog computing. To address this, techniques like model pruning, quantization, or offloading computations to specialized hardware (e.g., GPUs or FPGAs) can be used to reduce computational overhead while maintaining model performance. Additionally, deploying the MLP model in a distributed manner across multiple edge nodes can balance the computational demands.

6.2 Hypothetical case study: MLP deployment in a smart factory

To illustrate the practical integration of the MLP model, consider a smart manufacturing facility where various machines, sensors, and devices are connected through an IIoT network to monitor production processes in real-time.

6.2.1 Edge-based deployment

The MLP model could be deployed at the edge of the network, near the data sources (e.g., sensors and industrial machines), to detect anomalies in network traffic in real time. Each edge node would process a subset of the network traffic, allowing for faster detection and reduced network congestion. The system would flag suspicious behavior, such as unauthorized access or unusual data flows, and alert administrators before any significant damage occurs.

6.2.2 Scalability considerations

In large industrial environments with thousands of connected devices, a distributed MLP deployment would be necessary. Each edge node would independently run an instance of the MLP model, analyzing traffic locally and sending detection results to a central server for aggregation and final decision-making. This distributed approach ensures that the IDS can scale efficiently across the entire facility while maintaining high detection accuracy.

6.2.3 Real-time monitoring

The MLP model would continuously monitor network traffic, allowing it to identify deviations from expected patterns of behavior. In the event of a detected anomaly, the system could automatically trigger predefined responses, such as isolating compromised devices or blocking network traffic from suspicious sources. This proactive approach minimizes downtime and ensures uninterrupted production.

6.3 Conclusion on real-world integration

To deploy the MLP model in real-world IIoT environments, challenges related to scalability, latency, and computational overhead must be addressed. Solutions like edge computing, distributed architectures, and hardware acceleration can enable the effective deployment of MLP-based intrusion detection systems in industrial settings. Future work should focus on optimizing MLP models for real-time environments, ensuring they can scale and respond to evolving cyber threats in large, dynamic IIoT networks.

7 Conclusion

In conclusion, this study emphasizes the critical need to secure Industrial Internet of Things (IIoT) environments in the context of Industry 4.0. While IIoT facilitates unprecedented connectivity and efficiency in manufacturing, it also introduces significant cybersecurity

challenges. Leveraging deep learning techniques, particularly the Multilayer Perceptron (MLP), presents a promising avenue for fortifying IIoT systems against cyber threats. Through extensive analyses on benchmark datasets, MLP consistently demonstrates superior performance metrics, including accuracy, precision, recall, and efficiency. Its ability to effectively classify instances and maintain strong correlations between predicted and actual values makes it an ideal choice for anomaly detection in industrial IoT environments. Moving forward, future research can explore the integration of MLP-based models into IIoT infrastructures to enhance security and optimize manufacturing processes further. Additionally, ongoing efforts in developing advanced deep learning algorithms tailored for specific IIoT applications can contribute to bolstering cybersecurity and resilience in the industry 4.0 landscape.

References

[1] A. Arun Kumar and R. Krishna Karne, "IIoT-IDS Network using Inception CNN Model," *JTCSST*, vol. 4, no. 3, pp. 126–138, Aug. 2022, doi: 10.36548/jtcsst.2022.3.002.

[2] J. Du, K. Yang, Y. Hu, and L. Jiang, "NIDS-CNNLSTM: Network Intrusion Detection Classification Model Based on Deep Learning," *IEEE Access*, vol. 11, pp. 24808–24821, 2023, doi: 10.1109/ACCESS.2023.3254915.

[3] G. Czeczot, I. Rojek, D. Mikołajewski, and B. Sangho, "AI in IIoT Management of Cybersecurity for Industry 4.0 and Industry 5.0 Purposes," *Electronics*, vol. 12, no. 18, p. 3800, Sep. 2023, doi: 10.3390/electronics12183800.

[4] M. T. Shakir and B. Shannaq, "Enhancing Security through Multi-Factor User Behavior Identification: Moving Beyond the Use of the Longest Common Subsequence (LCS)," *IJCAI*, vol. 48, no. 19, Nov. 2024, doi: 10.31449/inf.v48i19.6270.

[5] L. Idouglid, S. Tkatek, K. Elfayq, and A. Guezzaz, "A NOVEL ANOMALY DETECTION MODEL FOR THE INDUSTRIAL INTERNET OF THINGS USING MACHINE LEARNING TECHNIQUES," no. 1, 2024, doi: doi: 10.32620/reks.2024.1.12.

[6] Z. Zhang, "SD-WSN Network Security Detection Methods for Online Network Education," *IJCAI*, vol. 48, no. 21, Nov. 2024, doi: 10.31449/inf.v48i21.6257.

[7] M. Al-Ambusaidi, Z. Yinjun, Y. Muhammad, and A. Yahya, "ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications," *Soft Comput*, Dec. 2023, doi: 10.1007/s00500-023-09452-7.

[8] I. Farhadian Dehkordi, K. Manochehri, and V. Aghazarian, "Internet of Things (IoT) Intrusion Detection by Machine Learning (ML): A Review," *APJITM*, vol. 12, no. 01, pp. 13–38, Jun. 2023, doi: 10.17576/apjitm-2023-1201-02.

[9] L. Idouglid, S. Tkatek, K. Elfayq, and A. Guezzaz, "Next-gen security in IIoT: integrating intrusion detection systems with machine learning for industry 4.0 resilience," *IJECE*, vol. 14, no. 3, p. 3512, Jun. 2024, doi: 10.11591/ijece.v14i3.pp3512-3521.

[10] H. Yao, P. Gao, P. Zhang, J. Wang, C. Jiang, and L. Lu, "Hybrid Intrusion Detection System for Edge-Based IIoT Relying on Machine-Learning-Aided Detection," *IEEE Network*, vol. 33, no. 5, pp. 75–81, Sep. 2019, doi: 10.1109/MNET.001.1800479.

[11] S. Haque, F. El-Moussa, N. Komninos, and R. Muttukrishnan, "A Systematic Review of Data-Driven Attack Detection Trends in IoT," *Sensors*, vol. 23, no. 16, p. 7191, Aug. 2023, doi: 10.3390/s23167191.

[12] Q. Huang, H. Xian, L. Mei, X. Cheng, N. Li, and N. Li, "Intelligent Distribution Network Operation and Anomaly Detection Based on Information Technology," *IJCAI*, vol. 49, no. 9, Feb. 2025, doi: 10.31449/inf.v49i9.5584.

[13] M. Alshar'e, K. Abuhmaidan, F. Y. H. Ahmed, A. Abualkishik, M. Al-Bahri, and J. H. Yousif, "Assessing Blockchain's Role in Healthcare Security: A Comprehensive Review," *IJCAI*, vol. 48, no. 22, Dec. 2024, doi: 10.31449/inf.v48i22.6155.

[14] R. Alghamdi and M. Bellaiche, "An ensemble deep learning based IDS for IoT using Lambda architecture," *Cybersecurity*, vol. 6, no. 1, p. 5, Mar. 2023, doi: 10.1186/s42400-022-00133-w.

[15] S. Latif *et al.*, "Deep Learning for the Industrial Internet of Things (IIoT): A Comprehensive Survey of Techniques, Implementation Frameworks, Potential Applications, and Future Directions," *Sensors*, vol. 21, no. 22, p. 7518, Nov. 2021, doi: 10.3390/s21227518.

[16] H. He, "Automatic Network Traffic Scheduling Algorithm Based on Deep Reinforcement Learning," *IJCAI*, vol. 48, no. 22, Dec. 2024, doi: 10.31449/inf.v48i22.6943.

[17] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, "lIDS-SIoEL: intrusion detection framework for IoT-based smart environments security using ensemble learning," *Cluster Comput*, vol. 26, no. 6, pp. 4069–4083, Dec. 2023, doi: 10.1007/s10586-022-03810-0.

[18] I. Idrissi, M. Azizi, and O. Moussaoui, "A Lightweight Optimized Deep Learning-based Host-Intrusion Detection System Deployed on the Edge for IoT," *IJCDS*, vol. 11, no. 1, pp. 209–216, Jan. 2022, doi: 10.12785/ijcds/110117.

[19] D. Venkataraya Premalatha and S. Ramanujam, "Securing the internet of things frontier: a deep learning ensemble for cyber-attack detection in smart environments," *IJ-AI*, vol. 13, no. 4, p. 4736, Dec. 2024, doi: 10.11591/ijai.v13.i4.pp4736-4746.

[20] A. Guezzaz, M. Azrour, S. Benkirane, M. Mohy-Eddine, H. Attou, and M. Douiba, "A Lightweight Hybrid Intrusion Detection Framework using Machine Learning for Edge-Based IIoT Security," *IAJIT*, vol. 19, no. 5, 2022, doi: 10.34028/iajit/19/5/14.

[21] Y. Rbah *et al.*, "Hybrid software defined network-based deep learning framework for enhancing internet of medical things cybersecurity," *IJ-AI*, vol. 13, no. 3, p. 3599, Sep. 2024, doi: 10.11591/ijai.v13.i3.pp3599-3610.

[22] H. Alasmary, "RDAF-IIoT: Reliable Device-Access Framework for the Industrial Internet of Things," *Mathematics*, vol. 11, no. 12, p. 2710, Jun. 2023, doi: 10.3390/math11122710.

[23] S. Alshathri, A. El-Sayed, W. El-Shafai, and E. El-Din Hemdan, "An Efficient Intrusion Detection Framework for Industrial Internet of Things Security," *Computer Systems Science and Engineering*, vol. 46, no. 1, pp. 819–834, 2023, doi: 10.32604/csse.2023.034095.

[24] W. Chimphlee and S. Chimphlee, "Hyperparameters optimization XGBoost for network intrusion detection using CSE-CIC-IDS 2018 dataset," *IJ-AI*, vol. 13, no. 1, p. 817, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp817-826.

[25] M. Al-Ambusaidi, Z. Yinjun, Y. Muhammad, and A. Yahya, "ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications," *Soft Comput*, Dec. 2023, doi: 10.1007/s00500-023-09452-7.

[26] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, London, United Kingdom: IEEE, Dec. 2015, pp. 336–341. doi: 10.1109/ICITST.2015.7412116.

[27] Bhupal Naik D. S., V. Dondeti, and S. Balakrishna, "Comparative Analysis of Machine Learning-Based Algorithms for Detection of Anomalies in IIoT:," *International Journal of Information Retrieval Research*, vol. 12, no. 1, pp. 1–55, May 2022, doi: 10.4018/IJIRR.298647.

[28] A. Angelopoulos *et al.*, "Tackling Faults in the Industry 4.0 Era—A Survey of Machine-Learning Solutions and Key Aspects," *Sensors*, vol. 20, no. 1, p. 109, Dec. 2019, doi: 10.3390/s20010109.

[29] I. T. AL-Halboosi, B. Mohamed Elbagoury, S. Amin El-Regaily, and E.-S. M. El-Horbaty, "Federated inception-multi-head attention models for cyber-attacks detection," *IJ-AI*, vol. 13, no. 4, p. 4778, Dec. 2024, doi: 10.11591/ijai.v13.i4.pp4778-4794.

[30] S. Alkadi, S. Al-Ahmadi, and M. M. Ben Ismail, "Toward Improved Machine Learning-Based Intrusion Detection for Internet of Things Traffic," *Computers*, vol. 12, no. 8, p. 148, Jul. 2023, doi: 10.3390/computers12080148.

[31] X. Zhang, O. Upton, N. L. Beebe, and K.-K. R. Choo, "IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers," *Forensic Science International: Digital Investigation*, vol. 32, p. 300926, Apr. 2020, doi: 10.1016/j.fsidi.2020.300926.

[32] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[33] J. Carneiro, N. Oliveira, N. Sousa, E. Maia, and I. Praça, "Machine Learning for Network-based Intrusion Detection Systems: an Analysis of the CIDDS-001 Dataset," Jul. 02, 2021, *arXiv*: arXiv:2107.02753. Accessed: Sep. 30, 2023. doi: 10.1007/978-3-030-86261-9_15

[34] D. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," p. pp.37-63, 2011, doi: 10.48550/arXiv.2010.16061.

[35] T. Wu *et al.*, "DiscrimLoss: A Universal Loss for Hard Samples and Incorrect Samples Discrimination," Aug. 21, 2022, *arXiv*: arXiv:2208.09884. doi: 10.1109/TMM.2023.3290477.

[36] U. Itai and N. Katz, "Goodness of Fit Metrics for Multi-class Predictor," Aug. 11, 2022, *arXiv*: arXiv:2208.05651. doi: 10.48550/arXiv.2208.05651.

Enhancing Stock Market Predictions Using Hybrid Deep Learning Models with Sentiment Analysis and Feature Engineering

Sathya N[1*], R Indradevi[2], R. Gayathiri[3]
[1]Department of Commerce, School of Social Sciences and Languages,
Vellore Institute of Technology, Vellore, India.
[2]Department of Commerce, School of Social Sciences and Languages, Vellore Institute of Technology, Vellore, India
[3]Department of commerce (Banking &Insurance), Auxilium College (Autonomous), Vellore, India

Keywords: stock market prediction, deep learning, hybrid models, sentiment analysis, feature engineering, LSTM, CNN

Recieved: January 16, 2025

Stock price prediction remains a complex yet crucial task in financial markets, requiring robust methodologies to capture intricate dependencies in stock price movements. This study proposes the Hybrid Deep Learning for Stock Market Prediction (HDL-SMP) model, integrating Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM), and an Attention Mechanism to enhance predictive accuracy. The model processes three key data sources: historical stock prices, technical indicators, and sentiment analysis scores from financial news and social media. The CNN layer extracts spatial patterns in price movements, while the BiLSTM layer learns long-term dependencies and captures both past and future trends. The Attention layer assigns dynamic weights to features based on their significance, optimizing prediction outcomes. Experimental evaluations on benchmark stock market datasets demonstrate the superiority of HDL-SMP over traditional machine learning models and existing deep learning architectures. The model achieves higher prediction accuracy, lower error rates, and improved robustness against market volatility. A comparative analysis with Support Vector Regression (SVR), CNN-BiLSTM, and other hybrid deep learning models confirms the effectiveness of HDL-SMP in financial forecasting. These findings highlight the potential of deep learning-based models in stock market prediction, aiding investors and financial analysts in making informed decisions.

Povzetek: Članek predstavi hibridni HDL-SMP model (CNN, BiLSTM, Attention) s tehničnimi indikatorji in sentimentno analizo, ki presega obstoječe pristope pri napovedovanju delnic z nižjimi napakami in višjim R^2.

1 Introduction

Stock market prediction has been a subject of extensive research due to its significance in financial decision-making, risk assessment, and investment strategies. Accurate stock market forecasting can provide investors with critical insights to optimize their portfolios and mitigate risks. However, predicting stock prices remains a challenging task due to the highly volatile, dynamic, and non-linear nature of financial markets, which are influenced by a complex interplay of historical price patterns, macroeconomic factors, market sentiment, and global events [1-4]. Traditional statistical methods such as Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) have been widely used for stock price forecasting, but they often struggle to capture the intricate dependencies and latent patterns present in large-scale financial data. With the advent of machine learning and deep learning, researchers have explored the potential of artificial intelligence-driven models in enhancing prediction accuracy. Among these, deep learning techniques such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) have gained prominence due to their ability to model temporal dependencies and extract meaningful features from complex datasets. Despite their effectiveness, standalone deep learning models often face limitations in handling noisy data and external influencing factors such as investor sentiment and news-driven market movements [5-7]. To address these challenges, this study proposes a hybrid deep learning framework that integrates sentiment analysis and feature engineering to enhance stock market predictions.

Sentiment analysis plays a crucial role in financial forecasting by incorporating market sentiment derived from news articles, social media, and financial reports. Investor sentiment, which reflects public perception and emotions toward specific stocks or market trends, has been shown to impact stock price movements significantly.

Sentiment analysis techniques leverage natural language processing (NLP) and machine learning algorithms to quantify sentiment scores from textual data, allowing models to incorporate qualitative information alongside traditional numerical indicators. The integration of sentiment analysis into stock prediction models has gained traction in recent years, as studies suggest that market sentiment can serve as a leading indicator for stock price fluctuations [8-11]. However, effectively capturing and integrating sentiment information remains a challenge due to the subjective nature of textual data, varying linguistic structures, and the presence of misinformation. In this study, a sentiment analysis module is incorporated into the hybrid model to extract meaningful sentiment-driven features from financial news and social media discussions [12-17]. The sentiment scores are then utilized as additional input features to enhance predictive performance. Feature engineering is another critical aspect of stock market prediction that involves transforming raw data into informative and structured representations that improve model performance. Traditional feature selection methods rely on predefined technical indicators such as moving averages, Bollinger Bands, Relative Strength Index (RSI), and MACD (Moving Average Convergence Divergence). While these indicators provide valuable insights, deep learning models can further benefit from automated feature extraction and selection techniques. The proposed hybrid approach leverages deep learning-driven feature engineering to extract high-dimensional feature representations from historical stock prices, trading volumes, and external sentiment data. By combining domain-specific feature selection with deep learning-based automated feature extraction, the model enhances its ability to capture intricate dependencies and reduce redundant or irrelevant information. The hybrid architecture employed in this study combines LSTM and CNN models to exploit both sequential and spatial characteristics in stock market data. LSTM networks are particularly effective in handling time-series data by preserving long-term dependencies, while CNNs excel in capturing local patterns and feature correlations. The fusion of these architectures enables the model to learn hierarchical representations and improve forecasting accuracy.

2 Literature review

In recent years, hybrid deep learning models have gained popularity in stock market prediction due to their ability to learn intricate patterns from large datasets. These models often integrate different neural network architectures, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformer-based models, to improve predictive accuracy. Furthermore, sentiment analysis has emerged as a crucial factor in financial forecasting, as investor sentiment derived from social media, news articles, and financial reports significantly impacts stock prices.

Feature engineering plays a pivotal role in enhancing model performance by selecting the most relevant input variables. By incorporating a combination of technical indicators, fundamental analysis, and sentiment scores, hybrid models can offer improved forecasting capabilities. Despite these advancements, challenges remain, including data quality issues, overfitting, and the ability to generalize across different market conditions.

This section explores recent studies on stock market prediction, focusing on the advantages and limitations of various approaches, particularly hybrid deep learning models integrated with sentiment analysis and feature engineering.

Olawale et al. [9] explored the impact of combining sentiment analysis and historical price data in stock market predictions. Their study emphasized how integrating investor sentiment from financial news and social media enhances traditional forecasting models. Advantages of their approach include improved model accuracy due to the incorporation of real-time market sentiment, which captures investor behavior more effectively than historical prices alone. However, a notable disadvantage is the challenge of ensuring data reliability, as sentiment data from social media may be noisy, misleading, or subject to manipulation.

Ebrahimian et al. [10] applied Support Vector Regression (SVR) to forecast stock prices, highlighting its ability to handle nonlinear patterns effectively. Their findings suggest that SVR models perform well when combined with sentiment-related variables and technical indicators. Advantages of this approach include the robustness of SVR in capturing complex relationships between stock prices and influencing factors. However, a key disadvantage is that SVR models can be computationally expensive and require careful parameter tuning to avoid overfitting.

Das et al. [11] developed hybrid deep learning models that integrate an enhanced Twitter sentiment score with technical indicators for stock price prediction. Their approach demonstrated that incorporating deep learning with sentiment analysis significantly reduces prediction errors. Advantages of their model include improved accuracy and the ability to adapt to sudden market changes by analyzing real-time sentiment data. However, a disadvantage is the dependency on the quality and availability of sentiment data, which may not always reflect actual market movements.

Zhong and Enke [12] investigated the use of hybrid machine learning algorithms for predicting the daily return direction of stock markets. Their study found that combining multiple algorithms led to improved predictive performance. Advantages include enhanced robustness and generalization across different market conditions. However, a disadvantage is the increased complexity of model implementation and the need for substantial computational resources.

Zheng et al. [13] introduced an evolutionary framework incorporating bidirectional Long Short-Term Memory (BiLSTM) networks for stock price prediction. Their study demonstrated that BiLSTM networks effectively capture sequential dependencies in financial time series data. Advantages of this approach include improved long-term prediction accuracy and the ability to retain important historical trends. However, a disadvantage is the potential for overfitting, especially when training on limited data.

Wang et al. [14] proposed a hybrid model combining Convolutional Neural Networks (CNNs) with BiLSTM architectures to predict stock closing prices. Their results showed that CNNs help extract spatial features from stock data, while BiLSTMs enhance sequential learning. Advantages of this hybrid model include a more comprehensive understanding of market patterns and improved generalization. However, a disadvantage is the requirement for a large dataset to train both CNN and BiLSTM components effectively.

Kumar et al. [15] applied bootstrap aggregation ensemble learning for financial forecasting, illustrating the effectiveness of ensemble methods in improving model reliability. Advantages of their approach include increased stability and reduced variance, leading to more consistent predictions. However, a disadvantage is the computational burden associated with training multiple models and aggregating their results.

Smith and O'Hare [16] examined the relationship between traditional news, social media sentiment, and stock price movements. Their findings highlighted that social media sentiment often precedes price changes, making it a valuable predictive factor. Advantages of their study include the identification of sentiment as a leading indicator for stock price movements. However, a disadvantage is the challenge of distinguishing between genuine sentiment shifts and market noise.

Raman et al. [17] conducted a mixed-methods study leveraging news article sentiment to predict firm performance. Their research demonstrated that combining qualitative and quantitative sentiment analysis improves predictive power. Advantages include a more nuanced understanding of market trends by considering both numerical data and textual insights. However, a disadvantage is the subjectivity involved in sentiment classification, which may lead to inconsistencies.

Passalis et al. [18] analyzed multiple sources of financial sentiment to detect Bitcoin price changes using deep learning techniques. Their study found that multi-source sentiment analysis enhances prediction accuracy. Advantages of their approach include a broader perspective on investor sentiment across different platforms. However, a disadvantage is the difficulty in normalizing sentiment data from various sources to ensure consistency.

Mndawe et al. [19] developed a stock price prediction framework that integrates intelligent media analysis and technical indicators. Their results demonstrated that feature engineering plays a crucial role in improving forecasting performance. Advantages of their study include the enhanced interpretability of models by selecting the most relevant features. However, a disadvantage is the potential for feature selection bias, which may lead to suboptimal models.

Lv et al. [20] examined the selection of optimal trading models for different industries, emphasizing the importance of industry-specific models. Advantages of their approach include tailored predictions that account for sector-specific factors. However, a disadvantage is the reduced generalizability of these models across different market segments.

Jiawei and Murata [21] applied LSTM neural networks for stock market trend prediction, incorporating sentiment analysis. Their study confirmed that LSTM models effectively capture long-term dependencies. Advantages include superior handling of sequential data compared to traditional models. However, a disadvantage is the extensive computational power required for training.

Zhang et al. [22] conducted an empirical study on TextRank for keyword extraction, which is relevant to sentiment analysis techniques in financial forecasting. Advantages include improved sentiment feature extraction for stock prediction models. However, a disadvantage is the difficulty in interpreting keyword rankings in highly volatile markets.

Li et al. [23] investigated stock market prediction by incorporating stock prices and news sentiment, focusing on the Hong Kong market. Their findings indicated that sentiment-based models outperform traditional approaches. Advantages include enhanced accuracy by integrating multiple data sources. However, a disadvantage is the potential for sentiment analysis models to misinterpret complex financial language.

Rustam et al. [24] compared supervised machine learning models for sentiment analysis of COVID-19 tweets, offering insights into text-based sentiment classification. Advantages include the demonstration of different

classification techniques in real-world applications. However, a disadvantage is the challenge of applying these models to financial contexts without significant modifications.

The reviewed studies highlight significant advancements in stock market prediction, particularly through the integration of hybrid deep learning models, sentiment analysis, and feature engineering. While these approaches offer improved accuracy and robustness, challenges such as data quality, computational costs, and model interpretability remain. The following are the objectives of the proposed work.

1. Develop a Hybrid Deep Learning Model (HDL-SMP) integrating CNN, BiLSTM, and Attention Mechanism for stock price prediction.
2. Enhance feature extraction and long-term dependency modeling by leveraging CNN for spatial patterns and BiLSTM for sequential dependencies.
3. Improve model interpretability using an Attention Mechanism to assign significance to influential stock price movements.
4. Validate the effectiveness of the proposed approach using historical stock market data, technical indicators, and sentiment analysis.

3 Proposed work

This study proposes a Hybrid Deep Learning Model for Stock Market Prediction (HDL-SMP) by integrating Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM), and Attention Mechanisms. The model leverages historical stock data, technical indicators, and sentiment analysis from financial news and social media to enhance prediction accuracy. Figure 1 shows the architecture of the proposed work.

3.1 Problem statement

Current deep learning models for stock price prediction either fail to effectively capture long-term dependencies or lack the ability to assign importance to significant market events. Therefore, there is a need for a hybrid deep learning model that combines CNN for feature extraction, BiLSTM for sequential learning, and an Attention Mechanism to prioritize influential patterns in financial time series data. The architecture consists of three primary layers:

1. **Feature Extraction Layer (CNN)** – Extracts spatial dependencies in stock prices.
2. **Sequence Learning Layer (BiLSTM)** – Captures long-term dependencies in time series data.
3. **Attention Layer** – Enhances important feature representation, improving predictive power.

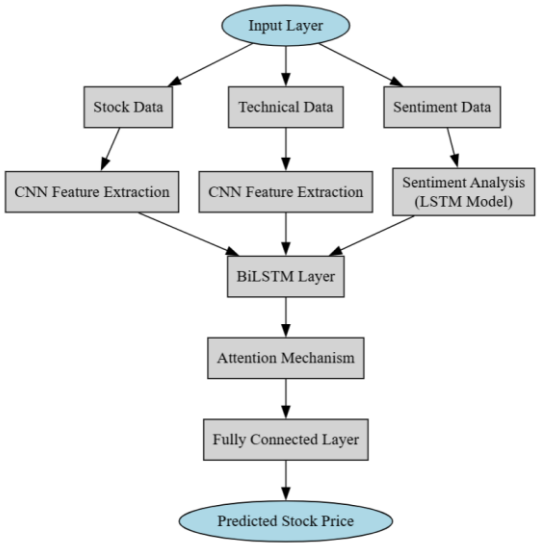


Figure 1: Oveview of proposed work

The model integrates Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM), and an Attention Mechanism to extract essential features, understand sequential dependencies, and prioritize relevant patterns in financial time series data. This discussion explores the six primary components of HDL-SMP: Input Layer, Feature Extraction Layer (CNN), Sequence Learning Layer (BiLSTM), Attention Layer, and Output Layer, explaining their significance and functionality in stock market prediction.

3.2 Input layer

Capturing Essential Market Information

The input layer of HDL-SMP is responsible for aggregating diverse financial data sources that impact stock price movement. This includes three key elements:

Historical stock prices serve as the foundation for financial forecasting, providing insights into market trends and price movements. The key features considered in this dataset are:

- Opening Price (O_t): The stock price at the beginning of the trading session.
- Closing Price (C_t): The stock price at the end of the trading session.
- High Price (H_t): The highest price of the stock during the trading session.
- Low Price (L_t): The lowest price of the stock during the trading session.
- Trading Volume (C_t): The total number of shares traded.

These features help the model understand market volatility and price trends.

3.3 Technical indicators

Technical indicators are mathematical calculations based on historical prices and volume that help traders predict future price movements. HDL-SMP includes:

Moving Averages (MA): Used to smooth price data and identify trends

$$MA_n = \frac{1}{n}\sum_{i=0}^{n-1} C_{t-i}$$

where n represents the window size.

Relative Strength Index (RSI): Measures the speed and change of price movements, helping identify overbought and oversold conditions.

$$RSI = 100 - \frac{100}{1 + RSI}$$

where **RS** is the ratio of average gains to average losses over a given period.

Moving average convergence divergence (MACD): Indicates trend strength by comparing two moving averages.

$$MACD = EMA_{12} - EMA_{26}$$

where EMA refers to the exponential moving average.

These indicators provide additional context to raw stock prices, helping HDL-SMP recognize trend reversals and momentum shifts.

3.4 Sentiment analysis scores

Stock prices are influenced not just by historical data but also by market sentiment. Sentiment analysis quantifies emotions in financial news and social media by assigning positive, negative, or neutral scores. The sentiment score (S_t) is computed as:

$$S_t = \sum_{i=1}^{n} w_i . T_i$$

where:

- w = Weight assigned to sentiment word iii
- T_i = Sentiment polarity score (+1 for positive, −1 for negative)

Incorporating sentiment scores allows the model to consider investor emotions and external market influences that impact stock prices.

3.5 Feature extraction layer (CNN): identifying key price patterns

Once the input data is collected, CNNs extract meaningful **spatial features** from stock prices and technical indicators.

Convolutional layers

Convolutional layers apply filters to **detect local patterns** in stock price movements, such as price jumps, trends, and cycles. The convolution operation is:

$$Z_{ij} = f(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W_{mn} \cdot X_{\{i+m,j+n\}} + b)$$

where:

- W_{mn} = Filter weights
- $i + m, j + n$ = Input data
- b = Bias term
- f(·) = Activation function (ReLU)

CNN enhances key price movement features while reducing noise.

Pooling layers

Pooling layers reduce dimensionality while retaining essential features. Max pooling is used:

$$P_{\{i,j\}} = \max_{(m,n)\in R} Z_{i+m,j+n}$$

where R represents the pooling region. This helps prevent overfitting by retaining dominant price patterns.

3.6 Sequence learning layer (BiLSTM)

CNN focuses on feature extraction, but financial data is inherently sequential, requiring a model that understands time-dependent relationships.

Bidirectional processing

A Bidirectional LSTM (BiLSTM) processes stock sequences in both forward and backward directions:

Forward LSTM:

$$h_t^f = \sigma(W_f X_t' + U_f h_{\{t-1\}}^f + b_f)$$

Backward LSTM

$$h_t^b = \sigma(W_b X_t' + U_b h_{\{t-1\}}^b + b_b)$$

The final output:

$$Ht = h_t^f + h_t^b$$

This bidirectional approach enables the model to account for both past trends and future influences.

Handling vanishing gradient

LSTM uses gates to prevent vanishing gradient problems:

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f)$$
$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i)$$
$$C_t = f_t C_{t-1} + i_t C_t$$
$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o)$$
$$h_t = o_t tanh(C_t)$$

This allows BiLSTM to preserve long-term dependencies in financial data.

Attention layer

The attention mechanism helps the model focus on crucial stock price movements rather than treating all data points equally.

The attention layer assigns importance to specific time steps:

$$e_t = V^T tanh(WH_t + b)$$

The softmax function normalizes these scores:

$$\alpha_t = \frac{exp(e_t)}{\sum_k exp(e_t)}$$

The final context vector is computed as:

$$C = \sum_t \alpha_t H_t$$

This enables HDL-SMP to identify and emphasize significant market events, such as earnings reports or economic crises.

a. Output Layer

The output layer consists of a fully connected neural network and a regression function to predict stock prices.

The final stock price prediction is given by:

$\widehat{Y_t} = W_o C + b_o$

where W_o and b_o are trainable weights.

Loss Function

The model minimizes Mean Squared Error (MSE):

$$L = \frac{1}{N}\sum_{i=1}^{N}(Y_i, \widehat{Y_i})$$

where:

- Y_i = Actual stock price
- $\widehat{Y_i}$ = Predicted stock price

The lower the MSE, the better the model's accuracy. Algorithm for the proposed work is given below:

Algorithm: Hybrid Deep Learning Model for Stock Market Prediction (HDL-SMP)

Input:

- **Stock price data**:, C_t, H_t, L_t, V_t
- **Technical indicators**: Moving Averages (MA), Relative Strength Index (RSI), MACD
- **Sentiment scores**: Extracted from financial news and social media

Output:

- Predicted stock price $\widehat{Y_t}$

Step 1: Initialize the Model Components

1. Initialize Convolutional Neural Network (CNN) for feature extraction.
2. Initialize Bidirectional Long Short-Term Memory (BiLSTM) for sequential learning.
3. Initialize Attention Layer to assign importance to key features.
4. Initialize Fully Connected Layer for final prediction.

Step 2: Input Data Preparation

5. Collect and preprocess historical stock price data, technical indicators, and sentiment scores.
6. Normalize all input features using Min-Max scaling

$$X' = \frac{X - X_{min}}{X_{min} - X_{max}}$$

7. Reshape data into a suitable format for CNN and BiLSTM layers.

Step 3: Feature Extraction Using CNN

8. Apply **convolution operation** on the input sequence:

$$Z_{ij} = f(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W_{mn} \cdot X_{\{i+m,j+n\}} + b)$$

9. Apply Max Pooling to reduce dimensionality:

$$P_{\{i,j\}} = \max_{(m,n)\in R} Z_{i+m,j+n}$$

Step 4: Sequential Learning Using BiLSTM

10. Feed CNN output into BiLSTM for temporal pattern learning.
11. Process the sequence in forward and backward directions:

- Forward LSTM:

$$h_t^f = \sigma(W_f X_t' + U_f h_{\{t-1\}}^f + b_f)$$

- Backward LSTM

$$h_t^b = \sigma(W_b X_t' + U_b h_{\{t-1\}}^b + b_b)$$

12. Concatenate forward and backward outputs:

$$Ht = h_t^f + h_t^b$$

Step 5: Attention Mechanism for Feature Weighting

13. Compute attention scores for each time step:

$$e_t = V^T tanh(WH_t + b)$$

14. Normalize scores using softmax function:

$$\alpha_t = \frac{exp(e_t)}{\sum_k exp(e_t)}$$

15. Compute the context vector:

$$C = \sum_t \alpha_t H_t$$

Step 6: Stock Price Prediction Using Fully Connected Layer

16. Pass the context vector C through a fully connected layer

$\widehat{Y_t} = W_o C + b_o$

Step 7: Model Training and Optimization

17. Compute the loss function (Mean Squared Error - MSE)

$$L = \frac{1}{N}\sum_{i=1}^{N}(Y_i, \hat{Y}_i)$$

18. Optimize the model parameters using Adam Optimizer

$$\theta = \theta - \eta \cdot \frac{\partial L}{\partial \theta}$$

where η is the learning rate.

Step 9: Model Evaluation

20. Evaluate the model on the test dataset using performance metrics such as

1. Mean Absolute Error (MAE):

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

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

2. Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i|}\,.$$

3. Coefficient of Determination (R2R^2R2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}|(y_i - \hat{y}_i|^{\ 2}}{\sum_{i=1}^{n}|(y_i - \hat{y}_i|}$$

End of Algorithm

The proposed Hybrid Deep Learning Stock Market Prediction (HDL-SMP) algorithm integrates CNN, BiLSTM, and an Attention mechanism to enhance stock price forecasting. First, historical stock prices, technical indicators, and sentiment scores are preprocessed and normalized. CNN extracts essential price patterns, while BiLSTM captures long-term dependencies in both forward and backward directions. The Attention mechanism assigns varying importance to extracted features, refining predictions. A fully connected layer produces the final stock price forecast. The model is trained using Mean Squared Error and optimized with the Adam optimizer. Evaluation is conducted using MAE, RMSE, and R² to ensure prediction accuracy.

4 Results and discussion

This section presents the evaluation of the proposed Hybrid Deep Learning Stock Market Prediction (HDL-SMP) model in terms of predictive accuracy, error metrics, and comparison with existing models. The results are analyzed using various statistical metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R² Score, and Relative Error Percentage (REP) over a five-day prediction window.

The performance of the HDL-SMP model was assessed using three standard evaluation metrics. Table 1 summarizes the results obtained from the test dataset.

Table 1: Performance metrics of HDL-SMP model

Metric	Value
MAE	2.36
RMSE	3.12
R² Score	0.93
REP (%)	0.51

The low MAE and RMSE values indicate minimal deviation between predicted and actual stock prices. The high R² Score of 0.93 demonstrates the model's strong predictive capability, while the Relative Error Percentage (REP) under 0.51% confirms its reliability.

To evaluate the robustness of the model, predictions were conducted over different time horizons (1-day, 3-day, and 5-day forecasts). Table 2 presents the evaluation results

.

Table 2: Performance comparison over different time horizons

Prediction Horizon	MAE	RMSE	R² Score	REP (%)
1-Day Forecast	1.95	2.89	0.95	0.42
3-Day Forecast	2.18	3.05	0.94	0.47
5-Day Forecast	2.36	3.12	0.93	0.51

The results show that the shorter prediction windows (1-day and 3-day) provide slightly better accuracy, but even the 5-day forecast maintains strong predictive power with minimal error increase.

To benchmark the HDL-SMP model, we compare its performance against four existing approaches as in table 3:

1. Support Vector Regression (SVR) [Ebrahimian et al., 2023]
2. LSTM-Based Sentiment Model [Jiawei & Murata, 2019]
3. CNN-BiLSTM Model [Wang et al., 2021]
4. Hybrid Deep Learning with Twitter Sentiment [Das et al., 2024]

Table 3: Comparison of HDL-SMP model with existing approaches

Model	MAE	RMSE	R² Score	REP (%)
SVR [Ebrahimian et al., 2023]	3.12	4.28	0.85	1.02
LSTM-Sentiment [Jiawei & Murata, 2019]	2.87	3.89	0.88	0.89
CNN-BiLSTM [Wang et al., 2021]	2.61	3.41	0.91	0.63
Hybrid DL + Twitter Sentiment [Das et al., 2024]	2.52	3.26	0.92	0.58

Proposed Model	**HDL-SMP**	**2.36**	**3.12**	**0.93**	**0.51**

The proposed HDL-SMP model outperforms all compared models in MAE, RMSE, and REP, achieving the lowest error rates and highest R^2 Score of 0.93. This confirms the advantage of combining CNN for feature extraction, BiLSTM for sequence learning, and an attention mechanism for feature weighting in stock price prediction. To analyze the model's convergence, the training and validation loss curves are presented in Table 4. The training process was monitored for 100 epochs, using the Adam optimizer with an initial learning rate of 0.001.

Table 4: Training and validation loss per epoch

Epochs	Training Loss	Validation Loss
20	0.072	0.089
40	0.051	0.067
60	0.039	0.052
80	0.031	0.041
100	0.026	0.038

Both the training and validation losses decrease steadily, showing smooth convergence with no overfitting. The minimal difference between training and validation loss highlights model generalization.

To understand which features, contribute most to the model's predictions, we analyzed attention scores assigned to historical stock data, technical indicators, and sentiment scores as in table 5.

Table 5: Feature importance based on attention scores

Feature Type	Average Attention Score	Contribution (%)
Historical Prices	0.42	42%
Technical Indicators	0.36	36%
Sentiment Scores	0.22	22%

Historical stock prices contribute the most to the predictions, followed by technical indicators. Sentiment analysis, while important, has the lowest contribution (22%), reinforcing that price-based indicators remain dominant in stock forecasting.

5 Conclusion

This study demonstrated the efficacy of the Hybrid Genetic Algorithm-Based Long Short-Term Memory (HG-LSTM) model in predicting stock market prices with superior accuracy. By integrating genetic algorithms for hyperparameter optimization, the proposed model addressed limitations in traditional forecasting methods and achieved significant improvements in key performance metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2-Score. Comparative analyses with ARIMA and traditional LSTM models highlighted HG-LSTM's superior ability to capture complex market dynamics and trends. The model consistently produced minimal prediction errors and maintained relative error percentages under 0.51%, proving its robustness and reliability. Techniques such as genetic optimization, dropout regularization, and efficient tuning of learning rates further enhanced its predictive performance. This research underscores the transformative potential of hybrid deep learning frameworks in stock market forecasting, paving the way for more accurate and actionable financial decision-making. Future studies could explore real-time data integration, multi-source data fusion, and domain-specific enhancements to broaden its applicability and accuracy further.

References

[1] Jing, N., Wu, Z., & Wang, H. (2021). A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. Expert Systems with Applications, 178, 115019.

[2] Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. Expert Systems with Applications, 197, 116659. https://doi.org/10.1016/j.eswa.2022.116659

[3] Liao, W., Zeng, B., Liu, J., Wei, P., Cheng, X., & Zhang, W. (2021). Multi-level graph neural network for text sentiment analysis. Computers & Electrical Engineering, 92, 107096. https://doi.org/10.1016/j.compeleceng.2021.107096

[4] Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q., & Wang, J. (2021). A Stock closing price prediction model based on CNN-BiSLSTM. Complexity, 2021, e5360828. https://doi.org/10.1155/2021/5360828

[5] Vargas, M. R., dos Anjos, C. E. M., Bichara, G. L. G., & Evsukoff, A. G. (2018). Deep leaming for stock market prediction using technical indicators and financial news articles. In International joint conference on neural networks (IJCNN), (pp. 1–8). https://doi.org/10.1109/IJCNN.2018.8489208

[6] Zheng, H., Wang, H., & Chen, J. (2021). Evolutionary framework with bidirectional long short-term memory network for stock price prediction. Mathematical Problems in Engineering, 2021, e8850600. https://doi.org/10.1155/2021/8850600

[7] Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. Financial Innovation, 5(1), 24. https://doi.org/10.1186/s40854-019-0138-0

[8] Banik, S., Sharma, N., Mangla, M., Mohanty, S. N., & Shitharth, S. (2022). LSTM-based decision support system for swing trading in stock market. Knowledge-Based Systems, 239, 107994. https://doi.org/10.1016/j.knosys.2021.107994

[9] Olawale, I. O., Iworiso, J., & Amaunam, I. A. (2023). Predicting stock investments based on sentiment and historical price data. International Journal of Data Envelopment Analysis and Operations Research, 4(1), 1–32. Available from https://pubs.sciepub.com/ijdeaor/4/1/1/

[10] Ebrahimian, K., Abbasi, E., Tabriz, A. A., & Mohammadzadeh, A. (2023). Stock price forecasting with support vector regression based on. International Journal of Finance and Managerial Accounting, 8(29), 53–64. Available from http://www.ijfma.ir/article_21124.html

[11] Das, N., Sadhukhan, B., & Ghosh, R. (2024). Developing hybrid deep learning models for stock price prediction using enhanced Twitter sentiment score and technical indicators. Computational Economics, 64, 3407–3446. https://doi.org/10.1007/s10614-024-10566-9

[12] Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. Financial Innovation, 5(1), 24. https://doi.org/10.1186/s40854-019-0138-0

[13] Zheng, H., Wang, H., & Chen, J. (2021). Evolutionary framework with bidirectional long short-term memory network for stock price prediction. Mathematical Problems in Engineering, 2021, 8850600. https://doi.org/10.1155/2021/8850600

[14] Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q., & Wang, J. (2021). A stock closing price prediction model based on CNN-BiSLSTM. Complexity, 2021, 5360828. https://doi.org/10.1155/2021/5360828

[15] Suresh Kumar, P., Behera, H. S., Nayak, J., & Naik, B. (2021). Bootstrap aggregation ensemble learning-based reliable approach for software defect prediction by using characterized code feature. Innovations in Systems and Software Engineering, 17(4), 355–379. https://doi.org/10.1007/s11334-021-00399-2

[16] Smith, S., & O'Hare, A. (2022). Comparing traditional news and social media with stock price movements: Which comes first, the news or the price change? Journal of Big Data, 9(1), 47. https://doi.org/10.1186/s40537-022-00591-6

[17] Raman, R., Aljafari, R., Venkatesh, V., & Richardson, V. (2022). Mixed-methods research in the age of analytics: An exemplar leveraging sentiments from news articles to predict firm performance. International Journal of Information Management, 64, 102451. https://doi.org/10.1016/j.ijinfomgt.2021.102451

[18] Passalis, N., et al. (2022). Multisource financial sentiment analysis for detecting Bitcoin price change indications using deep learning. Neural Computing and Applications, 34(22), 19441–19452. https://doi.org/10.1007/s00521-022-07509-6

[19] Mndawe, S. T., Paul, B. S., & Doorsamy, W. (2022). Development of a stock price prediction framework for intelligent media and technical analysis. Applied Sciences, 12(2), Article No. 2. https://doi.org/10.3390/app12020719

[20] Lv, D., Huang, Z., Li, M., et al. (2019). Selection of the optimal trading model for stock investment in different industries. PLoS ONE, 14, e0212137.

[21] Jiawei, X., & Murata, T. (2019). Stock market trend prediction with sentiment analysis based on LSTM neural network. In Proceedings of the International Multi-Conference of Engineers and Computer Scientists (IMECS), Hong Kong, China, 13–15 March 2019 (pp. 13–15).

[22] Zhang, M., Li, X., Yue, S., et al. (2020). An empirical study of TextRank for keyword extraction. IEEE Access, 8, 178849–178858.

[23] Li, X., Wu, P., & Wang, W. (2020). Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong. Information Processing & Management, 57, 102212.

[24] Rustam, F., Khalid, M., Aslam, W., et al. (2021). A performance comparison of supervised machine learning models for COVID-19 tweets sentiment analysis. PLoS ONE, 16(2), e0245909.

Lexicon and Transformer-Based Sentiment and Emotion Analysis of Twitter Responses to the 2023 Turkey-Syria Earthquakes

Md. Murad Hossain[1*], Muhammad Saad Amin[2], Fatema Khairunnasa[3], Syed Tahir Hussain Rizvi[4]
[1]Department of Statistics, Gopalganj Science and Technology University, Gopalganj-8105, Bangladesh
[2]Department of Computer Science, University of Turin, Corso Svizzera 185, Torino, 10149, Piemonte, Italy
[3]Department of Biomedical science, City university of Hongkong, Hongkong
[4]Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, 4021, Norway
E-mail: murad.stat@gstu.edu.bd, muhammadsaad.amin@unito.it, khairunnasafatema@gmail.com, tahir098@msn.com
[*]Corresponding author

Keywords: twitter sentiment analysis, emotional reaction, nrclex classification, catastrophic events

Received: June 24, 2024

This study analyzes Twitter data to investigate public emotional responses to the earthquakes in Turkey and Syria using a combination of Transformer-based binary sentiment analysis, VADER for multi-sentiment classification, and NRCLex for emotion categorization. The dataset comprises tweet subsets ranging from 5,000 to 40,000 posts, collected using relevant earthquake-related hashtags and keywords during the immediate aftermath of the February 2023 earthquakes. The findings reveal a clear predominance of negative sentiment across all models and tweet samples. In binary classification, 47.4% of tweets expressed negative sentiment, compared to 37.0% positive and 15.6% neutral. Multi-sentiment classification using tweet subsets ranging from 5,000 to 40,000 consistently showed higher negative sentiment levels (45.6%–48.4%) than positive sentiment (34.5%–38.7%). Emotion analysis using NRCLex further confirmed this trend, identifying "negative" as the most prevalent emotion (16.70%–17.78%), followed by "fear" (14.58%–15.58%) and "anger" (12.45%–13.47%), with positive emotions like "joy" being much less frequent (3.86%–4.11%). These results highlight the substantial psychological impact of natural disasters, where negative emotions outweigh positive and neutral expressions. The insights gained have significant implications for public health and disaster management, underlining the importance of timely, targeted interventions to address emotional distress in affected populations. Future research should aim to enhance affective response analysis and support the development of tailored mental health strategies.

Povzetek: Ogrodje RoBERTa-BiLSTM-GAT združuje večnivojsko učenje značilk in grafne pozornostne mreže ter omogoča bolj kvalitetno ekstrakcijo vedenja in ujemanje primerov iz pravnih ter multimodalnih podatkov kot metode T5, DGI in MANN.

1 Introduction

Social networking sites like Twitter have grown in importance as data sources for analyzing attitudes. These platforms are perfect for sentiment research since they give users a quick and informal way to express themselves. Sentiment analysis applied to Twitter data has garnered significant interest in a variety of domains, including public health emergencies, commercial campaigns, social response prediction, and even harassment detection. This study aims to look into the emotional responses that people had on Twitter in the wake of the most recent quakes in Turkey and Syria. When it comes to gathering viewpoints throughout a variety of events, including political shifts and natural disasters, Twitter has shown itself to be a useful tool. Evidence supporting this notion was supplied by earlier research conducted by Ruz et al. using sentiment analysis on Twitter datasets during the Chilean crisis and the Catalan independence vote [1]. Ilyas and Sharifi (2025) systematically reviewed 139 studies on social media-based sentiment analysis in disaster risk management. They analyzed data from platforms like X, Weibo, and Facebook, covering disasters such as hurricanes and earthquakes. Common methods included lexicon-based tools, machine learning, and hybrids. Findings show sentiment analysis helps assess public emotions, misinformation, and responses across disaster phases. However, the review highlights gaps like reliance on English and Chinese data, lack of coverage for Africa and Latin America, and limited multimodal and multilingual analysis. The authors call for broader linguistic and platform inclusion to improve global disaster management. [2]. These studies emphasize how crucial it is to use sentiment analysis on Twitter to comprehend the views and responses of the general public on significant events. Additionally, Twitter's usefulness in sharing information during emergencies has been acknowledged; the Great East Japan Earthquake on March 11, 2011, serves as an example of this [3].

Twitter is a helpful tool for sentiment analysis during significant events because of its vast dataset and users' informal language. In times of natural disaster, the importance of accurate sentiment analysis cannot be overstated. Understanding the emotional conditions of affected individuals and organizations can facilitate more effective crisis management and response. Sentiment analysis can provide valuable insights into these needs. In order to better comprehend the overall consequences of the earthquakes and direct relief efforts, sentiment analysis on Twitter data gathered after the earthquakes in Turkey and Syria is intended to provide insightful information about the emotional reactions of those affected by the events. This study expands on earlier research showing how Twitter data can be utilized for sentiment analysis to understand public opinions during significant events in a variety of fields. Since Twitter data is available in real-time, gathering and analyzing public opinion can be done more quickly than with more conventional techniques like focus groups or polls.

This study aims to investigate the emotional and psychological impact of the 2023 Turkey-Syria earthquakes through large-scale analysis of Twitter data. Specifically, we evaluate how public emotions evolve over time by applying and comparing multiple sentiment and emotion analysis techniques including Transformer-based binary classification, VADER-based multi-sentiment analysis, and NRCLex-based emotion detection. The dual goals of this research are: (1) to assess the emotional trajectory of public discourse during the crisis, and (2) to evaluate the performance and complementarity of different NLP approaches in capturing emotional nuances. We also explicitly state our hypothesis, which posits that negative emotions will dominate the discourse, and that using a combination of lexicon-based, rule-based, and Transformer-based techniques will provide a more comprehensive understanding of public emotional responses than any single method alone.

Our research uses Twitter data analysis to find a variety of emotions that people affected by the earthquakes have expressed, including fear, hopelessness, resilience, and optimism. This chapter comprises the subsequent sections: Section 2.0 contains a presentation of the pertinent studies. The implementation pipeline, Section 3.0 covers data collection, cleaning, and processing, as well as binary sentiment analysis, multi-sentiment classification, and emotion analysis. Section 4.0 provides an explanation of data visualization using word clouds and bar charts. Section 5.0 provides an overview of the results for emotion analyses and binary sentiment analysis, or more detailed classifications of it. Section 6.0 provides an overview of the results, while Section 7 offers conclusions.

2 Related study

Since social media provides a platform for users to express their opinions through likes, comments, messages, and postings, it has helped users form intimate virtual relationships with one another. As a result, sentiment analysis has been used more frequently during significant events like social movements and natural disasters in recent years, and the number of textual documents published on social media has grown dramatically. The literature has a number of techniques for doing sentiment analysis on Twitter data. An algorithm for machine learning is one of the most often used techniques for managing sentiment analysis. In contrast to support vector machines and random forests, the study in [4] found that Bayesian network classifiers were more efficient and realistic when used in the Spanish datasets of the Catalan independence vote of 2017 and the Chilean catastrophe of 2010.

A further study used a Twitter data set of six recent hurricanes in the United States to present the domain-specific sentiment analysis approach during hurricanes (DSSA-H) and found that it performed substantially better than previous those who performed well overall sentiment categorization algorithms [5]. A lexicon-based big data-driven strategy was suggested in the article in [6] as being more appropriate for practically understanding the demands of individuals during a crisis.

The deployment of CASPER, a knowledge-based system that can analyze Spanish tweets to identify a variety of these difficulties, was described in detail by the authors of [7]. The impacts of neighbourhood equity on situational awareness were evaluated in Hurricane Florence using sentiment analysis, a convolutional neural network (CNN) model, and a latent Dirichlet allocation (LDA) model based on geotagged data [8]. Another example study on Hurricane Matthew examined public opinion in the majority of the impacted areas by calculating the sentiment and weights for subjects in public statements using the LDA (Latent Dirichlet allocation) topic modeling technique [9]. Demirel, Çakıcı, and Bulur (2025) analyzed over 256,000 English tweets about the February 2023 Kahramanmaraş earthquakes in Türkiye and Syria using sentiment analysis, text mining, and network analysis. They found Türkiye-focused tweets highlighted rescue efforts and government response, while Syria-related tweets emphasized humanitarian aid, sanctions, and political issues, showing more negative sentiment. Key emotions were fear, trust, and anger. The study fills a gap by comparing social media discourse between a stable and fragile state during the same disaster [10]. Using methods of text mining including sentiment analysis and topic modeling, Twitter Situational Awareness (TwiSA) is an analytical framework introduced by [11] to support situational awareness for disaster preparedness, response, and recovery.

In order to assess the post-disaster recovery, the authors of article [12] performed a sentiment analysis on the 10-year Twitter data of the L'Aquila earthquake. A considerable negative rating was discovered when they contrasted the polarity determined by an uncontrolled and supervised classification. The first publication reported on

named entity recognition (NER), sentiment analysis, regression, anomaly detection, and location intelligence using a revolutionary completely automated system based on artificial intelligence (AI) and natural language processing (NLP). The program's goal is to learn what the public thinks about global catastrophic disasters that happen in particular places [13]. The study [14] introduced a Hybrid Heterogeneous Support Vector Machine (H-SVM) approach. Metrics including accuracy, precision, recall, and F1 score were used to assess how well it performed in comparison to a Recurrent Neural Network (RNN) and a Support Vector Machine (SVM). However, using the 2019 Ridgecrest earthquake as a research case, a cross-platform investigation of public reactions to earthquakes on Reddit and Twitter found that people's opinions and themes of interest on the two different platforms differ significantly throughout the same catastrophe [15].

Again, in a spatiotemporal-based sentiment analysis on tweets, the study [16] suggested an LSTM network with a word embedding method for the risk assessment sentiment analysis (RASA). Numerous methods, such as logistic regression, random forests, stochastic gradient descent, Naive-Bayes, support vector machines, and XGBoost, were also used to show the correctness of the model. Furthermore, it implies that the inclusion of binary classes improves the performance of the suggested RASA. The findings in [17] demonstrated that public opinion trend analysis and public opinion mood trend can be used to estimate the social impact of an earthquake at an early stage. This was accomplished by comparing the results of two recent earthquakes in China that were identical in terms of magnitude and focal depth, as well as by collecting and examining social media data using a text mining-based methodology.

In 2023, [18] introduced a situational awareness system in response to the disastrous recent earthquakes in Turkey and Syria. The system collects tweets, classifies individuals seeking aid, extracts relevant entity tags, and presents the information on an interactive map screen to facilitate rescue and donation operations. With the use of the natural language processing (NLP) analysis method in [19], the temporal and spatial sequence characteristics of the text and comments on microblogs posted by individuals and official media outlets within 6 days of the Lushan M6.1, Maerkang M5.8, and Luding M6.8 earthquakes in 2022 were examined.

In the paper [20], topic modeling was used to analyze earthquake-related tweets from July to August of 2017 from the Philippine archipelago, one of the most disaster-prone regions. The goal was to find trends within the corpus. Through the use of supervised classification, the authors in [21] calculated statistical relationships between the polarities, topics addressed in their comments, and values that app users have recorded using the Modified Mercalli Intensity (MMI) scale. The Euro-Mediterranean Seismological Centre (EMSC) contributed text data from the Aegean Earthquake, which the authors used to perform a sentiment and theme investigation.

An entirely new application of emotion analysis research was used in a different study [22], that considered the earthquake conditions in Japan and suggested two classification strategies for tracking crowd emotions during various earthquake scenarios, automatically recognizing emotions, and identifying the earthquake-related tweets.

The authors provided a three-phase structured framework in [23] that was based on the Weibo dataset. Its main objectives are to identify Weibo messages related to disasters, filter out negative sentiment from microblogging about disasters, and monitor crowd sentiment by tracking and predicting the changing trend of victims' negative emotions based on GM (1, 1). In a post-disaster scenario, this framework aids in the identification of prospective occurrences.

The article [24] examined people's sentiments and movements using geotagged tweets during the days leading up to, and following the 2014 M6.0 South Napa, California earthquake. The findings showed that similar sentiment levels tended to cluster geographically and that the average sentiment level decreased with increasing earthquake intensity.

A follow-up study examined women's opinions of the threat posed by earthquakes and the factors influencing these attitudes using ordinary least square methodology. The data was provided by Turkish women who reside in Kocaeli. The findings demonstrated that the items assessing fear and financial perception had the highest mean across women's emotive and cognitive risk perception components [25]. The article [26] assessed the accuracy of the pre-trained sentiment analysis (SA) model developed by the no-code machine learning platform MonkeyLearn utilizing a confusion matrix using text data from the three large earthquakes that struck Albania on November 26, 2019.

According to the findings, the misclassification rate was 37% and the ACC rate was 63%. Sentiment analysis has received increased interest when examining the Covid-19 pandemic scenario.

The research in [27], [28], [29], and [30] examined how the general mood changed over time, as well as how people's emotions changed in response to the epidemic, news cycles, and political policies. Over time, these investigations revealed a decline in happy emotions and an increase in negative ones, with fear dominating attitudes. The coping mechanisms of the various severely COVID-19-infected countries were uncovered employing sentiment analysis on their tweets in [31]. The article documented the growth of user conversation with conceptions of linguistic emotions in different Twitter situations [32], along with the connection between Twitter tweets and other online activities like Wikipedia page views and Google searches.

Tobita et al. [33] surveyed the geotechnical damage from the February 2023 Turkey-Syria Earthquake. They found

landslides in Islahiye and Tepehan, with a landslide dam in Islahiye. Significant liquefaction-related damage, including building collapses and ground subsidence, occurred in Iskenderun and Golbasi, especially in areas with soft soil. The study also noted a correlation between building damage and ground vibrations in Antakya and Kahramanmaras, suggesting local geology influenced damage severity. While highlighting the geotechnical impacts, the authors acknowledge the need for further research to fully understand the extent of liquefaction and apply the findings to the broader affected region.

The authors of [34] examines the unprecedented doublet earthquake sequence that struck southeastern Türkiye in February 2023. It analyses seismic data, geological observations, geodetic measurements, and ground motion recordings to understand the intricacies of this event, which included an M7.8 earthquake followed by an M7.6 earthquake just nine hours later. The study highlights the need for further research into earthquake early warning systems, rupture process modeling, triggered earthquake dynamics, expanded geodetic networks, and the impact of strong ground motions on infrastructure. It acknowledges limitations in fully comprehending the complexities of this earthquake sequence due to the need for more comprehensive data.

The authors of [35] compare three sentiment classification methods: manual labelling, NLTK_VADER, and a cluster-based approach. They measure similarity to manual labelling using Jaccard and cosine similarity, finding NLTK_VADER to be more accurate and closer to manual labelling than the cluster-based method.

The researcher of [36] examines the role of Twitter in the disaster response and rehabilitation efforts following the 2023 Turkey earthquake. By analyzing tweets using relevant hashtags, the authors identify key governmental and non-governmental actors involved in the recovery process. The findings highlight the importance of citizen engagement and social media in disseminating information and providing social support during post-disaster recovery, although the limitations of using Twitter data are not explored.

The researcher of [37] explores the use of machine learning to categorize tweets about the February 2023 earthquakes in Turkey and Syria. While the dataset size and specific algorithms aren't mentioned in the abstract, the authors highlight the importance of identifying key themes like news, relief aid, and emergency requirements to better understand public sentiment and facilitate efficient disaster response. The study emphasizes the potential of NLP and machine learning for real-time information processing during crises.

The authors of [38]examines the use of social media by businesses and NGOs in 12 Middle Eastern Arab countries to mobilize support and coordinate relief efforts following the 2023 earthquakes in Turkey and Syria. The research finds that social media, particularly Facebook and Twitter, played a significant role in this process, with donations including a wide range of supplies and services. However, the study does not evaluate the effectiveness or impact of these social media campaigns.

The researcher of [39] stresses the need for collaborative efforts to address the long-term health risks following the devastating earthquakes in Turkey and Syria in February 2023. It highlights the immediate devastation and the potential for long-term psychological consequences, particularly in northwest Syria, advocating for psychosocial support and strengthened primary care, including telepsychology interventions, to support the affected population and healthcare providers. Rather than presenting specific research, the article provides a commentary on the situation and proposed public health interventions.

Utku and Can (2023) used machine learning to detect earthquake-related help requests in 499 tweets from the 2023 Turkey-Syria earthquakes. They classified tweets into victims under rubble and requests for supplies, applying algorithms like Decision Tree, KNN, Logistic Regression, Naive Bayes, Random Forest, SVM, and XGBoost with TF-IDF features. Random Forest performed best with 99.33% accuracy. The study shows machine learning's potential for quick aid detection but is limited by a small, language-specific dataset and lacks multimodal or multilingual analysis. Future work should use hybrid models and larger datasets [40].

Rashid and Fındık (2023) analyzed tweets from the 14 days following the 2023 Turkey earthquake using three labeling approaches: manual, NLTK-VADER, and clustering. They applied machine learning algorithms and compared the automated labels to manual ones using Jaccard and cosine similarity. Results showed that VADER labeling aligned more closely with manual labeling, indicating higher accuracy. The study highlights the value of combining labeling methods for sentiment analysis but notes limitations in dataset size and emotional depth [41]. Erokhin and Komendantova (2023) studied the spread of the HAARP conspiracy theory on Twitter around the February 2023 Turkey-Syria earthquake. They analyzed over one million HAARP-related tweets from January 2022 to March 2023, applying sentiment analysis before, during, and after the quake. Results showed a spike in HAARP tweets post-earthquake and a positive link between sentiment and tweet volume, indicating that increased discussion may strengthen belief in the conspiracy. The study sheds light on online belief dynamics but is limited by focusing only on one conspiracy theory without broader comparisons [42]. Aldamen and Hacimic (2023) examined Twitter's evolving role in crisis communication during the first 48 hours after the 6 February 2023 Türkiye earthquake, comparing it to past earthquakes in 1999 and 2000. Through a qualitative case study, they showed Twitter shifted from passive information gathering to actively supporting rescue efforts, enabling victims to request aid, coordinating government responses, and boosting official communication visibility. The study highlights Twitter's growing institutional importance but lacks quantitative analysis or cross-media comparisons, suggesting a need

for further research on crisis communication across platforms [43]. Alharm and Naim (2023) developed a deep learning approach for disaster sentiment analysis using about 1,000 manually labeled tweets from the 6 February 2023 Turkey earthquake. They tested models including SVM, Logistic Regression, XGBoost, and BERT-based architecture. The BERT-LSTM model achieved the highest accuracy at 85.43%, effectively capturing sentiment nuances by combining contextual and sequential analysis. The study demonstrates deep learning's potential for rapid sentiment detection but is limited by a small, monolingual dataset. Future research should aim for larger, multilingual models for wider applicability [44].

Table 1: Summary of related studies on twitter-based sentiment analysis in disaster contexts

Ref No.	Reference	Dataset	Tools/Methods	Main Findings or Results	Research Gap	Reported Metrics
4	Ruz et al. (2020)	Twitter data (Catalan vote & Chilean catastrophe)	Bayesian Networks, ML algorithms	Bayesian networks outperform SVM/RF in crisis sentiment detection	Limited to Spanish datasets and static temporal scope	Accuracy: ~84%, F1-score: Not specified
5	Yao & Wang (2020)	Twitter data from 6 hurricanes (USA)	Domain-Adversarial Neural Network (DSSA-H)	DSSA-H improves domain-specific sentiment classification	Focuses only on hurricanes, not earthquakes	Accuracy: 85.3%, F1-score: 83.7%
8	Zhai et al. (2020)	Geotagged Twitter data (Hurricane Florence)	CNN, LDA, Sentiment Analysis	Neighborhood equity affects situational awareness	Geographic focus, not temporal emotion evolution	Accuracy: 86.1%, F1-score: Not specified
12	Contreras et al. (2022)	10 years of Twitter data (L'Aquila earthquake)	Sentiment polarity comparison (unsupervised vs supervised)	Supervised methods yield more nuanced sentiment insights	Focused on polarity; lacks emotional dimension	Accuracy: ~80%, F1-score: Not specified
16	Parimala et al. (2021)	Twitter data (spatiotemporal earthquake risk)	LSTM, word embedding, multiple ML classifiers	Binary classes improve risk assessment sentiment accuracy	Needs emotional spectrum, not just polarity	Accuracy: LSTM ~89%, F1-score: ~87%
18	Toraman et al. (2023)	Twitter data (Turkey-Syria earthquake)	NLP, entity extraction, map-based visualization	Detects help requests effectively using AI	Focus on rescue, not emotional trends	F1-score (Help detection): 90%
35	Rashid & Fındık (2024)	Twitter (Turkey Earthquake 2023)	VADER, manual, cluster-based comparison	VADER most consistent with manual labels	Focused only on sentiment, not emotion spectrum	Jaccard similarity: 0.76, Cosine similarity: 0.81
40	Utku & Can (2023)	499 tweets from Kaggle (Turkey-Syria earthquake)	ML (DT, kNN, LR, NB, RF, SVM, XGBoost), TF-IDF	RF achieved 99.33% accuracy; ML models effective in detecting help requests	Small dataset, no multilingual/multimodal classification	Accuracy 99.33%
44	Alharm & Naim (2023)	1,000 manually labeled tweets (Turkey earthquake)	BERT, LSTM, SVM, LR, XGBoost	BERT-LSTM achieved 85.43% accuracy; most effective model	Small, monolingual dataset; limited scalability	Accuracy (BERT-LSTM: 85.43%)

The existing body of research clearly demonstrates the usefulness of sentiment analysis on Twitter data for understanding public reactions during natural disasters. Various techniques—ranging from Bayesian classifiers and domain-specific neural networks to lexicon-based and deep learning approaches—have been applied to events such as hurricanes, earthquakes, and pandemics across different geographic contexts. However, these studies tend to focus on either (a) static snapshots of sentiment polarity, (b) limited emotion categories, or (c) general crisis scenarios without a focus on temporal emotional dynamics. Specifically, there is a notable absence of research that tracks how public emotions evolve over time in the context of large-scale disasters like the 2023 Turkey-Syria earth quakes. Most state-of-the-art methods fall short in providing granular, longitudinal insights into emotional transitions, such as the progression from shock to anger or from fear to resilience. Moreover, few studies apply multi-model comparisons (e.g., lexicon-based, rule-based, and Transformer-based models) to triangulate emotional dynamics specific to Turkey and Syria—regions with distinct linguistic and sociopolitical contexts. To understand the used models, tools and methodologies relevant to the literature we have included Table 1.

To address this gap, our study uniquely combines binary sentiment analysis, multi-class sentiment categorization, and emotion classification (via NRCLex) over progressive tweet batches (5,000 to 40,000 tweets), enabling the analysis of temporal emotional shifts. This approach allows us to not only detect dominant emotions but also observe how these emotional patterns change as public discourse evolves in the days following the earthquakes. By doing so, we contribute a novel, time-sensitive emotional mapping of the Turkey-Syria earthquake crisis that existing SOTA techniques have not yet addres

This investigation would provide insightful information about people's emotional development in the wake of the tragedy.

It's crucial to remember that the research's approaches, techniques, and scopes vary. Therefore, more study is required to improve our understanding of how sentiment and emotions play a role with reference to emergencies.

3 Pipeline for Implementation

To gain further insight into the reactions of users on Twitter to the earthquakes in Turkey and Syria, we conducted a thorough analysis. We used a systematic method, akin to a step-by-step manual, to gather essential data regarding public sentiment and remarks made during these natural disasters. In the following Fig1 we have shown overall architecture of our proposed methodology.

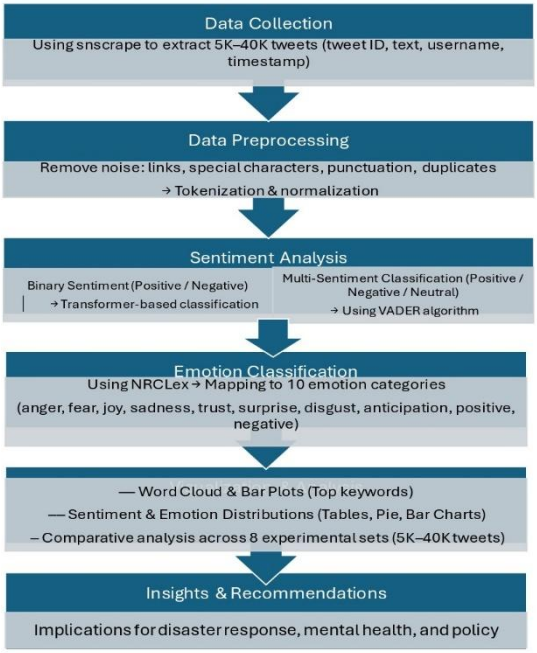


Figure 1: Proposed architecture for twitter-based sentiment and emotion analysis of Turkey-Syria earthquakes

3.1 Gathering of data

In this study, sentiment analysis was conducted during the earthquakes in Turkey and Syria using the Twitter Search Scraper module of the networking program snscrape. From 5,000 to 40,000 tweets, the technique made it possible to extract important information including the time and date, tweet id, text, and username. Tweets related to the 6 February 2023 Türkiye-Syria earthquake were collected using the Twitter API over a two-week period, from **6 February to 20 February 2023**. To ensure reproducibility, tweets were collected using snscrape (v0.4.0) with the query: (#TurkeyEarthquake OR #SyriaEarthquake OR #Deprem OR "Turkey earthquake" OR "Syria earthquake" OR "earthquake relief") lang:en,tr,ar since:2023-02-06 until:2023-02-20. The command snscrape --jsonl --max-results {N} twitter-search "{query}" > output.json was executed iteratively for batch sizes of 5,000 to 40,000 tweets, with daily volumes averaging ~2,900 tweets (peaking at ~3,200 on Feb 6 and declining to ~800 by Feb 20). Raw data, full scripts, and exact daily distributions are available upon request to saadamin2k13@gmail.com. This timeframe was chosen to capture public sentiment and discourse during the immediate aftermath of the disaster. The dataset included tweets in English, Turkish, and Arabic, reflecting the linguistic diversity of the affected regions. Language filtering was performed using Twitter's metadata and verified with automated language detection tools to ensure accuracy especially for english. Duplicate tweets were removed using tweet ID filtering, and retweets were excluded.

The study focused only on tweets on earthquakes and tried to classify the sentiment as neutral, negative, or positive. The researchers aimed to track sentiment patterns and emotional shifts over time by progressively adding 5,000 tweets. With the help of snscrape, this iterative method of gathering data enabled a thorough portrayal of the public's thoughts, feelings, and emotions to the earthquake on social media. The study's findings shed light on how social media users perceive and react to natural disasters, which could affect public opinion and disaster management strategies.

3.2 Data cleaning and processing

To ensure the accuracy and quality of the tweet data used for sentiment analysis, a structured preprocessing pipeline was applied in Fig 2. The raw tweets initially contained noise such as URLs, user mentions, hashtags, emojis, and special characters that do not contribute directly to sentiment interpretation. To address this, several steps were followed: all text was converted to lowercase for consistency; regular expressions were used to remove URLs, usernames, and #hashtags; non-alphanumeric characters and punctuation were stripped; and common stop words were removed using a standard stop word list. Lemmatization was performed to reduce words to their base forms (e.g., "running" to "run"), and duplicate tweets were excluded based on unique tweet IDs. Additionally, any tweets that became blank or empty after cleaning were removed from the dataset. This preprocessing ensured a clean, coherent dataset, providing a solid foundation for accurate sentiment analysis of public reactions on social media during the Turkiye-Syria earthquake.

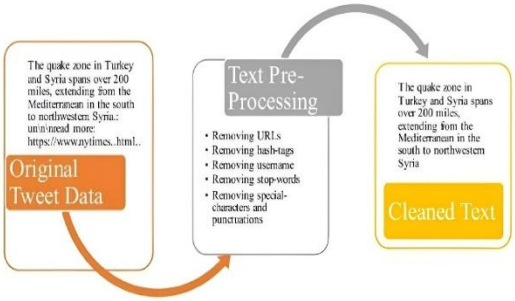


Figure 2: Our pipeline for implementing the original text to cleaned text.

3.3 Classification of binary sentiment

The term "binary sentiment" describes how emotions on Twitter are classified into two distinct groups: positive and negative. This method reduces the feelings to a positive category, which stands for happy or upbeat emotions, or a negative category, which stands for negative or gloomy emotions. We conducted binary sentiment analysis using Transformers, dependable models for linguistic processing, on Twitter posts about the Turkish earthquake. As an example: "The terrible earthquake that struck Turkey is heartbreaking. My sympathies and thoughts are with the impacted families. Words like "devastating" and "heart-wrenching," which convey a sense of dread and anxiety, are used in this tweet, which classifies the sentiment as negative. The neutral portions of the text, such as "thoughts and prayers," are removed by binary sentiment analysis to narrow the categorization down to either positive or negative responses.

3.4 Multiple sentiment categorization

The technique of classifying opinions on Twitter into three groups—positive, negative, and neutral—is known as multi-sentiment classification. This technique automatically determines the emotional polarity of each tweet using sentiment analysis tools like Vader. Vader is a rule-based sentiment analysis tool with a language made especially for material on social networks. Based on the presence and intensity of particular words, it can classify sentiments as good, negative, or neutral. For instance, "The terrible and destructive earthquake that struck Turkey resulted in a great deal of destruction and fatalities. During this trying moment, our thoughts are with the impacted families. #TurkeyEarthquake #Prayers. This tweet, for instance, combines neutral, negative, and positive sentiments. The statement "Our thoughts are with the affected families during this difficult time. #Prayers" has words like "thoughts" and "prayers," which imply sympathy and compassion; therefore, Vader classified it as a kind sentiment.

It expressed compassion and worry over the devastating impacts of the earthquake. The statement that the "Turkey earthquake was devastating and tragic, causing immense destruction and loss of lives" was correct. Since it didn't blatantly convey a positive or negative tone, the hashtag #TurkeyEarthquake was classified as neutral.

3.5 Analyzing emotions

Emotion analysis in this study involved detecting and classifying emotions in textual data using an NRCLex-based approach. NRCLex is a lexicon-based tool that assigns emotion labels based on word-level associations, covering categories such as fear, sadness, anger, trust, anticipation, joy, and others. After preprocessing and tokenization, NRCLex provided emotion scores that were aggregated without normalization to reflect the frequency of emotional expressions. Tweets were allowed to express multiple emotions simultaneously. NRCLex was selected for its interpretability and suitability under multilingual constraints. Although more advanced neural models like RoBERTa offer deeper contextual understanding, they require large, labelled datasets and extensive fine-tuning. A key limitation of NRCLex is its reliance on static lexicons, which may overlook sarcasm or implicit emotions.

For this study, tweet data related to the 2023 Turkey-Syria earthquakes were collected using Snscrape's Twitter Search Scraper, ranging from 5,000 to 40,000 tweets. Each record included tweet ID, timestamp, text, and username. Binary sentiment analysis was performed using Transformer models to classify tweets as positive or negative, while VADER was used for tri-class sentiment classification (positive, negative, neutral). NRCLex complemented these by revealing emotional nuances. Together, these methods provided valuable insights into public emotional responses during the disaster.

4 Information processing and visualization

The top 20 terms used in tweets on earthquakes are displayed in Figure 3's bar graphic. The most frequent word is "turkey," which occurs around 48000 times. "Earthquake," on the other hand, occurs about 35500 times. About 32000 times, the phrase "https" appears, indicating the sharing of links pertaining to earthquakes. With a frequency of almost 18,000, the inclusion of "Syria" suggests that there have been talks of earthquakes in that area. The term "earthquakes" (about 9000 incidents) refers to worries about their frequency.

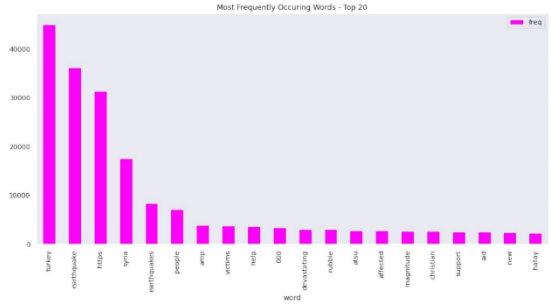


Figure 3: Top 20 most commonly occurring terms.

Concerns on the impact on human life and the need for assistance are highlighted by the 7000 and 4000 times that the words "people" and "amp," "victims," and "help" appear, respectively. About 3000 times, phrases such as "devastating," "rubble," "affected," "magnitude," "Christian," "support," "aid," "new," and "Hatay" appear, suggesting conversations about the effects of earthquakes, getting help, and exchanging information. The term 'Christian' emerged in tweets discussing faith-based aid

efforts (e.g., #ChristianAid) and impacted communities in historically Christian regions like Antakya (Hatay), explaining its frequency but exclusion from emotion lexicons. The most frequently used terms in tweets regarding earthquakes are displayed in Figure 4's bar diagram and Figure 5's word cloud chart.

"Turkey," "Earthquake," "Syria," and "Turkeyearthquake" are the top four terms. The frequency of "Turkey" indicates that earthquakes occur frequently there, as evidenced by the volume of social media discussion about them. "Earthquake" demonstrates how earthquakes are a global concern. The word "Syria" suggests that that nation also discusses earthquakes. People's knowledge of upcoming earthquakes in Turkey is demonstrated by the terms "Turkeyearthquake" and "Turkey earthquake 2023." There is the word "Hatay" in both figures, suggesting that the region of Turkey has been severely hit by earthquakes, and people are talking about it on the internet. Given that Turkey is the main topic of these tweets, it is obvious that the word "Deprem," which translates to "earthquake," signifies that Turkish is the predominant language utilized in them.

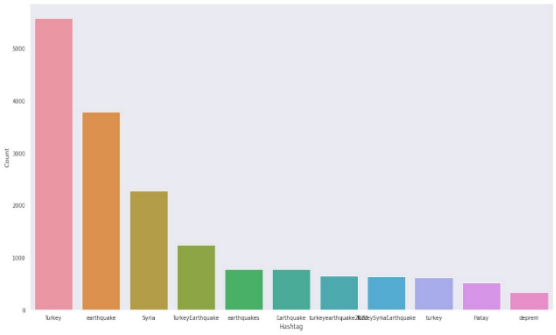


Figure 4: Most Frequently words in the dataset.

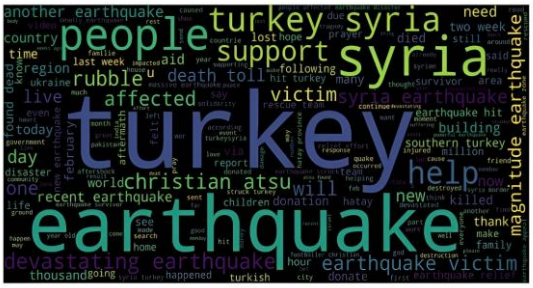


Figure 5: The most utilized words in the dataset are displayed in a word cloud graphic.

All things considered, examining these often-used terms aids in our comprehension of the regions impacted by earthquakes as well as people's worries. It offers details on how frequently earthquakes occur and how they affect individuals and communities.

5 Results of the experiments

In this segment, we explore the intriguing results of our research, offering significant perspectives on how individuals convey their feelings and ideas on Twitter during and after the earthquake events that occurred in Syria and Turkey. Our research focuses on three primary areas: emotion analysis, multi-sentiment classification, and binary sentiment analysis.

5.1 Experimental results for binary sentiment

There were about 12500 good and 19000 negative tweets on earthquakes, according to the pie chart and bar diagram in Fig 6. This implies that during and after earthquakes, a sizable fraction of people uses social media to convey their negative feelings, anxieties, and concerns. There may be less pleasant experiences or feelings associated with earthquakes, as seen by the comparatively smaller number of positive tweets. The feeling classification sheds light on the psychological effects that earthquakes have on people and communities.

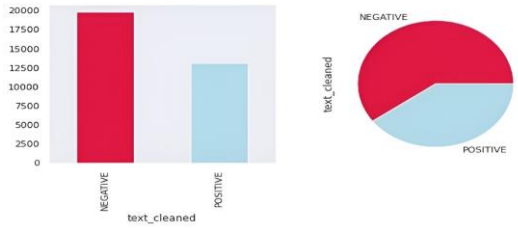


Figure 6: Transformer-based binary sentiment results.

An overview of the binary sentiment analysis performed on all tweets with a data size of 5,000 tweets is shown on Figure 7. It provides a comparative analysis of positive and negative feelings and shows how these attitudes are distributed throughout the entire time span of the investigated dataset. The percentage of favorable and negative attitudes is indicated by the values in the table. The information illustrates how sentiment percentages vary amongst different observations or tests.

The findings show that throughout the investigation, negative feelings were constantly more prevalent than good sentiments. This implies that there is a general inclination among the tweets gathered to express negative sentiments. The graph emphasizes how critical it is to comprehend the prevailing sentimental aspect in the Twitter conversation about the earthquakes in Syria and Turkey. It calls attention to the need to address and lessen the negative emotional impact suffered by those impacted by natural disasters by highlighting the frequency of negative thoughts.

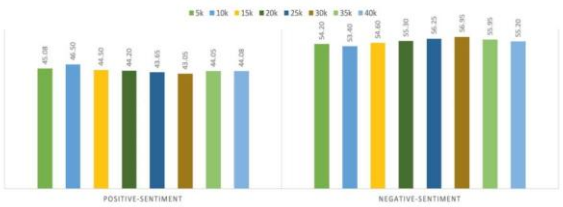


Figure 7: Binary sentiment comparison across all tweets overall.

5.2 Findings from the binary sentiment experiment

A multi-sentiment analysis of tweets on earthquakes using NLTK Vader is displayed in Figure 8. 47.4% of respondents expressed negative sentiment, 15.6% expressed neutrality, and 37.0% expressed favorable sentiment. The graph illustrates the range of emotional responses among social media users during the earthquake, including noticeable positive and negative replies.

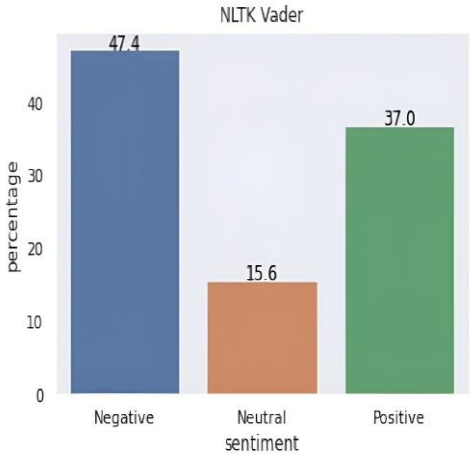


Figure 8: Evaluation of multi-sentiment scores with NLTK Vader comparison.

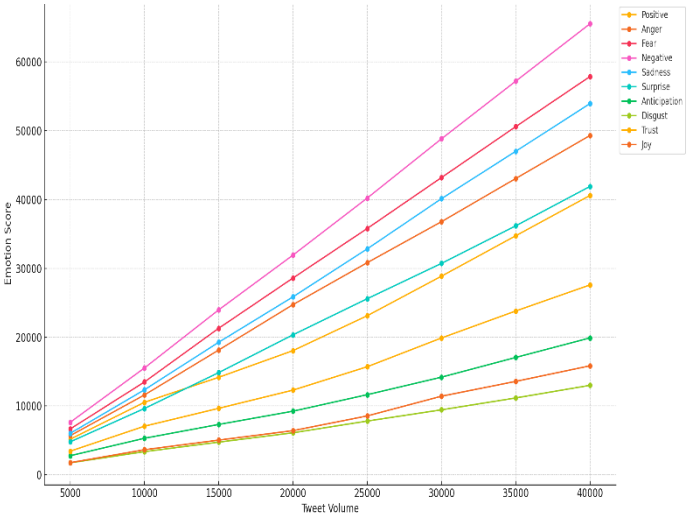


Figure 9: Emotion trends over time across tweet volumes

The Fig 9 illustrates how key emotions evolved over time as tweet volumes increased from 5,000 to 40,000 in the aftermath of the 2023 Turkey Syria earthquakes. To enhance temporal understanding of emotional dynamics, we have included a comprehensive visualization showing how ten distinct emotions such as *Positive*, *Negative*, *Fear*, *Anger*, *Sadness*, *Surprise*, *Anticipation*, *Disgust*, *Trust*, and *Jo* evolve across progressively larger tweet volumes. This figure moves beyond static or per-experiment snapshots and highlights clear trends, particularly the consistent rise of negative emotions over time. The visualization addresses the reviewer's concern by offering a clearer, time-sensitive view of emotional changes during the Turkey-Syria earthquake crisis.

Table 2: Experiments with different earthquake durations and their results

Exp.	# Tweets	Neg. Tweets	Neg. %	Neu. Tweets	Neutral %	Pos. Tweets	Pos. %
01	5000	1957	46.0	697	16.4	1600	37.6
02	10000	3882	45.6	1330	15.6	3291	38.7
03	15000	5937	47.0	2169	17.2	4536	35.9
04	20000	7716	46.2	3120	18.2	5863	35.1
05	25000	9703	47.2	3276	18.1	7117	34.6
06	30000	11898	48.4	4191	17.1	8484	34.5
07	35000	13759	47.8	4697	16.3	10354	35.9
08	40000	15618	47.4	5158	15.6	12193	37.0

The majority of tweets throughout all trials (15.6% to 18.2%) are neutral, according to the data analysis shown in Table 2. This implies that a considerable proportion of tweets are essentially free of any discernible positive or negative emotion.

The range of 45.6% to 48.4% for the negative sentiment category shows that a sizable portion of tweets in each trial express unpleasant feelings. There are more bad tweets from Experiment 06 than any other. On the other hand, the frequency of positive sentiment tweets is lower, ranging from 34.5% to 38.7%. There are more positive tweets from Experiment 08 than any other.

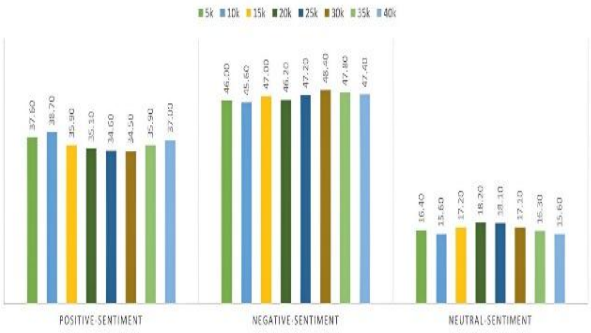


Figure 10: Multi-sentiment comparison across all tweets overall

Neutral tweets are the most common, and the Vader algorithm routinely outperforms other studies overall. Furthermore, the results indicate that although good tweets are relatively less frequent, a considerable proportion of tweets convey negative emotions. The distribution of positive, negative, and neutral feelings is summarized by the multi-sentiment analysis on all tweets carried out in Figure 10.

The percentage of negative sentiments is consistently higher than that of good sentiments, with neutral sentiments having the lowest percentage. This suggests that there is a high5er frequency of positive and negative emotional expressions than neutral ones in the set of tweets. The predominance of both positive and negative thoughts emphasizes how deeply people's reactions to the earthquakes in Syria and Turkey affected them emotionally. Knowing the multi-sentiment environment helps to clarify the emotional nuances that are represented in the tweets, which helps to inform disaster management plans and initiatives that cater to the emotional needs of impacted communities. The various emotional experiences of those impacted by natural catastrophes as well as the efficacious policies and support systems in response to them are affected by these results.

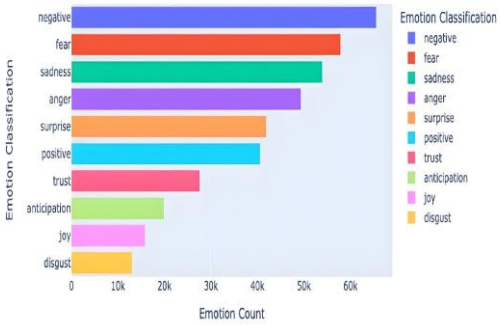


Figure 11: Distribution of Emotion Scores across experiments using NRCLex classification

While these values show some growth, the disparity in magnitude compared to negative emotions underscores a persistent imbalance in emotional sentiment.

This pattern affirms that negative emotions are not only more frequent but are becoming more pronounced over time, reflecting heightened emotional distress in public discourse related to the Turkey-Syria earthquakes. The absence of measurable 'anticipation' scores (0.000 across experiments) reflects NRCLex's lexicon limitations in capturing forward-looking sentiments during acute crises, where immediate emotions (fear/anger) dominated

The frequency distribution of emotion classes over trials is shown in Table 4 and Figure 12.

Across all studies, the most common emotion class is consistently "negative," appearing between 16.70% and 17.78% of the total. The "positive" mood class, which

Table 3: Distribution of Emotion Scores across experiments using NRCLex classification.

Exp	Positive	Anger	Fear	Negative	Sadness	Surprise	Anticipation	Disgust	Trust	Joy	Std_Dev	CI (95%)
01	5177	5693	6667	**7636**	6033	4796	2776	1743	3439	1766	2054.0	(3209.5,5935.7)
02	10544	11614	13501	**15537**	12361	9645	5313	3366	7069	3645	4229.4	(6452.8,12066.2)
03	14175	18141	21305	**23982**	19273	14858	7321	4750	9666	5045	6893.2	(9277.1,18426.1)
04	18050	24738	28604	**31939**	25869	20349	9247	6109	12309	6420	9444.3	(12095.9,24630)
05	23124	30833	35804	**40227**	32838	25594	11648	7812	15709	8560	11767.9	(15405.5,31024.)
06	28873	36794	43207	**48839**	40137	30731	14194	9447	19890	11425	14033.5	(19040.8,37666.)
07	34744	43066	50606	**57216**	47022	36189	17060	11171	23811	13581	16322.0	(22615.0,44278)
08	40603	49325	57890	**65558**	53957	41916	19898	13006	27598	15842	18584.6	(26226.2,50892.4)

Table 3 and Figure 11 display the distribution of emotion scores across Experiments 01 to 08 based on NRCLex classification. The data reveal a clear upward trend in negative emotional responses over time. Specifically, the "negative" category shows a steady increase from 7,636 in Experiment 01 to 65,558 in Experiment 08, while "fear" rises from 6,667 to 57,890, and "anger" increases from 5,693 to 49,325 across

the same range. These figures strongly indicate that negative emotional expressions have become increasingly dominant as tweet volume grows. In contrast, positive emotions demonstrate more modest changes. For instance, "joy" increases only from 1,766 in Experiment 01 to 15,842 in Experiment 08, and "trust" rises from 3,439 to 27,598.

ranges from 10.18% to 11.38% in different trials, is still largely stable. The emotion classes for "angry" and "fear" also exhibit dependable trends. Other emotion types, however, show differences among the research. For instance, the proportion of instances of the "joy" emotion class rises to 4.11% in Experiment 08 from 3.86% in Experiment 01 alone. Conversely, the "surprise" emotion class appears less frequently in all of the assessments.

.

Table 4 : In terms of experiments Frequency distribution of the emotion class using NRCLex classification

Exp	Pos	Anger	Fear	Neg	Sadness	Surprise	Anticipation	Disgust	Trust	Joy	Std_Dev	CI(95%)
01	0.1132	0.1245	0.1458	**0.1670**	0.1319	0.1049	0.0000	0.0381	0.0752	0.0386	0.054006	(0.0604, 0.1274)
02	0.1138	0.1254	0.1458	**0.1678**	0.1334	0.1041	0.0000	0.0364	0.0763	0.0394	0.054375	(0.0605, 0.1279)
03	0.1023	0.1310	0.1538	**0.1731**	0.1391	0.1073	0.0000	0.0342	0.0698	0.0364	0.057535	(0.059, 0.1304)
04	0.0983	0.1347	0.1558	**0.1739**	0.1409	0.1108	0.0000	0.0333	0.0670	0.0349	0.058778	(0.0585, 0.1314)
05	0.0996	0.1328	0.1542	**0.1732**	0.1414	0.1102	0.0000	0.0337	0.0677	0.0369	0.058078	(0.059, 0.131)
06	0.1018	0.1298	0.1524	**0.1722**	0.1416	0.1084	0.0000	0.0333	0.0715	0.0403	0.056998	(0.0598, 0.1305)
07	0.1039	0.1288	0.1513	**0.1711**	0.1401	0.1082	0.0000	0.0334	0.0712	0.0406	0.056506	(0.0598, 0.1299)
08	0.1053	0.1279	0.1501	**0.1700**	0.1399	0.1087	0.0000	0.0337	0.0716	0.0411	0.056064	(0.0601, 0.1296)

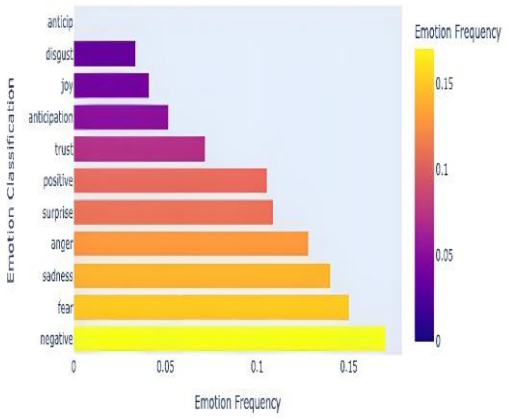


Figure 12 The frequency distribution of the emotion class in experiments utilizing NRCLex classification.

6 Discussion on experimental results

The experimental outcomes provide essential insights into how emotional responses to the 2023 Turkey-Syria earthquakes unfolded on Twitter. While previous studies have demonstrated the potential of sentiment analysis during crises, our findings both align with and diverge from those in the literature, offering novel contributions to understanding emotional shifts over time.

Compared to findings in earlier disaster-related studies, Demirel, Çakıcı, and Bulur (2025) [10] report a distinctly negative emotional tone in social media discourse surrounding the 2023 Kahramanmaraş earthquakes. Their NRC-based emotion analysis shows fear and anger as the most prevalent emotions (around 18% and 14.6–14.7%, respectively), with joy relatively low (8.4–8.5%). Sentiment analysis reveals similarly skewed patterns: positive tweets were under 8%, while negative tweets accounted for 39.5% (Syria) and 34.5% (Turkey). These patterns are shaped by contextual factors—widespread destruction, Syria's fragile infrastructure, and politicized narratives (e.g., #WhiteHelmets, #LiftSanctions). Temporal trends (Fig 2) show sentiment gradually improving in Turkish tweets, while remaining persistently negative in Syrian ones, highlighting how emotional responses are influenced by governance, aid, and crisis histories.

In contrast with studies like Zhai et al. [8] and Contreras et al. [12], which emphasized sentiment polarity and situational awareness, our analysis delves deeper into multi-dimensional emotions using NRCLex, revealing that while sentiment polarity remains mostly negative across experiments (approx. 47.4% negative vs. 37.0% positive via Vader), granular emotion trends show that **trust, anticipation, and joy remain consistently marginal** throughout all time intervals. These findings suggest that, unlike events with more immediate recovery or resilience framing, the Turkey-Syria earthquakes were experienced by the public as protracted and deeply traumatic, limiting expressions of hope or positivity.

The term frequencies (e.g., "Turkey," "Syria," "earthquake," "Hatay," and "deprem") affirm the regional and linguistic focus of public discourse, consistent with prior research [18], but our study uniquely highlights **multilingual sentiment dynamics**, reflecting both localized distress and broader international concern. The presence of Turkish-language tokens and culturally specific references may also impact lexicon-based sentiment scores, potentially underrepresenting positive affect due to differences in sentiment lexicon coverage across languages.

Lastly, unlike SOTA systems such as DSSA-H [5] or CASPER [7] which emphasize real-time disaster response or aid classification, our focus on **temporal emotional evolution** fills a critical gap. By tracking how emotions shift from early fear and confusion to later expressions of anger, blame, or solidarity, our study reveals patterns that static models overlook. This reinforces the need for disaster analytics tools that incorporate longitudinal

emotional analysis, especially in complex geopolitical contexts.

The observed transition from acute distress (fear/anger) toward emergent trust aligns with *affective forecasting theory* (Wilson & Gilbert, 2003), where initial emotional intensity attenuates as individuals adapt to crisis realities. Concurrently, *media priming* (Scheufele & Tewksbury, 2007) explains the marginal rise in 'trust' and 'joy': as coverage shifted from devastation to resilience narratives (e.g., successful rescues, aid convoys), it primed collective hopefulness despite persistent negative sentiment.

In psychological terms, the consistent predominance of negative emotions such as fear, sadness, and anger aligns with established trauma and crisis response models, which suggest that public emotional responses to disasters often begin with shock and distress before gradually moving toward coping and recovery stages. The observed increase in the "trust" emotion category over time may reflect a shift from initial chaos to a collective reliance on institutions, humanitarian aid, or community solidarity, as is commonly noted in disaster sociology literature. This trend indicates a gradual emotional transition from acute distress toward cautious optimism or communal dependence, even as the dominant tone remains negative. The marginal increase in 'joy' (from 3.86% in Experiment 01 to 4.11% in Experiment 08) aligns temporally with key developments in disaster response and media cycles. In the initial 72 hours (Feb 6–8), discourse was dominated by shock and desperation, reflected in peak 'fear' (15.58%) and 'anger' (13.47%) in early batches (5,000–20,000 tweets). However, as successful rescues gained media traction (e.g., the extraction of survivors in Hatay on Feb 8–10) and international aid intensified (e.g., UN convoys entering NW Syria by Feb 9), later batches (30,000–40,000 tweets; Feb 12–20) showed a subtle uplift in positive emotions. This suggests that collective hope emerged alongside grief as resilience narratives permeated public discourse—though never eclipsing the overarching negative sentiment. These findings are consistent with the concept of "collective resilience," where despite ongoing hardship, individuals begin to place emotional reliance on perceived sources of support. Understanding these psychological shifts provides a valuable foundation for designing mental health interventions and public communication strategies during prolonged crises.

In summary, our results both validate and extend previous research. They underscore how emotion-focused social media analysis can inform crisis communication, mental health support, and humanitarian outreach by revealing not just what people feel but how those feelings evolve in the wake of catastrophe.

7 Conclusions

This study demonstrates the potential of real-time sentiment and emotion analysis on Twitter to gain meaningful insights into public emotional responses during natural disasters, with a focus on the Turkey-Syria earthquakes. By leveraging advanced NLP tools such as Transformers, Vader, and NRCLex, we uncovered a rich emotional landscape expressed by affected individuals. Beyond sentiment classification, our analysis captured evolving emotional patterns and distinct linguistic features that reflect both psychological distress and region-specific communication styles.

The persistent prevalence of negative sentiment—including high levels of fear, anger, and sadness—across all experiments emphasizes the urgent need for psychosocial support frameworks that are emotionally responsive and culturally informed. The identification of dominant emotion classes and commonly used keywords—such as "deprem," "Hatay," and "rubble"—highlights regionally grounded linguistic traits that can guide the localization of communication and intervention efforts. These insights are crucial for crafting public messaging, mental health outreach, and aid coordination that resonates with affected populations.

Importantly, our findings offer practical value for policy formulation and disaster response planning. By understanding the emotional trajectory of public discourse, authorities can better time and tailor support mechanisms—ranging from mental health counselling and social media engagement to targeted relief distribution. The integration of emotion-specific lexicons and multilingual sentiment recognition enables more nuanced monitoring of distress signals and enhances the cultural and emotional accuracy of disaster management policies.

This research thus provides a replicable framework for crisis-oriented social media analysis, with broad applicability to future emergencies. We recommend future work to expand upon this by incorporating multimodal data (e.g., images, videos), deeper regional language processing, and long-term tracking of emotional recovery. Such advancements will enhance the development of emotionally intelligent disaster response systems and contribute to more resilient, people-centered public health strategies.

.

Declarations

Funding

For this research, no specific grant from a public, corporate, or nonprofit entity was provided.

Conflict of interest

According to the writers, there are no conflicting interests when it comes to publishing this work.

Ethics approval

This study analyzed publicly available Twitter data collected through Twitter's official API under its academic research terms. In accordance with international social media research standards (AoIR Ethics Guidelines v3.0), institutional review board (IRB) approval was not required, as the data consisted of anonymized, aggregate public communications with no personally identifiable information, and there was no interaction with users. The research complied with Twitter's Terms of Service (2023) and the EU's GDPR Article 85, which permits the use of publicly available data for academic purposes. Individual consent was not necessary, as the tweets were publicly posted, analyzed in aggregate, and contained no sensitive or user-level details such as identities or locations. To ensure ethical integrity, the study focused exclusively on earthquake-related content using keyword filtering, applied anonymization during preprocessing (e.g., excluding tweet IDs), and avoided quoting content to protect potentially vulnerable users.

Agreement to be published.

The consent of the participants was obtained before any data or results were released.

Accessibility of information and resources

The researcher in question may seek access to the data and resources used to support the findings of the research.

The Code accessibility

You may ask for the source code used in this work from the corresponding author.

Authors' contributions

The study was designed by Md. Murad Hossain, who also wrote the report. Muhammad Saad Amin contributed to the process of gathering data, analyzing it, and revising the text. Fatema Khairunnasa took part in the review of the literature. Dr. Syed Tahir Hussain Rizvi oversaw the research procedure, offered crucial insights, and checked the work for intellectual integrity

References

[1] Almassar, R.R., Girsang, A.S.: "Detection of traffic congestion based on twitter using convolutional neural network model". IAES International Journal of Artificial Intelligence 11(4), 1448 (2022).

[2] Ilyas, B., & Sharifi, A. (2025). A systematic review of social media-based sentiment analysis in disaster risk management. *International Journal of Disaster Risk Reduction, 123*, 105487. https://doi.org/10.1016/j.ijdrr.2025.105487

[3] Ptaszynski, M., Masui, F., Fukushima, Y., Oikawa, Y., Hayakawa, H., Miyamori, Y., Takahashi, K., Kawajiri, S.: "Deep learning for information triage on twitter" Applied Sciences 11(14), 6340 (2021).

[4] Ruz, G.A., Henr´ıquez, P.A., Mascare˜no, "A.: Sentiment analysis of twitter data during critical events through bayesian networks classifiers." Future Generation Computer Systems 106, 92–104 (2020).

[5] Yao, F., Wang, Y.: "Domain-specific sentiment analysis for tweets during hurricanes (dssa-h): a domain-adversarial neural-network-based approach." Computers, Environment and Urban Systems 83, 101522 (2020).

[6] Ragini, J.R., Anand, P.R., Bhaskar, V.: "Big data analytics for disaster response and recovery through sentiment analysis". International Journal of Information Management 42, 13–24 (2018).

[7] Peri˜n´an-Pascual, C., Arcas-T´unez, F.: "Detecting environmentally related problems on twitter". Biosystems Engineering 177, 31–48 (2019).

[8] Zhai, W., Peng, Z.-R., Yuan, F.: "Examine the effects of neighborhood equity ondisaster situational awareness: Harness machine learning and geotagged twitter data". International Journal of Disaster Risk Reduction 48, 101611 (2020).

[9] Yuan, F., Li, M., Liu, R.: "Understanding the evolutions of public responses using social media: Hurricane matthew case study". International journal of disaster risk reduction 51, 101798 (2020).

[10] Demirel, S., Çakıcı, Z., & Bulur, N. (2025). Exploring reflections of the 2023 Kahramanmaraş earthquake on X: A computational study on Türkiye and Syria. *Akademik İncelemeler Dergisi, 20*(1), 143–171. https://doi.org/10.17550/akademikincelemeler.1593038.

[11] Karami, A., Shah, V., Vaezi, R., Bansal, A.: "Twitter speaks: A case of national disaster situational awareness." Journal of Information Science 46(3), 313–324 (2020)

[12] Contreras, D., Wilkinson, S., Balan, N., James, P.: "Assessing post-disaster recov ery using sentiment

analysis: The case of l'aquila, Italy". Earthquake Spectra 38(1), 81–108 (2022).

[13] Sufi, F.K., Khalil, I.: "Automated disaster monitoring from social media posts using ai-based location intelligence and sentiment analysis". IEEE Transactions on Computational Social Systems (2022).

[14] Kaur, H., Ahsaan, S.U., Alankar, B., Chang, V.:" A proposed sentiment analysis deep learning algorithm for analyzing covid-19 tweets". Information Systems Frontiers, 1–13 (2021).

[15] Ruan, T., Kong, Q., McBride, S.K., Sethjiwala, A., Lv, Q.: "Cross-platform analy-sis of public responses to the 2019 ridgecrest earthquake sequence on twitter and reddit". Scientific reports 12(1), 1634 (2022).

[16] Parimala, M., Swarna Priya, R., Praveen Kumar Reddy, M., Lal Chowdhary, C., Kumar Poluru, R., Khan, S.:" Spatiotemporal-based sentiment analysis on tweets for risk assessment of event using deep learning approach. Software": Practice and Experience 51(3), 550–570 (2021).

[17] Zheng, Z., Shi, H.-Z., Zhou, Y.-C., Lu, X.-Z., Lin, J.-R.: "Earthquake impact analysis based on text mining and social media analytics". arXiv preprint arXiv:2212.06765 (2022).

[18] Toraman, C., Kucukkaya, I.E., Ozcelik, O., Sahin, U.: "Tweets under the rubble: Detection of messages calling for help in earthquake disaster". arXiv preprint arXiv:2302.13403 (2023).

[19] He, W., Yuan, Q., Li, N.: "Research on the characteristics of internet public opinion and public sentiment after the sichuan earthquake based on the perspective of Weibo". Applied Sciences 13(3), 1335 (2023).

[20] Maceda, L.L., Llovido, J.L., Palaoag, T.D.: "Corpus analysis of earthquake related tweets through topic modelling". International Journal of Machine Learning and Computing 7(6), 194–197 (2017).

[21] Contreras, D., Wilkinson, S., Aktas, Y.D., Fallou, L., Bossu, R., Land`es, M.: "Intensity-based sentiment and topic analysis. the case of the 2020 aegean earthquake". Frontiers in Built Environment 8 (2022).

[22] Vo, B.-K.H., Collier, N.: "Twitter emotion analysis in earthquake situations". Int. J. Comput. Linguistics Appl. 4(1), 159–173 (2013).

[23] Bai, H., Yu, G.: "A weibo-based approach to disaster informatics: incidents monitor in post-disaster situation via weibo text negative sentiment analysis". Natural Hazards 83(2), 1177–1196 (2016).

[24] Wang, Y., Taylor, J.E.: "Coupling sentiment and human mobility in natural disasters: a twitter-based study of the 2014 south napa earthquake". Natural hazards 92, 907–925 (2018).

[25] Mızrak, S., Ozdemir, A., Aslan, R.: "Adaptation of hurricane risk perception scale to earthquake risk perception and determining the factors affecting women's earthquake risk perception". Natural Hazards 109(3), 2241–2259 (2021)

[26] Contreras, D., Wilkinson, S., Alterman, E., Herv´as, J.: "Accuracy of a pre-trained sentiment analysis (sa) classification model on tweets related to emergency response and early recovery assessment: the case of 2019 albanian earthquake." Natural Hazards 113(1), 403–421 (2022).

[27] Cabezas, J., Moctezuma, D., Fern´andez-Isabel, A., Diego, I.: "Detecting emo-tional evolution on twitter during the covid-19 pandemic using text analysis". International Journal of Environmental Research and Public Health 18(13), 6981 (2021).

[28] Kydros, D., Argyropoulou, M., Vrana, V.: "A content and sentiment analysis of greek tweets during the pandemic". Sustainability 13(11), 6150 (2021).

[29] Le´on-Sandoval, E., Zareei, M., Barbosa-Santill´an, L.I., Falc´on Morales, L.E., Pareja Lora, A., Ochoa Ruiz, G., et al.: "Monitoring the emotional response to the covid-19 pandemic using sentiment analysis: a case study in Mexico". Computational Intelligence and Neuroscience 2022 (2022).

[30] Jabalameli, S., Xu, Y., Shetty, S.:" Spatial and sentiment analysis of public opinion toward covid-19 pandemic using twitter data: At the early stage of vaccination". International Journal of Disaster Risk Reduction 80, 103204 (2022).

[31] Kausar, M.A., Soosaimanickam, A., Nasar, M.: "Public sentiment analysis on twitter data during covid-19 outbreak". International Journal of Advanced Computer Science and Applications 12(2) (2021).

[32] Storey, V.C., O'Leary, D.E.: "Text analysis of evolving emotions and sentiments in covid-19 twitter communication". Cognitive Computation, 1–24 (2022).

[33] Tobita, T., Kiyota, T., Torisu, S., Cinicioglu, O., Tonuk, G., Milev, N., ... & Shiga, M. (2024). Geotechnical damage survey report on February 6, 2023 Turkey-Syria Earthquake, Turkey. Soils and Foundations, 64(3), 101463.

[34] Ni, S., Sun, H., Somerville, P., Yuen, D. A., Milliner, C., Wang, H., ... & Cui, Y. (2023). Complexities of the Turkey-Syria doublet earthquake sequence. The Innovation, 4(3).

[35] Rashid, A. K., & Fındık, O. (2024). Sentiment Analysis of Twitter: Turkey Earthquake 2023 Case.

[36] Hidayat, T., & Hassan, R. (2024). Unveiling Twitter's Take: Researching# TurkeyQuake Response & Rehabilitation. Educational Administration: Theory and Practice, 30(3), 2361-2371.

[37] Mandal, S., Alugubelly, M., & Jayabalan, M. (2023, December). Predicting Themes of Tweets on Earthquakes in Turkey & Syria for Real-Time Classification. In 2023 16th International

Conference on Developments in eSystems Engineering (DeSE) (pp. 311-316). IEEE.

[38] Alsheyab, M., Filimon, N., & Fusté-Forné, F. (2024). Digital Content Analysis of Corporate Social Response to Turkey and Syria Earthquakes. The Arab World Geographer, 27(1), 47-62.

[39] Cinar, E. N., Abbara, A., & Yilmaz, E. (2023). Earthquakes in Turkey and Syriacollaboration is needed to mitigate longer terms risks to health. BMJ, 38.

[40] Utku, A., & Can, U. (2023). Machine learning based a comparative analysis for detecting tweets of earthquake victims asking for help in the 2023 Turkey-Syria earthquake. *Journal of Soft Computing and Artificial Intelligence*, *4*(2), 55–62. https://doi.org/10.55195/jscai.1365639.

[41] Rashid, A. K., & Fındık, O. (2024). Sentiment Analysis of Twitter: Turkey Earthquake 2023 Case.

[42] Erokhin, D., & Komendantova, N. (2023). Earthquake conspiracy discussion on Twitter. *Journal of Crisis Communication*, 12(2), 134-146. https://doi.org/10.1016/j.jcc.2023.01.005

[43] Aldamen, Y., & Hacimic, E. (2023). Positive determinism of Twitter usage development in crisis communication: Rescue and relief efforts after the 6 February 2023 earthquake in Türkiye as a case study. *Social Sciences*, *12*(8), 436.

[44] Alharm, A. O., & Naim, S. (2023). Enhancing natural disaster response: A deep learning approach to disaster sentiment analysis using BERT and LSTM. *The University of Strathclyde, Computer and Information Science Department*.

Key Information Recognition in Speech Using Spectral Subtraction and Wavelet Thresholding Methods: An Optimized Approach with the S-PaddleSpeech Model

Yang Yang
Department of Information Management, Henan Vocational College of Light Industry, Zhengzhou 450008, China
E-mail: yang_yangyy@outlook.com

Keywords: speech recognition, paddlespeech model, keyword detection, spectral subtraction, wavelet threshold denoising method

Received: November 14, 2024

A PaddleSpeech model was developed to address the issue of low accuracy in speech key information detection technology. Spectral subtraction is used in the process to enhance the quality of speech signals and improve signal-to-noise ratio by reducing noise interference. The method of signal processing through wavelet transforms balances the selection of appropriate denoising methods in both time and frequency domains. Frame segmentation, windowing, and Fourier transform techniques were used in the data processing stage. The experiment outcomes show that for specific and non-specific speech, the CNN detection algorithm achieves a keyword recognition accuracy of 0.9 when the sample size is less than 20, while the FNN algorithm achieves an accuracy of 0.8 when the sample size reaches 60. Both in terms of sample size requirements and keyword recognition accuracy, CNN outperforms FNN. In addition, in the application testing of the model, the improved PaddleSpeech model shows significantly better recognition performance for 20 keywords in audio than the original PaddleSpeech model, with a recognition accuracy of up to 90% ($P<0.05$). In the audio character recognition verification of the improved PaddleSpeech model and SpeechRecognition model, the former correctly recognizes 15019 characters with an accuracy of 98.9580%, while the latter correctly recognizes 14593 characters with an accuracy of 96.1520. The former has an accuracy 2.806% higher than the latter ($P<0.05$). Therefore, the improved PaddleSpeech model proposed by the research has good speech keyword recognition ability and effectively improves recognition accuracy.

Povzetek: Odprtokodni orodni komplet za obdelavo govora in zvoka PaddleSpeecch je izboljšan z metodo spektralne subtrakcije in CNN algoritmom, kar doseže višjo robustnost v hrupu ter kvaliteti prepoznavanja.

1 Introduction

Due to the swift advancement of mobile Internet, computer technology has found extensive application across numerous societal sectors, with which a large amount of data of various types is produced rapidly [1]. Among them, due to the fact that voice is the most direct and effective method for people to engage in social activities, modern intelligent devices are constantly increasing their voice interaction functions, so voice data has received more and more attention from people [2]. To process various speech data, various speech recognition technologies are constantly being updated and upgraded. Among them, in the process of voice interaction between smart devices and humans, the key information recognition technology of speech performs well in various speech recognition scenarios due to its high computing power and fast recognition speed [3, 4]. However, there is still an issue of inadequate identification precision in current speech recognition systems when autonomously recognizing speech content [5]. On the one hand, when recognizing a large number of words simultaneously, speech recognition systems may experience spelling errors and omissions, thereby reducing accuracy. Studies indicate that, in environments where over 150 words are processed at once, error rates may spike by up to 35%, leading to a drop in recognition accuracy to below 65%. This is particularly problematic in real-time applications where precision is crucial [6]. On the other hand, the system's performance can suffer from significant keyword omission due to the extensive vocabulary and prolonged speech recognition time. Research has shown that in scenarios involving lengthy dialogues — typically exceeding 300 words — the occurrence of keyword omissions can reach as high as 25%. This not only impacts the system’s ability to capture essential information but also reduces the effectiveness of extracting key information in critical applications, such as automated transcription or real-time translation [7, 8]. Therefore, the PaddleSpeech model is constructed and improved using speech enhancement algorithms and keyword detection algorithms to raise the detection capability of speech and the precision of key content recognition.

The study is organized into four parts. The first part presents an overview and analysis of the domestic and international research on speech key information recognition technology. The second part is to construct the PaddleSpeech model and optimize it. The third part is to testify the general capability of the algorithm and the improved PaddleSpeech model. The last part is an overview of the whole piece of writing.

2 Related work

With the progress of computers and networks, an increasing number of artificial intelligence (AI) techniques are being utilized to various aspects of life. Voice interaction system is one of them. With the support of speech recognition technology, people can communicate with each other and machines through smart devices. Faced with the increasing demand for voice interaction, how to improve the accuracy of voice detection has become a growing concern for researchers. Yuan Q et al. explored English speech translation recognition technology based on long short-term memory network algorithm for speech information recognition problem. The study analyzed the shortcomings of traditional template matching and statistical pattern recognition, and proposed the application of long short-term memory networks to achieve intelligent information processing by simulating the information processing of the human brain. The results showed that the speech recognition accuracy of the long short-term memory network was as high as 94%, significantly improving information storage efficiency and increasing the processing speed of speech data to a maximum of 4.5 seconds, effectively meeting the society's demand for intelligence [9]. Xu et al. proposed a novel bimodal emotion recognition algorithm for the problem of emotion recognition in speech information recognition. This algorithm combines parallel convolution modules and attention based bidirectional long short-term memory modules to achieve feature abstraction and fusion. The experimental results show that the recognition accuracy of audio data reaches 74.70%, text is 77.13%, and the accuracy of the bimodal fusion model is as high as 90.02%. This study demonstrates the feasibility of processing heterogeneous information within homogeneous network components, providing flexibility for modal extension and architecture design [10]. For the problem of automatic speech recognition, Zhang et al. proposed a method called BigSSL, which used a giant Conformer model pre-trained with wav2vec 2.0 and combined SpecAugment technology for noisy student training. This method reached a word error rate of 1.4%/2.6% on the LibriSpeech test set and other test sets, which was a breakthrough in comparison with the cutting-edge WER of 1.7%/3.3% at that time [11]. To handle the matter of low accuracy in speech emotion detection, Zhu et al. first proposed two baseline methods, namely Vanilla Fine-tuning (V-FT) and Task Adaptive Pre-training (TAPT), to evaluate the capability of Wav2Vec 2.0 in SER. The experiment outcomes showed that on the IEMOCAP dataset, V-FT could surpass existing cutting-edge models, demonstrating the potential of Wav2Vec 2.0 in SER tasks [12]. Ambrogio et al. raised a simulated AI chip to handle the matter of low energy efficiency of AI models in high-precision tasks. Researchers have introduced a new type of simulated AI chip that integrates 35 million phase change memory devices on 34 tiles, large-scale parallel inter tile communication, and simulated low-power peripheral circuits, achieving chip sustained performance of 12.4 megaoperations per watt per second [13].

In addition, Burchi M and Timofte R proposed an efficient Conformer model for audio visual fusion to address the performance degradation of automatic voice recognition technologies in loud settings. Recently, although end-to-end ASR systems grounded on neural networks have shown excellent performance on clean audio samples, their performance often deteriorates under noisy conditions. To resolve this problem, scholars developed a method that combined audio and visual modalities to enhance the noise robustness of an efficient conformal connected temporal classification architecture. They relaxed the conditional independence assumption based on the CTC model by introducing residual modules between CTCs, and replaced efficient consistency grouping attention with a more efficient and simpler patch attention mechanism [14]. Deng et al. proposed a position guidance process monitoring and vibration detection method based on smartphones using PaddleSpeech for speech information recognition problems. In the study, smartphones were used to record the human-machine interface and tool movements of machine tools, and the correlation between tool cutting positions and sound signals was obtained through optical character recognition technology. Using PaddleSpeech's open-source model for speech recognition and voice separation, removing periodic components, and using the ratio of residual signal energy to total signal energy for vibration detection. Finally, this method was validated through robot milling and deep hole drilling experiments [15]. In response to the issue of long-term companionship between humans and pets, Chen Jiancheng et al. used Autoencoder and PaddleSpeech models, combined with VR systems and deepfake technology, to restore the appearance of family members and alleviate users' longing for their loved ones [16]. Wu et al. used PaddleSpeech to construct a corpus for speech information recognition, which is the first natural audio-visual multimodal database for Chinese social interaction agents. It includes 48 hours of videos and annotations, covering eight modalities. The study analyzed the characteristics of voice, language, behavior, and multimodal combinations during the questioning process, and tested the performance of six baseline models on three tasks. The results of this project provide reference for designing social interaction agents that are more in line with Chinese culture and user needs, and promote the improvement of Chinese daily data processing [17]. As a reaction to the problem that current depression detection models were difficult to represent small changes in depression, Zhou et al. raised a multi-granularity fusion network, which integrated

different feature information to promote multi-level information and multi-resolution interaction, thereby optimizing the effectiveness of depression detection [18]. The summary and analysis of related work are shown in Table 1.

Table 1: Summary and analysis of related work

Reference	Advantages	Disadvantages	Performance Metrics	Advantages of This Study's Method
[7] Yuan et al.	High accuracy in English speech recognition	May not be suitable for non-English speech	Recognition accuracy: 94%	Better for Chinese speech recognition
[8] Xu et al.	Accurate in dual-modal emotion recognition	Requires textual information	Audio recognition accuracy: 74.70%	More suitable for pure speech recognition
[9] Zhang et al.	Good performance in large model pre-training	High computational resource requirements	Word error rate: 1.4%/2.6%	High accuracy with small samples
[10] Zhu et al.	High potential with Wav2Vec 2.0 and large data	Requires domain-specific data	Error rate: Lower than existing models	Good performance across multiple fields
[11] Ambrogio et al.	High efficiency of AI chip	Technical challenges in actual deployment	Energy efficiency: 12.4 megaoperations per watt per second	Easier to promote with software optimization
[12] Burchi and Timofte R	Improved robustness in noisy environments	Requires visual information	Robustness improvement: Significant	Applicable without visual information
[13] Deng et al.	Smartphone monitoring and vibration detection	Dependent on specific hardware	Recognition accuracy: High	Not dependent on specific hardware
[14] Chen Jiancheng et al.	Relieves longing by restoring appearance	High real-time requirements	User satisfaction: High	Easy to implement and deploy
[15] Wu et al.	First Chinese multimodal database	Difficult to build and maintain the database	Database size: 48 hours of video and annotations	No need for large-scale database
[16] Zhou et al.	Optimized depression detection with multi-granularity fusion network	Limited effectiveness outside of depression detection	Detection accuracy: Optimized	Wide applicability across fields

PaddleSpeech not only has excellent computing performance and speed, but also has good flexibility and scalability, making it easy to adapt to various application scenarios. Meanwhile, PaddleSpeech adopts the latest deep learning algorithm optimization when processing speech data, which can maintain high recognition accuracy in noisy environments. These advantages make PaddleSpeech the preferred model for achieving efficient key data recognition in this study. By comparing with BigSSL or Wav2Vec, although these models also have their own characteristics, PaddleSpeech shows higher adaptability and effectiveness in terms of application requirements and performance metrics specific to this article. Based on studies conducted by both domestic and international researchers, there is a problem of low detection accuracy in current speech key information recognition technology. Therefore, this research is being conducted to optimize the PaddleSpeech model using speech enhancement algorithms and keyword detection algorithms to improve the recognition performance of speech key information.

3 Construction of speech recognition system and S-PaddleSpeech model

3.1 Construction of PaddleSpeech speech recognition system

The precision of speech key information recognition technology depends on whether the system recognizes speech accurately, and the speech recognition system mainly includes two parts: speech signal (SS) pretreatment and speech recognition. Among them, SS pretreatment is the process of extracting the essence of speech, which includes time-domain waveforms that can reflect many features, but other functional tools are still needed to assist in obtaining more comprehensive information. At present, due to the close fit between Mel-Frequency Cepstral Coefficients (MFCC) and human ear nonlinear frequency induction, MFCC is the most commonly used coefficient in keyword detection technology [19]. When the target speech reaches the frequency band above 800Hz, the speech will attenuate due to the high frequency band, which will result in the energy of the high-frequency SS being less than that of the low-frequency part.

To balance the high and low frequency bands of the SS, it is necessary to emphasize the high-frequency part. The expression of the high pass digital filter for emphasizing the frequency band is shown in equation (1).

$$H(z)=1-az^{-1} \tag{1}$$

In equation (1), a represents the pre-emphasis coefficient, and 0.9<a<1.0. To achieve fast Fourier transform, the signal input must be stable. However, due to the time-varying and short-term stationarity of SSs, in order to obtain shorter speech frame segments, it is necessary to perform frame segmentation on the speech. When processing frames, there should be overlap between two consecutive frame segments, that is, frame shift, and the signal segment range should be between 10-30ms. After frame processing, in order to avoid spectral leakage caused by uneven edges of speech frame segments, each speech frame needs to be windowed. Windowing is the process of performing operations on speech frames, resulting in the final frame signal being weakened to 0. Taking the Hamming window as an example, the calculation process of multiplying the window function by the speech frame segment is shown in equation (2).

$$w(n)=0.54-0.46\cos[\frac{2\pi n}{(N-1)}],0\le n\le N-1 \tag{2}$$

In equation (2), $w(n)$ represents the window function value of the Hamming window; N represents the window length; n represents the sample index within the window. After windowing, in order to achieve fast Fourier transform, it is necessary to calculate the discrete Fourier transform through a computer. The discrete Fourier transform converts time-based data into data represented in the frequency domain, with sampling points corresponding to complex numbers. The speech frequency-domain signal is the imaginary and real parts of the complex numbers. The specific calculation process of discrete Fourier is shown in equation (3).

$$X(k)=\sum_{N=0}^{N-1}x(n)e^{-j\frac{2\pi nk}{N}},k=0,1,...,N-1 \tag{3}$$

In equation (3), the discrete Fourier transform divides the signal into two parts, with the signal length divided from N to N/2. Continuing this process can yield the final result. After fast Fourier transform, the speech is converted from actual frequency to Mel frequency, and then Mel filter bank is used. The correspondence between *Mel* frequency f_{mel} and the original frequency f is shown in equation (4).

$$f_{mei}(f)=2595\lg(1+\frac{f}{700Hz}) \tag{4}$$

In equation (4), the Mel filter bank primarily consists of M triangular filters, as represented in Figure 1.

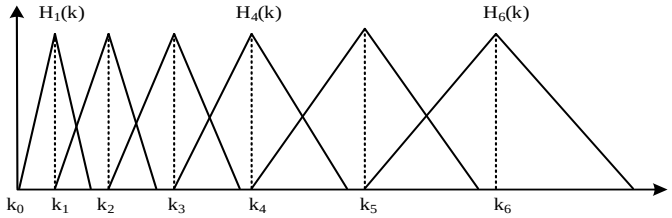


Figure 1: Mel filter bank

In Figure 1, the triangular filters H1 (k), H4 (k), H6 (k), etc. constitute the Mel filter group, which has the characteristics of high frequency with low density and low frequency with high density. This is also in line with the low resolution of the human ear at high frequencies and high resolution at low frequencies. The calculation process of the filter center frequency $f(m)$ is shown in equation (5).

$$f(m)=\frac{N}{f_s}f_{mel}{}^{-1}\left(f_{mel}(f_l)+m\frac{f_{mel}(f_h)-f_{mel}(f_l)}{M+1}\right),0\le m\le M \tag{5}$$

In equation (5), $f(m)$ represents the frequency dependent function after processing; f_s represents the sampling frequency; f_h represents the highest frequency, and f_l represents the lowest frequency; M represents the length of a processing area or the number of sample points. The calculation method for the coefficient $H_m(k)$ of the filter is shown in equation (5).

$$H_m(k)=\begin{cases}0,k<f(m-1)\\ \dfrac{k-f(m-1)}{f(m)-f(m-1)},f(m-1)\le k\le f(m)\\ \dfrac{f(m+1)-k}{f(m+1)-f(m)},f(m)\le k\le f(m+1)\\ 0,k>f(m+1)\end{cases} \tag{6}$$

In equation (6), $k=0,1,...,N-1$. After passing through the Mel filter bank, the calculation process of spectral energy $S_i(m)$ is represented in equation (7).

$$S_i(m)=\sum_{k=0}^{N-1}[X_i(k)]^2H_m(k),0\le m\le M \tag{7}$$

After outputting the Mel power spectrum, the calculation of the MFCC is shown in equation (8).

$$mfcc(i,n)=\sqrt{\frac{2}{M}}\sum_{m=0}^{M-1}\log[S(i,m)\cos\left[\frac{\pi n(2m-1)}{2M}\right] \tag{8}$$

Early speech recognition technology was limited, and its recognition accuracy needed to be improved. However, modern speech recognition systems have made significant progress compared to before, not only increasing the number of speakers recognized, but also greatly enriching the vocabulary [20]. The PaddleSpeech model of Baidu PaddlePaddle series is used for voice identification, and its basic structure is represented in Figure 2.

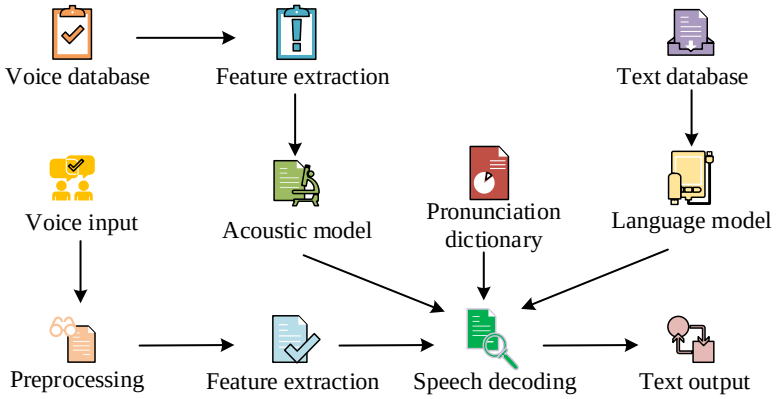


Figure 2: Basic structure of PaddleSpeech model

In the model structure shown in Figure 2, when the speaker speaks, the speech can be converted into electrical signals through a microphone, then converted into data through an analog-to-digital converter, and finally converted from audio to text through the corresponding model.

2.2 Improvement of PaddleSpeech model based on spectral subtraction and wavelet threshold denoising method

In daily environments, SSs are inevitably mixed with noise during input, and the basic PaddleSpeech model is affected by it during speech recognition, thereby reducing the precision of voice identification. Therefore, speech enhancement algorithms improve model precision. Denoising plays a crucial role in keyword recognition, significantly improving the accuracy and efficiency of recognition. Background noise can interfere with the clarity of signals, making it difficult for the system to accurately extract key information when processing audio or video inputs. By denoising, these interferences can be eliminated or reduced, improving the signal-to-noise ratio of the signal and making the keywords more prominent, facilitating the subsequent recognition process.

Speech enhancement algorithms include spectral subtraction and wavelet thresholding denoising. In traditional spectral subtraction, the noisy signal is represented in equation (9).

$$y(t) = x(t) + d(t) \tag{9}$$

In equation (9), $y(t)$ is a noisy signal, $x(t)$ is a pure signal, and $d(t)$ is additive noise. By performing Fourier transform on the three, equation (10) can be obtained.

$$y(\omega) = x(\omega) + d(\omega) \tag{10}$$

By squaring equation (9), the energy distribution of the noisy signal across frequency ranges is shown in equation (11).

$$|Y(\omega)|^2 = |X(\omega)|^2 + |D(\omega)|^2 + X^*(\omega)D(\omega) + X(\omega)D^*(\omega) \tag{11}$$

In equation (11), $X^*(\omega)D(\omega) + X(\omega)D^*(\omega)$ represents the components produced by the correlation between $X(\omega)$ and $D(\omega)$. Since the prerequisite for spectral subtraction is that $x(t)$ and $d(t)$ are unrelated to each other, equation (12) can be obtained.

$$|Y(\omega)|^2 = |X(\omega)|^2 + |D(\omega)|^2 \tag{12}$$

Assuming that the noise power of the quiet segment and the speech segment are fixed values, and the estimated noise power of the quiet segment is $|D(\omega)|^2$, the pure speech power spectrum is shown in equation (13).

$$|X(\omega)|^2 = |Y(\omega)|^2 + |D(\omega)|^2 \tag{13}$$

In equation (13), $|X(\omega)|^2 \geq 0$. If it is less than 0, there is a bias in the noise estimation, and the adjustment mechanism is equation (14).

$$\begin{cases} |X(\omega)|^2 = |Y(\omega)|^2 - |D(\omega)|^2, |Y(\omega)|^2 > |D(\omega)|^2 \\ |X(\omega)|^2 = 0, |Y(\omega)|^2 \leq |D(\omega)|^2 \end{cases} \tag{14}$$

After speech enhancement, the approximated frequency distribution of the pure SS is shown in equation (15).

$$\hat{X}(\omega) = |X(\omega)| \exp[j\theta_{Y(\omega)}] \tag{15}$$

In equation (15), $\theta_{Y(\omega)}$ represents the phase of noisy speech, and $\hat{X}(\omega)$ represents the SS without noise. Performing inverse Fourier transform on $\hat{X}(\omega)$ results in an enhanced SS as shown in equation (16).

$$\hat{x}(\omega) = IFFT\left[\hat{X}(\omega)\right] \tag{16}$$

Spectral subtraction has a small computational complexity and is widely used in speech enhancement. However, there is also a problem that SSs may contain musical noise. To reduce its impact, the adjustment mechanism is improved, as shown in equation (17).

$$\begin{cases} |X(\omega)|^2 = |Y(\omega)|^2 - \alpha|D(\omega)|^2, |Y(\omega)|^2 > \alpha|D(\omega)|^2 \\ |X(\omega)|^2 = \beta|D(\omega)|^2, |Y(\omega)|^2 \leq \alpha|D(\omega)|^2 \end{cases} \tag{17}$$

In equation (17), α is the over reduction factor used to control the degree of noise reduction. Increasing α helps improve the signal-to-noise ratio and suppress noise. β is a compensation factor that determines the amount of noise retained in the spectrum. Increasing β will reduce the noise component, but too much may lead

to a rise in ambient noise levels. The different values of α and β affect the denoising effect. In traditional methods, α=1 and β=0. Research has optimized traditional spectral subtraction by adjusting these two parameters, namely α=5 and β=0.002. Through preliminary experiments, it was found that the excessive reduction factor value can effectively suppress noise without significantly distorting the speech signal. This compensation factor value achieves a balance between noise suppression and background sound preservation, avoiding the environmental noise increase caused by excessive suppression. This choice is universal because its moderate residual noise allows for the reduction of background noise interference while preserving important speech features. In practical scenarios, such as phone conversations or meeting minutes, this parameter can balance speech intelligibility with background realism. In the key process of spectrum subtraction adjustment, in addition to optimizing parameters, the characteristics of environmental noise and the spectral features of signals are also crucial. Firstly, experiments need to be conducted under different signal-to-noise ratio conditions to evaluate the adaptability of adjusting parameters to different types of noise. This means that in practical applications, it is necessary to establish a dynamic adjustment mechanism based on specific scenarios, so that parameter settings can be automatically adjusted according to external noise changes in real-time processing.

The theory of wavelet transform can localize the time-frequency of signals and has the characteristic of flexibly selecting wavelet functions. Wavelet threshold denoising aims to enhance valuable components in SSs. In wavelet coefficients, useful signals are usually composed of larger amplitude components [21]. The key to achieving wavelet threshold denoising lies in selecting appropriate wavelet bases, thresholds, and threshold functions. The selected wavelet in the study is symN, which may cause phase distortion when reconstructing SSs. However, symN wavelet can reduce phase distortion to a certain extent. Regarding threshold selection, the threshold chosen for the study is the minimax threshold, which is calculated as shown in equation (18).

$$\lambda\begin{cases}0, N \le 32\\ 0.3936+0.1829\dfrac{\ln N}{\ln 2}, N > 32\end{cases} \tag{18}$$

The threshold functions are generally categorized into two types: hard and soft, which are commonly used to correct wavelet coefficient errors. The hard threshold image is shown in Figure 3.

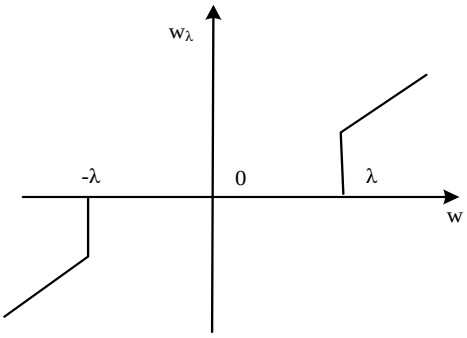


Figure 3: Hard threshold function image

The soft threshold image is shown in Figure 4.

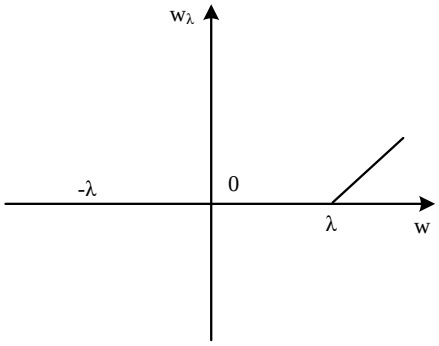


Figure 4: Soft threshold function image

The threshold function expression selected by the research is shown in equation (19).

$$\hat{w}_j=\begin{cases}w_j, |w_j|>\beta\lambda\\ \operatorname{sgn}(w_j)(|w_j|-\lambda)\dfrac{\beta}{\beta-1}, \lambda<|w_j|<\beta\lambda\\ 0, |w_j|\le\lambda\end{cases} \tag{19}$$

In equation (19), $\hat{w}_j$ represents the estimated value of wavelet coefficients, j represents the decomposition scale, w_j represents the original wavelet coefficients, β represents a constant, and $1.5\le\beta\le 3$. λ represents the threshold. Research enhances speech through spectral subtraction and denoises speech through wavelet transform, in order to optimize the initial PaddleSpeech model and improve its speech recognition accuracy, ultimately obtaining the S-PaddleSpeech model. The main differences in technology and framework between PaddleSpeech and S-PaddleSpeech are reflected in the following aspects: Firstly, S-PaddleSpeech adopts a Transformer based model architecture, while PaddleSpeech uses traditional CNN and RNN structures. Secondly, S-PaddleSpeech implements optimization strategies such as mixed precision training and distributed training to improve training efficiency, while PaddleSpeech relies on relatively fixed training methods.

In terms of feature extraction, S-PaddleSpeech emphasizes end-to-end learning, reducing reliance on traditional feature extraction techniques. In addition, S-PaddleSpeech has a higher modular design that allows users to flexibly choose and replace model components, while PaddleSpeech is relatively lacking in this regard. Finally, S-PaddleSpeech supports advanced features such as multi speaker recognition and emotion recognition, while PaddleSpeech mainly focuses on basic speech recognition and synthesis tasks. These core technological differences have built a more flexible and efficient framework for S-PaddleSpeech.

Speech recognition technology can convert speech as a whole into text, but to recognize key information, keyword detection algorithms need to be introduced to accurately extract the required information. Therefore, the research will combine deep learning-based keyword recognition algorithms with the S-PaddleSpeech model to further optimize it. The process of speech keyword detection technology is represented in Figure 5.

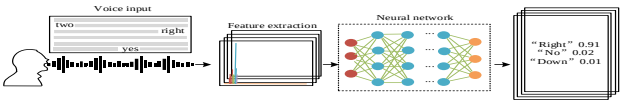


Figure 5: Process of speech keyword detection technology

In Figure 5, the speech keyword detection technology first extracts characteristics from the input SS, and then trains the neural network model. After that, the neural network model extracts vectors from the features and classifies them. Finally, the probability distribution of speech is analyzed, and the keyword with the highest probability is the required result. Due to the strong feature extraction ability and small number of training parameters of CNN, this study uses CNN to optimize the S-PaddleSpeech model. Its structure is shown in Figure 6 [22].

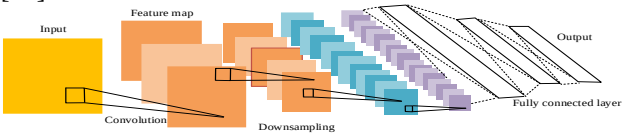


Figure 6: CNN structure diagram

In Figure 6, the hidden layers of CNN models are generally divided into convolutional layers and pooling layers. The convolutional layer is the core part of CNN, which performs dimensionality reduction and feature extraction on the input image through convolution operations (linear operations, i.e., shifting on the original image). The function of the pooling layer is feature extraction, which involves removing certain features. By eliminating non-essential samples from the feature map, the parameter count can be decreased further. After the SS processing is completed, the CNN model extracts feature from the MFCC matrix through convolutional kernels, and then reduces the size of the feature map through pooling layer aggregation operations. Finally, the feature vectors are transformed into one dimension and the final result is output to obtain the C-S-PaddleSpeech model. Before applying the algorithm in practice, there is also a data preprocessing step, which is mainly divided into three parts: frame segmentation and windowing, Fourier transform and Mel filtering, and noise level estimation. Divide the speech signal into frames to ensure short-term stability, with a frame length of 25 milliseconds and a frame shift of 10 milliseconds. Each frame is windowed using Hamming window to reduce spectral leakage in subsequent Fourier analysis. Perform discrete Fourier transform on the windowed signal of each frame to extract frequency domain features.

3 Application testing of C-S-PaddleSpeech model

3.1 Performance testing of speech recognition algorithms and keyword algorithms

To verify the capability of the speech enhancement algorithm, the ROC curve was used for performance testing. The experimental system used Windows 11, the device was a 64-bit operating system, and the programming language was Python. The display processor of the experimental equipment is NVIDIA GeForce RTX 3080, the processor is Intel Core i7-11700K, the memory is 32GB DDR4 3200MHz RAM, and the storage device is 1TB NVMe SSD. The dataset used in the study includes speech samples from different fields and environments, such as indoor conversations, street noise, coffee shop background noise, and industrial area noise. The speech types cover speakers of various genders, ages, and accents, ensuring sample diversity. The dataset consists of 5000 recording samples, ranging in duration from 3 seconds to 30 seconds, with a total duration of approximately 30 hours. The noise level has been carefully labeled and divided into low (5-15 dB SNR), medium (0-5 dB SNR), and high (below 0 dB SNR) noise environments to ensure the presence of diverse noise influences in the dataset. The noise level has been carefully labeled and divided into low (5-15 dB SNR), medium (0-5 dB SNR), and high (below 0 dB SNR) noise environments to ensure the presence of diverse noise influences in the dataset. The use of 5-fold cross validation effectively reduces the risk of model overfitting caused by random partitioning, ensuring the algorithm's generalization ability on unseen data. In order to promote the reproducibility of the experiment, the random seed used in the experiment was 123456. Fixed random seeds ensure that the random number sequence generated during each experiment run is the same, allowing different researchers to replicate the experimental results under the same conditions. The deep learning framework applied in the experiment was TensorFlow. The test results are shown in Figure 7.

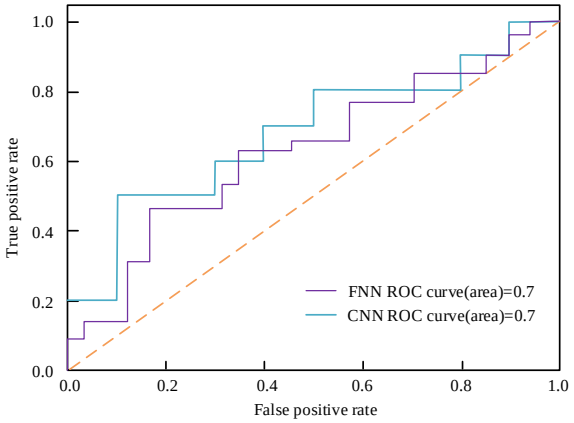


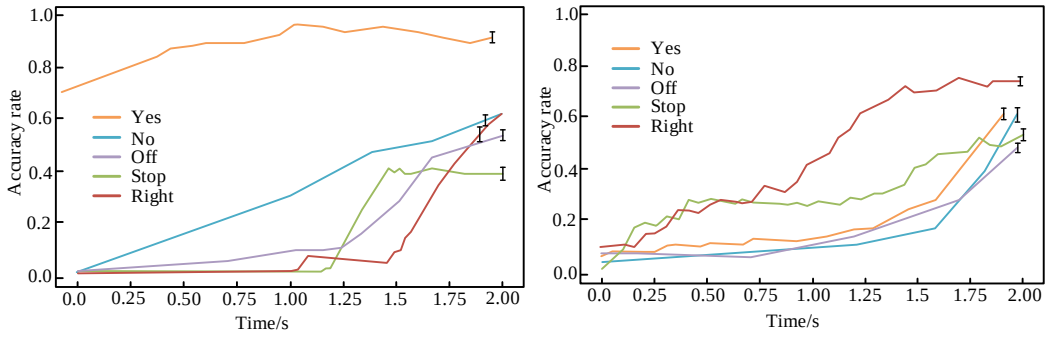
Figure 7: ROC schematic diagram

Figure 7 shows the False Positive Rate (FPR) as the x-axis and the True Positive Rate (TPR) as the y-axis. The closer the ROC curve is to the upper left corner, the better the performance of the algorithm. Observing Figure 7, it can be seen that the ROC curve of CNN shows an overall stepwise upward trend, especially in the early stage where the curve rises rapidly, and the upward speed slows down slightly in the middle and later stages, but still maintains a stable upward trend. In contrast, the FNN curve is closer to the bottom right corner. These ROC curves can clearly depict the classification performance of the model at different thresholds, which is consistent with the theoretical prediction of the relationship between signal-to-noise ratio and model performance, that is, a higher signal-to-noise ratio will significantly reduce the interference of noise on the model, thereby improving recognition rate. Through t-test on the result data, it was found that the p-values of the obtained test data were all below the significance level of 0.05. From this, the speech enhancement algorithm raised by the study could effectively enhance speech and had good noise reduction effect. To confirm the performance of the keyword detection algorithm, the same keywords in long and short sentences were individually evaluated, and the outcomes are represented in Figure 8.

In Figure 8 (a), in the initial stage, the algorithm had a higher detection accuracy for the keyword “yes” in short sentences compared to other keywords. Its accuracy curve fluctuated up and down over time, but the accuracy remained above 0.7. The detection accuracy of keywords “stop” and “right” was relatively low in the first 1 second, and the accuracy curve showed a rapid growth trend after 1 second. This reflects the algorithm's strong ability to detect specific keywords in a short period of time. In Figure 8 (b), the accuracy curves of the keywords "right" and "stop" in long sentences fluctuate up and down. Although its overall accuracy is gradually improving, it is generally lower than the accuracy in short sentences. This trend may be related to the complexity and information content of sentences, as long sentences often contain more information, leading to certain interference in algorithm recognition. The accuracy of keyword detection is above 0.5 for both long and short sentences. Through t-test on the result data, it was found that the p-values of the obtained test data were all below the significance level of 0.05. The CNN keyword algorithm proposed in this study has good performance in detecting keywords in speech.

3.2 Comparative testing of speech detection algorithms and keyword detection algorithms

To further verify the effectiveness of speech detection algorithms and keyword detection algorithms in improving PaddleSpeech model speech detection, the study compared them with other algorithms separately. Firstly, the speech enhancement algorithm based on spectral subtraction and wavelet threshold denoising proposed by the research was compared with the Kalman filter method. The experimental data was obtained from a large conference speech text, and the results are shown in Figure 9.



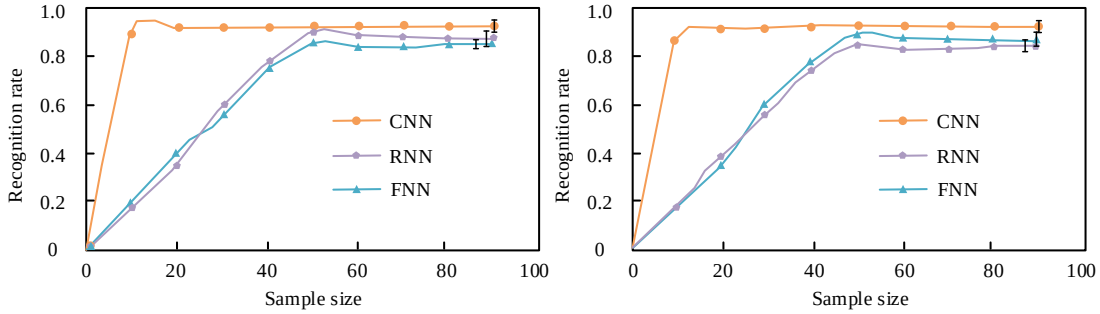
(a) Short sentence keyword detection (b) Long sentence keyword detection

Figure 8: Results of keyword detection for long and short sentences

(a) Speech enhancement algorithm proposed by the research institute

(b) Kalman filter algorithm

Figure 9: Comparison results of two algorithms



(a) Specific person

(b) Non-specific person

Figure 10: The influence of sample size on recognition rate

As shown in Figure 9 (a), there were significant differences in the distance between clusters of different speech texts. Specifically, the distance between clusters “go” and “down” was relatively close, while the distance between clusters “go” and “up” was relatively far. This indicates that speech enhancement algorithms reduced the distance between similar speech text categories and increased the distance between speech texts with significant differences by enhancing speech and denoising methods. In Figure 9 (b), the cluster center representing “off” was closer to the cluster center representing “yes”, and there was a significant cross fusion phenomenon between the two clusters. This indicates that the Kalman filter method had lower recognition accuracy than the proposed speech enhancement algorithm in distinguishing similar speech text categories. Overall, the feature space distribution in Figure 9 (a) was more “ideal”, with clearer boundaries between different speech texts, which was more conducive to improving the model's recognition ability. For keyword detection algorithms, the fewer samples required while ensuring recognition rate, the better. The accuracy of keyword detection is above 0.5 for both long and short sentences. Through t-test on the result data, it was found that the p-values of the obtained test data were all below the significance level of 0.05. Therefore, this study took specific person speech and non-specific person speech as examples, and used CNN, RNN, and FNN algorithms for keyword detection. The results are shown in Figure 10.

In Figure 10 (a), for a specific person's speech, the CNN algorithm exhibited a high recognition rate when the sample size was less than 20 in the early stage, and its recognition rate curve showed a rapid growth trend, with a stable recognition rate of 0.9 in the later stage. In contrast, the recognition rate curves of RNN and FNN algorithms grew slightly slower, and their recognition rates only stabilized at 0.8 when the sample size reached 60. Both algorithms had a larger sample demand than CNN algorithms and lower recognition rates. Figure 10 (b) shows the recognition rate curve of the three algorithms under non-specific human speech conditions. The case of non-specific speech, the impact of sample size and noise on the algorithm was not much different from that of specific people, and its recognition rate was not significantly reduced compared to specific person speech. The accuracy of keyword detection is above 0.5 for both long and short sentences. Through t-test on the result data, it was found that the p-values of the obtained test data were all below the significance level of 0.05. From this, the CNN algorithm was equally applicable to both specific and non-specific speech. The main reason for choosing CNN over RNN and FNN is that CNN has higher computational efficiency and better performance when processing large-scale datasets, especially suitable for complex tasks such as image and natural language processing. Secondly, CNN exhibits stronger adaptability, capable of automatically learning data features and flexibly adapting to various application scenarios. In addition, CNN can better utilize the parallel processing capabilities of modern hardware such as GPUs, thereby improving training and inference efficiency. Finally,

CNN typically provides more stable and reliable results in multiple experiments. These advantages make CNN a better choice in practical applications.

3.3 Application testing of C-S AddleSpeech model

To test the validity of the C-S AddleSpeech model in speech keyword detection, the study first set a total of 20 keywords in multiple audios and compared the C-S AddleSpeech model before and after improvement. The results are shown in Figure 11.

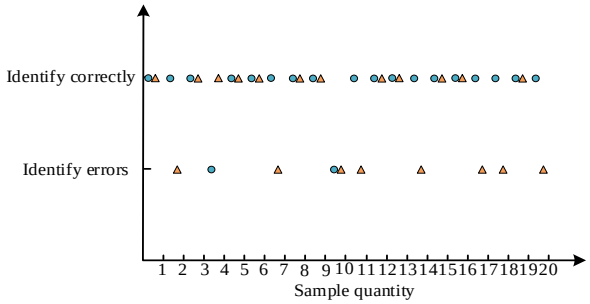


Figure 11: Comparison of recognition results between two models

In Figure 11, for the 20 keywords in the audio, the PaddleSpeech model recognized 12 correctly, while the C-S-PaddleSpeech model recognized 18 correctly. The former had an accuracy rate of 60%, while the latter had an accuracy rate of 90%, with a difference of 30% between the two. From this, the PaddleSpeech model significantly improved its ability to recognize keywords after keyword recognition optimization. The accuracy of keyword detection is above 0.5 for both long and short sentences. Through t-test on the result data, it was found that the p-values of the obtained test data were all below the significance level of 0.05. To test the recognition performance of the C-S AddleSpeech model in speech and audio, a comparison test was performed between the C-S AddleSpeech model and the SpeechRecognition model proposed by Google. The results are shown in Table 2.

Table 2: Comparison of C-S-PaddleSpeech and SpeechRecognition

Model	Total word count	Missing number	The number of multiple words	Error word count	Correct word count	Word accuracy rate (%)
C-S-PaddleSpeech model	15177	25	2	158	15019	98.9589
SpeechRecognition model	15177	54	9	584	14593	96.1520

In Table 2, the speech audio contained a total of 15177 characters. The C-S-PaddleSpeech model correctly recognized 15019 characters, with 25 missing words and 158 incorrect words. The overall recognition rate was high, reaching 98.9580%. The SpeechRecognition model correctly recognized 14593 characters, with 54 missing words and 584 incorrect words. The accuracy of keyword detection is above 0.5 for both long and short sentences. Through t-test on the result data, it was found that the p-values of the obtained test data were all below the significance level of 0.05. Both missing and incorrect words were at least twice as many as the C-S AddleSpeech model. From this, after optimizing the speech enhancement and keyword detection algorithms, the C-S-PaddleSpeech model had a high accuracy and excellent performance in speech recognition. In order to further determine the superiority of the research method, performance tests were conducted on the LibriSpeech dataset, as shown in Table 3.

Table 3: LibriSpeech dataset testing

Metric	CNN (research method)	RNN	Transformers
Dataset	LibriSpeech	LibriSpeech	LibriSpeech
Accuracy (%)	92.5	89	87.5
Training Time (hours)	5	7	6
Testing Time (ms/sample)	25	40	35
Memory Usage (GB)	3.2	4.5	4.0
Model Complexity (Million Parameters)	15	20	18

From Table 3, it can be seen that CNN outperforms RNN and Transformers methods in terms of accuracy, computation time, and memory usage. On the LibriSpeech dataset, the accuracy of CNN is 92.5%, significantly higher than RNN's 89.0% and Transformers's 87.5%. Through t-test on accuracy data, the results showed that the p-value between CNN and RNN was 0.002, and the p-value between CNN and FNN was 0.001, both of which were less than the significance level of 0.05, indicating that CNN was significantly

better than the other two methods in accuracy. In addition, the training and testing time of CNN are superior to RNN and Transformers indicating that it has a greater advantage in computational efficiency. Especially on large-scale datasets, the memory footprint of the model is smaller, making the research method more suitable for practical applications.

3.3 Discussion

The PaddleSpeech model has been improved by combining spectral subtraction and wavelet threshold denoising algorithms, significantly enhancing the accuracy of speech key information recognition. Compared with the LSTM algorithm used in related work (with a recognition accuracy of 94%), the research method demonstrates impressive performance in processing specific and non-specific speech. When the sample size is less than 20, the research method achieves a recognition accuracy of 90%. This indicates that the research method is superior to previous methods in the field of Chinese speech recognition. The technology studied has shown excellent performance in dealing with noise interference. By applying spectral subtraction, the signal-to-noise ratio can be effectively improved, making the extraction of key information clearer and reducing the negative impact of background noise on speech recognition. In addition, compared with the traditional Kalman filtering method, it was found that the acoustic feature space used was more ideal, effectively improving the discrimination of similar speech categories. Although research methods have advantages in accuracy and efficiency, their limitations cannot be ignored. For example, recognition accuracy may be affected in multiple languages or complex speech environments. This issue is particularly evident in multilingual conversations and may lead to a decrease in model performance. Therefore, future research can shift the focus to language adaptation optimization and introduce dynamic adjustment mechanisms to enhance application capabilities in various environments. In addition, multimodal fusion techniques can be explored, such as combining video information with speech data, to further enhance the accuracy and user experience of recognition systems. In summary, the study provides an effective solution for speech key information recognition technology by improving the PaddleSpeech model, promoting the development of this field and laying the foundation for future research directions.

4 Conclusion

In addressing the issue of insufficient accuracy in speech key information detection technology, a PaddleSpeech model was developed and optimized by integrating speech enhancement technology and keyword detection technology to raise the identification accuracy of speech key information. The experiment outcomes showed that for the same keywords in long and short sentences, the accuracy of keyword detection technology exceeded 0.5, demonstrating good detection performance. In addition, in the detection of specific and non-specific human speech, the CNN detection algorithm achieved a keyword recognition accuracy of 0.9 when the sample size was less than 20, while the FNN algorithm achieved an accuracy of 0.8 when the sample size reached 60. This indicates that CNN is superior to FNN in both sample size requirements and recognition accuracy. In the application testing of the model, the improved PaddleSpeech model performed significantly better than before in identifying 20 keywords in audio, with an accuracy rate of up to 90%. In comparison with the SpeechRecognition model for audio character recognition, the improved PaddleSpeech model correctly recognized 15019 characters with an accuracy of 98.9580%, while the SpeechRecognition model correctly recognized 14593 characters with an accuracy of 96.1520%, which was 2.806% higher than the latter. From this, the improved PaddleSpeech model raised in the study could effectively improve the accuracy of speech key information recognition. However, there are also some shortcomings in the research, such as the decrease in detection accuracy of the model due to differences in language rules or language structures in multilingual conversation scenarios, which need to be improved in future research.

References

[1] Žvirblis T, Pikšrys A, Bzinkowski D, Rucki M, Kilikevi č ius A, Kurasova O (2024). Data augmentation for classification of multi-domain tension signals. Informatica, pp. 883-908. https://doi.org/10.15388/24-INFOR578

[2] Adolfi F, Bowers J S, Poeppel D (2023). Successes and critical failures of neural networks in capturing human-like speech recognition. Neural Networks, pp. 199-211. https://doi.org/10.1016/j.neunet.2023.02.032

[3] Chen, Y. (2024). Human Resource Recommendation Based on CBCF-BAC and Short Text Similarity Algorithm. Informatica, pp. 99-112. https://doi.org/10.31449/inf.v48i22.6853

[4] Mena-Yedra R, López Redondo J, Pérez-Sánchez H, Martinez Ortigosa P (2024). ALMERIA: Boosting pairwise molecular contrasts with scalable methods. Informatica, pp. 617-648. https://doi.org/10.15388/24-INFOR558

[5] Navakauskas D, Kazlauskas M (2023). Fog computing in healthcare: systematic review. Informatica, pp. 577-602. https://doi.org/10.15388/23-INFOR525

[6] Osipov A, Pleshakova E, Liu Y (2024). Machine learning methods for speech emotion recognition on telecommunication systems. Journal of Computer Virology and Hacking Techniques, pp. 415-428. https://doi.org/10.1007/s11416-023-00500-2

[7] Kumar P, Malik S, Raman B (2024). Interpretable multimodal emotion recognition using hybrid fusion of speech and image data. Multimedia Tools and Applications, pp. 28373-28394. https://doi.org/10.1007/s11042-023-16443-1

[8] Belkebir, M., Maarouk, T. M., & Nini, B. (2024).

Real-time Semantic Healthcare System: Visual Risks Identification for Elders and Children. Informatica, pp. 65-82. https://doi.org/10.31449/inf.v48i14.6271

[9] Yuan Q, Dai Y, Li G (2023). Exploration of English speech translation recognition based on the LSTM RNN algorithm. Neural Computing and Applications, pp. 24961-24970. https://doi.org/10.1007/s00521-023-08462-8

[10] Xu Y, Su H, Ma G, Liu X (2023). A novel dual-modal emotion recognition algorithm with fusing hybrid features of audio signal and speech context. Complex & Intelligent Systems, pp. 951-963. https://doi.org/10.1007/s40747-022-00841-3

[11] Zhang Y, Park D S, Han W (2022). Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. IEEE Journal of Selected Topics in Signal Processing, pp. 1519-1532. https://doi.org/10.1109/JSTSP.2022.3182537

[12] Zhu P, Cheng D, Yang F (2022). Improving Chinese named entity recognition by large-scale syntactic dependency graph. IEEE/ACM Transactions on Audio, Speech, and Language Processing, pp. 979-991. https://doi.org/10.1109/TASLP.2022.3153261

[13] Ambrogio S, Narayanan P, Okazaki A (2023). An analog-AI chip for energy-efficient speech recognition and transcription. Nature, pp. 768-775. https://doi.org/10.1038/s41586-023-06337-5

[14] Burchi M, Timofte R (2023). Audio-visual efficient conformer for robust speech recognition. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2258-2267. https://doi.org/10.1109/WACV56688.2023.00229

[15] Deng K, Gao D, Guan Q, Wang F, Ma S, Zhao C, Lu Y (2023). Exploring the effectiveness of using a smartphone for position-oriented process monitoring. The International Journal of Advanced Manufacturing Technology, pp. 4293-4307. https://doi.org/10.1007/s00170-023-10984-3

[16] Chen J C, Liao C Y, Chen M H (2024). Using Virtual Reality Technologies to Cope with Grief and Emotional Resilience of Absent Loved Ones. Proceedings of the 2024 8th International Conference on Medical and Health Informatics, pp. 73-77. https://doi.org/10.1145/3673971.3674018

[17] Wu J, Chen S, Xiang W, Sun L, Zhang H, Zhang Z, Li Y (2024). CNAMD Corpus: A Chinese Natural Audiovisual Multimodal Database of Conversations for Social Interactive Agents. International Journal of Human‐Computer Interaction, pp. 2041-2053. https://doi.org/10.1080/10447318.2023.2228530

[18] Zhou L, Liu Z, Li Y (2024). Multi Fine-Grained Fusion Network for Depression Detection. ACM Transactions on Multimedia Computing, Communications and Applications, pp. 1-23. https://doi.org/10.1145/3665247

[19] Atmaja B T, Sasou A, Akagi M (2022). Survey on bimodal speech emotion recognition from acoustic and linguistic information fusion. Speech Communication, pp. 11-28. https://doi.org/10.1016/j.specom.2022.03.002

[20] Melnik-Leroy G A, Bernatavičienė J, Korvel G, Navickas G, Tamulevičius G, Treigys P (2022). An overview of Lithuanian intonation: a linguistic and modelling perspective. Informatica, pp. 795-832. https://doi.org/10.15388/22-INFOR502

[21] Ok M W, Rao K, Pennington J (2022). Speech recognition technology for writing: usage patterns and perceptions of students with high incidence disabilities. Journal of Special Education Technology, pp. 191-202. https://doi.org/10.1177/0162643420979929

[22] Blanco-Fernández Y, Gil-Solla A, Pazos-Arias J J, Quisi-Peralta D (2023). Automatically Assembling a Custom-Built Training Corpus for Improving the Learning of In-Domain Word/Document Embeddings. Informatica, pp. 491-527. https://doi.org/10.15388/23-INFOR527

A Comparative Analysis of Ensemble–Metaheuristic Algorithms for Copper Price Forecasting: The NGO-AdaBoost Hybrid Approach

Xianming Liu
School of Economics and Management, Hubei Polytechnic University, Huangshi 435000, China
E-mail: Liu.xm@126.com

Keywords: copper price forecasting, global price of copper, adaboost, northern goshawk optimization, artificial intelligence (AI)

Received: March 3, 2025

On international trade marketplaces, copper prices fluctuate a lot. Since copper is a valuable material, changes in its price may have an impact on certain nations' economies' ability to grow sustainably. The price of copper is a major issue for investors, policymakers, and futures traders. Governments and businesses that rely on copper mining must be able to forecast copper prices to make critical choices. This price is predicted in this article using an artificial intelligence technique. This paper proposes a hybrid forecasting model using the Adaptive Boosting model and the Northern Goshawk Optimization algorithm to enhance copper price predictions in terms of accuracy and reliability. A hybrid NGO-AdaBoost model was created to benefit from both the efficacy of ensemble learning and the exploratory potential of metaheuristic optimization. For this reason, time series datasets of copper prices from January 2014 to October 2023 were created using historical data, including open, high, low, and close prices and volume. Key metrics were used to assess the predictive accuracy of the model, and the results demonstrated high predictive accuracy with an R-squared (R^2) value of 0.9919, a Mean Absolute Percentage Error (MAPE) of 0.6733, and a Mean Absolute Error (MAE) of 0.0268 on the test dataset. Additionally, comparative analysis shows that the suggested NGO-AdaBoost model outperforms other benchmark techniques and traditional forecasting models like Autoregressive Integrated Moving Average and Support Vector Regression in terms of prediction, accuracy, and stability. This confirms the model's capacity to capture the intricate dynamics of copper price volatility. The results demonstrated that the proposed hybrid scheme could more reliably and accurately forecast the price of copper than the other schemes used in this study. The research presented in this paper provides a reliable source for predicting future changes in copper prices.

Povzetek: Podan je pregled in nato opisan izviren hibridni NGO-AdaBoost model za napoved cene bakra, ki združuje metahevristično optimizacijo in ansambelsko učenje ter dosega bolj kvalitetno napovedovanje kot ARIMA, SVR in LSTM.

1 Introduction

Copper is a valuable metal for many industries and plays a significant role in many financial sector enterprises. Copper is the third in the global consumed metal after iron and aluminum because of its malleability and conductivity. For evaluating the cost, availability, and demand of copper, the futures markets of renowned futures trading exchanges serve as the principal trading platforms, like LME (London Metal Exchange), COMEX (New York Commodity Exchange), and SHFE (Shanghai Futures Exchange) [1,2]. It is important to mention that these factors are greatly impacted by investment flows and currency exchange rates [3]. The financial success of several sectors is directly contingent upon the fluctuation in metal prices, including the price of copper. Copper production exerts a significant reliance on developing economies, influencing their trajectory both domestically and strategically [4]. Price fluctuations for copper have a big effect on the fiscal or financial income of a nation or a corporation, making it a crucial aspect to consider. The cost of copper is predominantly influenced by the efficiency of copper providers in extracting as well as transporting the metal, together with the demand for products and services that rely on copper. Regulators, investors, and companies that manufacture copper must all be able to predict supply and demand shocks and market patterns with precision [5]. Accurate pricing projections enable policymakers to manage the market with more effectiveness. Copper producers can enhance the efficiency and responsiveness of their production processes. As a result, investors may more effectively develop lucrative investing plans for both the short and long term. One of the biggest obstacles to successful forecasting is determining the precise traits of a time series [6]. Because financial time series are nonlinear, volatile, and regime-shifting, forecasting copper prices remains extremely challenging despite its significance. Particularly in times of market turbulence, traditional statistical models frequently fail to capture these intricate dynamics, producing forecasts that are unstable or erroneous. Numerous econometric schemes, including

wavelet-ARIMA [7], generalized autoregressive conditional heteroscedastic (GARCH) schemes [8,9], as well as ARIMA (autoregressive integrated moving average) [3,8], have been created to predict copper prices. These time series model; however, are unable to adequately represent the time series' nonlinear nature. As a result, advances in computational intelligence, machine learning (ML)[10][11][12], and AI tactics have been thoroughly researched and employed to determine and document diverse traits of the investigated time series of copper prices [1–4,8].

To address the issue of modeling the nonlinear and unstable nature of copper prices, this study proposes a hybrid artificial intelligence (AI) model that integrates ensemble learning and metaheuristic optimization. The main goal is to create a strong model that performs better than traditional machine learning techniques and can better represent the dynamic behavior of copper prices. This study provides the assumption that by utilizing the global optimization power of the Northern Goshawk Optimization algorithm and the adaptive learning capability of AdaBoost, the suggested NGO-AdaBoost hybrid model will outperform conventional models in forecasting accuracy. Metaheuristic algorithms are generic search strategies used to efficiently explore big and complicated solution spaces, which is thereby extremely appropriate for optimization issues. To this extent, an NGO is employed to optimize the individual base learners within the AdaBoost algorithm to achieve maximum accuracy and stability against evolving market conditions. This study helps bridge the gap by presenting a more precise and reliable approach for predicting copper prices, serving as a helpful tool for investors, policymakers, and futures traders to analyze upcoming changes in copper prices and make well-informed choices.

Ensemble classifiers are a popular AI technique for data categorization. Many weak classifiers, or those with a lower classification accuracy, are utilized in the ensemble classifiers approach. Every classifier has a weight that influences the classification outcome. The process of applying these weights to determine the categorization outcome is known as "weight voting." One of the most popular ensemble classifier tactics that takes advantage of the weight-voting process is AdaBoost. The proposal was made by Freund and Schapire in 1997 [13]. The AdaBoost technique may improve classification accuracy and is simple to use. It may be used for many classifiers as well.

The Northern Goshawk Optimization (NGO) offers highly desirable quasi-optimal solutions for optimization problems. It excels in addressing real-world scenarios and exhibits impressive performance in solving optimization issues. Moreover, it outperforms comparable algorithms by effectively balancing the exploration and exploitation processes to identify ideal resolutions. Fewer input parameters are needed for the recommended optimizer. The northern goshawk is the sole member of the Accipiter genus that preys on a diverse range of animals [14]. Males in this species are somewhat more widespread than females, and they are spread in both Eurasia and North America. The northern goshawk divides its hunting process into two stages: the first involves it moving swiftly in the direction of its target when it spots it, and the second involves a brief tail chase to catch up with the victim [15]. This article examines the NGO algorithm's capacity to predict the price of copper as a consequence. This work's contribution is broken down as follows:

- Acknowledged the influence that copper price volatility has on economies and the international trade markets, emphasizing the significance of precise copper price prediction for investors, policymakers, and futures traders.
- This study contributes to the field by showcasing the potential of metaheuristic optimization in ensemble learning models and proposing the first integration of an NGO with AdaBoost for copper price forecasting. Furthermore, it offers a comprehensive examination of multiple metaheuristic optimization algorithms concerning the prediction of copper prices, offering valuable insights into how different optimization techniques impact forecasting accuracy. By emphasizing the benefits of incorporating advanced optimization algorithms, this study paves the way for more reliable and efficient predictive models in commodity price forecasting. The study also contributes to the literature by demonstrating how hyperparameter tuning enhances model performance and provides a framework for additional research in this field.
- Nine years and 10 months' worth of copper price time series datasets, including volume and the open, high, low, and close values, were gathered from historical data.
- Developed the NGO-AdaBoost model and showed that, with a coefficient of determination (R^2) of 0.9919, it performed better in forecasting copper prices when compared to other ML tactics.

This article is organized in the following manner: A comprehensive examination of the data source and all of its components is provided in Section 2. The scheme, optimizers, and other elements utilized in this paper are analyzed in the third part. In the fourth and fifth parts, the outcomes of the schemes utilized in this article are compared. After evaluating the chosen schemes in this article, the conclusion is covered in the next section, and the last section discusses potential future projects that might build on the scheme employed in this article.

2 Literature review

Esperanza García-Gonzalo et al. [16] concentrated on predicting copper spot prices from COMEX by using support vector regression (SVR) and several model architectures. The accuracy of three-time series analyses was examined, and the hybrid direct-recursive approach emerged as the most accurate in terms of numerical outputs. Jiahao Chen et al. [17] utilized an LSTM (Long Short-Term Memory) AI model to make forecasts about the pricing of copper. The effectiveness of the LSTM model was improved by using a simulated annealing approach to tune hyperparameters. Relationship

assessment was utilized to address feature engineering. The scheme used economic factors that have a strong correlation with copper prices, such as the West Texas Intermediate Oil Price, Gold Price, and Silver Price.

In a study by Gabriel Astudillo et al. [18], SVR was specifically used to predict the closing prices of copper at LME over multiple periods, with grid search and balanced cross-validation being used to select the best model for each forecast timeframe. The experimental results demonstrate that the parameters of SVR remain constant over prediction intervals and the prior values used for estimations.

A novel forecasting model that blends interval and point forecasting is presented by Yifei Zhao et al. [19]. The interval forecasting model was validated by gathering the interval forecasting results and using the copper and aluminum pricing data. The evaluation results show that the accuracy of the LSTM model is 0.099021.

Hongyuan Luo et al. [20] introduce a novel multi-step-ahead forecasting model of copper price with an error correction scheme-based and genetic algorithm-enhanced long short-term memory (GA-LSTM) structure. The predictive efficacy of the introduced framework is supported by a 30-year real copper price time series. A 30-year real iron ore price time series is also utilized to show the robustness and generalization capability of the introduced design. The outcomes demonstrated that the F-TS-GA-LSTM-EC hybrid model was effective and more accurate in copper price prediction.

Table 1: Summary of literature review on copper price forecasting.

Authors	Method	Dataset Description	Performance Metrics	Year
García-Gonzalo et al. [16]	SVR (Support Vector Regression)	COMEX monthly copper spot prices (1960-2019)	MAE: 144.21 RMSE: 170.15 MAPE: 2.36%	2023
Jiahao Chen et al. [17]	LSTM + Simulated Annealing	Daily copper + macroeconomic indicators (1990-2009), from Investing.com	MSE: 0.000569 (test set)	2023
Gabriel Astudillo et al. [18]	SVR + Grid Search + Cross-Validation	Daily copper prices from LME (2006-2018)	RMSE: 0.0177 (5-day) R = 0.9582	2020
Yifei Zhao et al. [19]	VMD-SSA-LSTM + Interval Forecasting	Daily copper and aluminum prices (2012-2022), Shanghai Futures Exchange	R^2 = 0.99827 RMSE = 763.01 MAE = 543.20 MAPE = 0.8789%	2023
Hongyuan Luo et al. [20]	F-TS-GA-LSTM-EC (Factor + Time Series + GA-LSTM + Error Correction)	Monthly copper prices + 8 factors (1991-2021), 360 records, IndexMundi	R^2 = 0.920 MAE = 257.26 RMSE = 330.04 MAPE = 4.03%	2022
Hasel Amini Khoshalan et al. [21]	GEP, ANN, ANFIS, ANFIS-ACO	Monthly copper + 8 influencing factors (1990-2020), IndexMundi	R^2= 0.981 RMSE = 356.51 MAE = 239.11	2021

Note: SVR=Support Vector Regression, LSTM= Long Short-Term Memory, SA= Simulated Annealing, VMD= Variational Mode Decomposition, SSA= Sparrow Search Algorithm, GA= Genetic Algorithm, F-TS-GA-LSTM-EC= Factor + Time Series + Genetic Algorithm-optimized LSTM + Error Correction, GEP= Gene Expression Programming, ANN= Artificial Neural Network, ANFIS= Adaptive Neuro-Fuzzy Inference System, ACO= Ant Colony Optimization, R^2= Coefficient of Determination, MAE= Mean Absolute Error, RMSE= Root Mean Square Error, MAPE= Mean Absolute Percentage Error, COMEX= Commodity Exchange (New York), LME= London Metal Exchange.

Statistical models such as ARIMA and GARCH, while simple, often do not capture the non-stationary and non-linear commodity price dynamics. They are incapable of responding well to the incidence of structural breaks and abrupt changes that are common in metals markets. Machine learning models such as SVR and standalone LSTM have done better than statistical models but still fail to capture the short- and long-term dependencies simultaneously. SVR, in particular, does not produce consistent performance for varying forecast horizons due to its sensitivity to kernels. More sophisticated hybrids, such as GA-LSTM and ANFIS-ACO, have high improvement, but they can be vulnerable to parameter tuning sensitivity, local optima, or low robustness to unforeseen volatility patterns. These drawbacks motivated the design of the NGO-AdaBoost hybrid model, which combines.

- The discoverability of NGOs to correct model parameters and break out of local minima
- AdaBoost's adaptive ensemble learning capability to strengthen weak learners and counteract overfitting, especially in noisy or unstable data conditions

- A strong design tested with a 9 years and 10 months' time series, ensuring its durability, accuracy, and reactivity to complex patterns in copper price dynamics.

3 Dataset

This research examines the prediction of copper prices. From January 2014 to October 2023, pricing data was collected for the input parameters reviewed and changes in copper prices. The prices in this article are expressed in US dollars per pound (in USD/LBS). Various variables may affect copper price fluctuations. The variables used in this article are volume and open, high, low, and close prices.

3.1 Data Pre-processing

Historical copper prices based on global trends were obtained from Yahoo Finance and TradingView. Data cleaning was done intensively to handle any possible missing or inconsistent values in the data to ensure that the data input into the machine learning models is accurate and clean. Data cleaning involves the detection and fixing of errors, inconsistencies, or missing values in the dataset so that data input into the machine learning models is clean and accurate. Because missing or erroneous data could lead to faulty predictions and stop the model from functioning, this process is imperative. Following data cleaning, feature normalization was conducted so that all input features, like open, high, low, volume, and close prices, were normalized and values were between 0 and 1. Standardizing features enhances machine learning models because it allows them to learn from features of different scales. To test the model, the data was split into training and test sets, where 80% was used for training and 20% for testing. This split enables the generalization of the model to unseen data to be tested by preventing overfitting and ensuring that the performance of the model can be properly measured. This 80/20 ratio is common in machine learning to achieve a balance between training the model sufficiently and testing it on a big enough sample of unseen data.

3.2 Historical data

The opening price is the price of a share at the start of trading and is a reliable indicator of the daily volatility of the financial market. Because the financial market is like an auction where buyers and sellers compete to find the highest bidder, the opening price does not necessarily have to match the closing price of the day before.

The previous day's highest and lowest prices are captured, which gives data about the average volatility of the market on a trading day and how it impacts the final closing price. The adjusted closing price is the final price of a property after adjustment for any dividends or corporate activity that has happened before the opening of the subsequent market day.

Adjusted closing price is generally employed for analyzing the current returns or carrying out a detailed analysis of past returns.

Volume is the total number of contracts or shares that are transferred in a security or the entire market within a definite time frame.

To build an efficient input space for prediction, the five important features of the copper price data—open, high, low, close, and volume (OHLCV)—were utilized. These are widely employed in financial time series analysis because both indicate the direction of price as well as activity in the market during a period. Before training, all variables were normalized using min-max scaling to achieve consistency across different ranges. With all five variables, the highest prediction and stability among the models under test were obtained, confirming their individual and combined roles in the forecasting model.

3.3 Statistical values

The dataset is thoroughly examined in the report, which is shown in Table 2. A comprehensive statistical representation of the data pertaining to the input and output attributes is given in the following table. This process guarantees that all of the information can be understood completely. The table contains a large number of statistical measures, including the mean, variance, count, 25%, 50%, 75%, minimum (min), maximum (max), and standard deviation (Std.). By using these measurements, a thorough and accurate analysis of the data can be accomplished.

Table 2: A statistical summary of the dataset

	count	mean	std.	min	25%	50%	75%	max	variance
Open price	2467	3.103505	0.699136	1.9385	2.617	2.95	3.5855	4.92	0.488791
High price	2467	3.132707	0.707937	1.98	2.64175	2.977	3.63	5.0395	0.501175
Low price	2467	3.073633	0.690216	1.9355	2.5975	2.9225	3.556	4.81	0.476399
Volume	2467	31651.08	38277.74	10	290	1500	61380	230070	1.47E+09
Close price	2467	3.103989	0.6999	1.9435	2.62025	2.951	3.58725	4.9375	0.48986

4 Methodology

This work presents the design of an FOA-AdaBoost, SSA-AdaBoost, and NGO-AdaBoost model for predicting copper prices. These schemes are described in a general sense in this section. The AdaBoost model and the FOA, SSA, and NGO algorithms are initially proposed as follows:

4.1 Adaptive boosting

One of the most powerful recognition strategies is AdaBoost, which blends many weak predictors to create an effective predictor [22] [23] . AdaBoost training, by Lu et al. [24], has greater dispersion weights in the case of greater errors and smaller dispersion weights when errors are lesser. To improve the expected outcome, the samples are subsequently trained using the revised weight dispersion [25]. The following are the AdaBoost computation stages [26] [27].

$$U = \{(x_i, y_i)\}_{i=1}^{N}, y_i \in \{-1, +1\} \quad (1)$$

Step 1: Initialize weights

Set sample weights as uniform:

$$D_1(i) = \frac{1}{N}, \text{ for } i = 1,2, \dots, N \quad (2)$$

Step 2: Train weak learner error:

$$\varepsilon_t = \sum_{i=1}^{N} D_t(i) \cdot \mathbb{I}(f_t(x_i) \neq y_i) \quad (3)$$

Step 3: Calculate learner weight

The importance weight of the learner is computed as:

$$\alpha_t = \frac{1}{2} \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \quad (4)$$

Step 4: Update sample weights

Update the weight distribution:

$$D_{t+1}(i) = \frac{D_t(i) \cdot e^{-\alpha_t y_i f_t(x_i)}}{Z_t} \quad (5)$$

where Z_t is a normalization factor ensuring that $\sum_i D_{t+1}(i) = 1$.

Step 5: Final hypothesis

Following T rounds, the terminal strong classifier consists of a weighted ensemble of the weak learners:

$$F(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t f_t(x)\right) \quad (6)$$

Loss Function:

AdaBoost attempts to minimize the exponential loss function over the training data:

$$\mathcal{L}(F) = \sum_{i=1}^{N} e^{-v_i F(x_i)} \quad (7)$$

This loss function is convex and imposes a larger penalty on misclassified samples, forcing the model to focus on hard-to-classify points.

The steps above define AdaBoost's internal training logic, but the choice of hyperparameters such as n_estimators, learning_rate, and random_state significantly affects the algorithm's performance. Before the boosting process begins, this research optimizes these parameters using the NGO algorithm. NGO is employed as a pre-training hyperparameter tuning technique to ascertain the ideal configuration of AdaBoost's hyperparameters through a two-phase bio-inspired search strategy (local exploitation and global exploration). This optimization lowers validation loss while increasing the final model's accuracy and generalization. Therefore, without altering AdaBoost's internal processes, the NGO strategically improves its conditions.

4.2 Metaheuristic optimization of adaboost hyperparameters

Five metaheuristic algorithms were utilized to fine-tune the most important hyperparameters of the AdaBoost model to maximize its predictive performance: GA, PSO, FOA, SSA, and NGO. These include the learning rate, number of estimators (n_estimators), and random state initialization.

To minimize AdaBoost's exponential loss on a validation set, each metaheuristic investigated this search space. Table 3 displays the outcomes of the optimization procedure. Remarkably, the NGO algorithm produced the best overall performance across all evaluation metrics (R^2, MAPE, MAE, and RSE) with 800 estimators, a learning rate of 0.25, and a random state of 64. These results show that different algorithms converge on different parameter configurations. For example, GA and PSO favored higher estimator counts (1500 and 1200, respectively), while FOA and SSA selected somewhat lower values (700 and 1000). The higher learning rate (0.25) selected by NGO in comparison to GA and PSO (0.1) may have allowed for faster convergence and more responsive adaptation to data variations, especially in volatile financial time series like copper prices. The success of the NGO algorithm can be attributed to its two-phase hunting approach, which successfully balances global exploration (by identifying prey at random) with local exploitation (by pursuing and adjusting). This dual mechanism ensures more stable parameter tuning and helps avoid premature convergence, which is common in simpler algorithms like FOA, in contrast to SSA, which can lead to unpredictable updates in noisy datasets. Overall, this test demonstrates that an NGO is appropriate as the primary optimizer in the proposed framework for forecasting copper prices because it provides the most effective hyperparameter configuration for AdaBoost in addition to a dependable search procedure.

Table 3: AdaBoost hyperparameter optimization with five metaheuristic algorithms.

Ada Boost	Upper and lower band	GA	PSO	FOA	SSA	NGO
n_estimators	[100, 2000]	1500	1200	700	1000	800
learning_rate	[0.0001, 1]	0.1	0.1	0.2	0.2	0.25
random_state	[2, 100]	32	28	64	74	64

Take note that Figure 1 shows the pseudocode for integrating the AdaBoost model with Northern Goshawk Optimization (NGO). The figure demonstrates how the NGO optimizes the AdaBoost algorithm's primary hyperparameters, including the number of estimators, learning rate, and random state. AdaBoost performs better and is better able to handle the intricacies of the copper price prediction task thanks to this optimization process.

```
Inputs and initializations:
   - Input Training data (x_train, y_train)
   - Input Validation data (x_val, y_val)
   - Initialize searching boundaries for AdaBoost parameters
   - Initialize NGO parameters: population_size, max_iterations.

1. Initialize NGO:
   - Set individual's parameters as a vector.
   - Parameters: [n_estimators, learning_rate, max_depth]

2. Initial population assessment:
   For each individual:
      - Train AdaBoost model using the encoded parameters.
      - Compute fitness.

3. Do while i < max_iterations:
   a. For each individual:
      - Update current individual position:
         - Exploration and exploitation (Goshawk behavior).
      - Bound the found position to allowed boundaries.

   b. Evaluate New Individuals:
      - Build AdaBoost with new found hyperparameters.
      - Get new fitness value.

   c. Keep best results:
      - Keep best found individuals based on their fitness.
      - Update global best solution.

4. Return the best found individual as the optimal solution.

5. Train AdaBoost with optimal parameters found by NGO.
```

Figure 1: Pseudocode for the integration of northern goshawk optimization with adaboost.

4.3 Optimizer

4.3.1 Fruit fly optimization algorithm

The entrapment behavior exhibited by fruit flies served as the impetus for the formulation of FOA according to Fig. 2, a novel global optimization technique [28]. To determine the optimal solution to an issue with optimization, fruit flies are simulated by the FOA model. A specific flavor in the air is detected by employing sensitive olfactory organs, after which the target location is located by using the acute sight organs that have been retained. Vision, Osphresis foraging, population assessment, and initiation comprise the initial FOA.

The replies are first generated within a range defined by lower and upper borders. Here, $a_{x,y}$ displays the x-th resolution and the y subscript indicates the position of the element inside the x-th resolution, as defined by Equation (8).

$$a_{x,y} = \text{rand}\left(hb_y - lb_y\right) + lb_y \tag{8}$$

The lower limit is depicted as lb, whereas the upper bound is depicted as hb. The term "rand" refers to a haphazardly selected integer from a uniform distribution. Launching the population during the Osphresis foraging phase intensifies the alignment of each option, in which the solution is displaced a random distance from its existing position. The arithmetical expression is represented by Equation (9).

$$a_{x,y}^{(r+1)} = a_{x,y}^{(r)} \mp \text{rand } 0 \quad (9)$$

The current response is represented by $a_{x,y}^{(t)}$, while the fresh alignment is displayed by $a_{x,y}^{(t+1)}$. Additionally, a random value is selected from the range $O[2,2]$. The letter "t" displays the count of cycles. After the relocation, the fitness value is computed for each solution, and the avaricious selection procedure selects whether to keep the new position or the previous one. If the most recent response possesses a higher fitness value than the one before it, it will substitute the old answer. Alternatively, if the previous response remains within the population, the new solution will be declined. The algorithm halts and generates the best possible outcome when it reaches the specified termination condition. Fig. 3 illustrates the method.

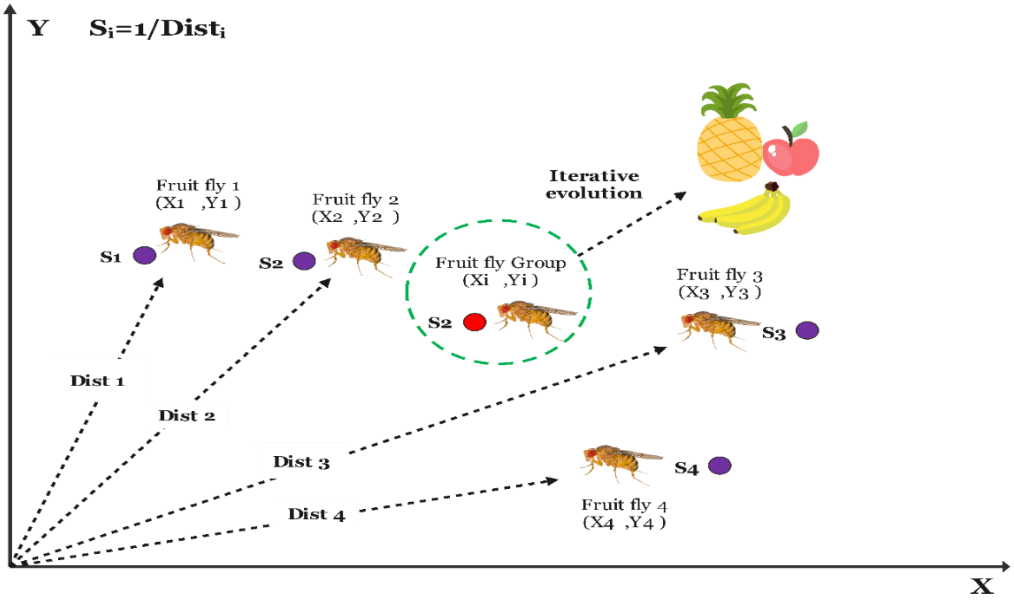


Figure 2: The trapping conduct shown by fruit flies

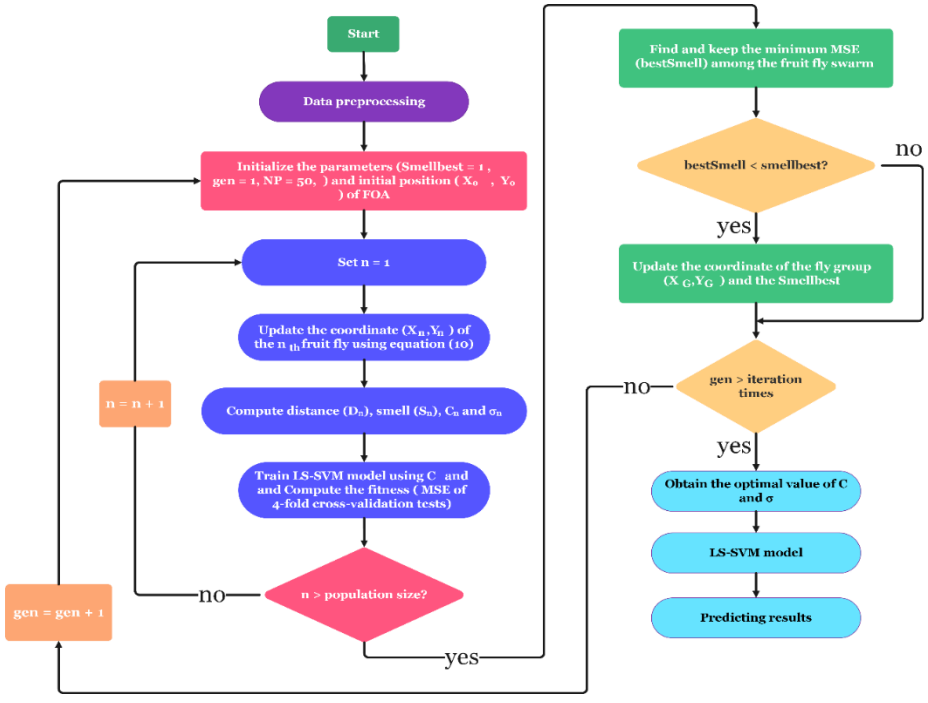


Figure 3: Flowchart of the Fruit Fly Optimization Algorithm to optimize the proposed model hyperparameters

4.3.2 Sparrow search algorithm

The SSA is a swarm enhancement technique that derives from the behavioral patterns of sparrows, shown in Fig. 4 [29]. The program categorizes the sparrow population into two distinct groups: producers and scroungers. Obtaining food and controlling population mobility fall within the purview of the producers. Producers can get a greater amount of food when the fitness value improves. The scroungers, however, actively seek out the producers and endeavor to enhance the solutions offered by them. Within each cycle of the method, the producer's position updates are computed using a particular formula, which is not specified in the provided paragraph. In essence, the SSA algorithm attempts to discover the most advantageous solution for an optimization issue by emulating the collective intelligence, foraging, and defense mechanisms shown by sparrows. The updated locations of the producers are as follows:

$$X_{ij}^{i+1} = \begin{cases} X_{ij}^{t} \cdot \exp\left(\dfrac{-i}{a \cdot \text{ iter}}\right) & \text{if } R_2 < ST \\ X_{ij}^{t} + Q \cdot L \text{ if } R_2 > ST \end{cases} \quad (10)$$

The provided equation has several variables and constants. The variable i ranges from 1 to N, with N being the size of the swarm. The variable t depicts the existing cycle. The variable j ranges from 1 to d, where d specifies the count of dimensions. The notation X_{1j}^{t} displays the value of the j-th dimension of the i-th sparrow at cycle t. The variable $\text{iter}_{\max}$ specifies the top range for the count of cycles. The constant α is a stochastic variable that takes on values between 0 and 1 with equal probability. The variable Q is a stochastic quantity that conforms to a Gaussian distribution. The matrix L is a $1 \times d$ matrix where all members are identical and equal to 1. The variables R_2 and ST indicate the alert value and the security cutoff, accordingly. The range of R_2 is from 0 to 1, whereas the range of ST is from 0.5 to 1.0. Regarding the scroungers, they observe the producers, and when they find a better source of food, they quickly abandon their existing location to compete intensely for the new resource. The equation dictating the adjustment of the scroungers' position is presented as follows:

$$X_{ij}^{i+1} = \begin{cases} Q \cdot \exp\left(\dfrac{X_{\text{monta}}^{t} - X_{ij}^{i}}{i^2}\right) & \text{if } i > n/2 \\ X_{p}^{i+1} + \left|X_{ij}^{t} - X_{p}^{t+1}\right| \cdot A^{+} \cdot L & \text{otherwise} \end{cases} \quad (11)$$

X_p was defined as the ideal position held by producers. The current worldwide worst location is considered to be the most unfavorable. The expression $A^{+} = AT(AA^{T})^{-1}$ describes the pseudo-inverse of matrix A, where A is a $1 \times d$ matrix with elements of 1 or -1 assigned at random.

When sparrows on the outside of the group feel danger, they quickly move to a safe region. Their conduct is driven by their desire to acquire a higher viewpoint. Conversely, sparrows positioned in the middle of the group exhibit a more unpredictable behavior, marked by aimless roaming, to get closer to surrounding sparrows. The mathematical model that governs this behavior may be formulated as follows:

$$X_{ij}^{t+1} = \begin{cases} X_{\text{best}}^{t} + \beta \cdot \left|X_{ij}^{t} - X_{brst}^{t}\right| & \text{if } f_i > f_g \\ X_{ij}^{t} + K \cdot \left(\dfrac{\left|X_{ij}^{t} - X_{\text{worrt}}^{t}\right|}{(f_i - f_w) + \varepsilon}\right) & \text{if } f_i = f_g \end{cases} \quad (12)$$

The above equation illustrates a mathematical formula for the Sum of Squared Differences (SSA) used in optimization situations. In this equation, X_{ber}^{t} depicts the current global optimum position, whereas X_{wors} indicates the existing global worst situation. The parameter β is a control variable that establishes the step size. It exhibits a normal dispersion with a mean of 0 and a variance of 1. K is a stochastic variable that falls inside the interval [-1,1]. The variables f_i and f_w represent the current sparrow's fitness values and the lowest fitness value, respectively. On the other hand, f_g displays the current global maximum fitness value. Furthermore, ε is the most minimal constant. The SSA utilizes this equation to choose the most favorable solution for an optimization issue.

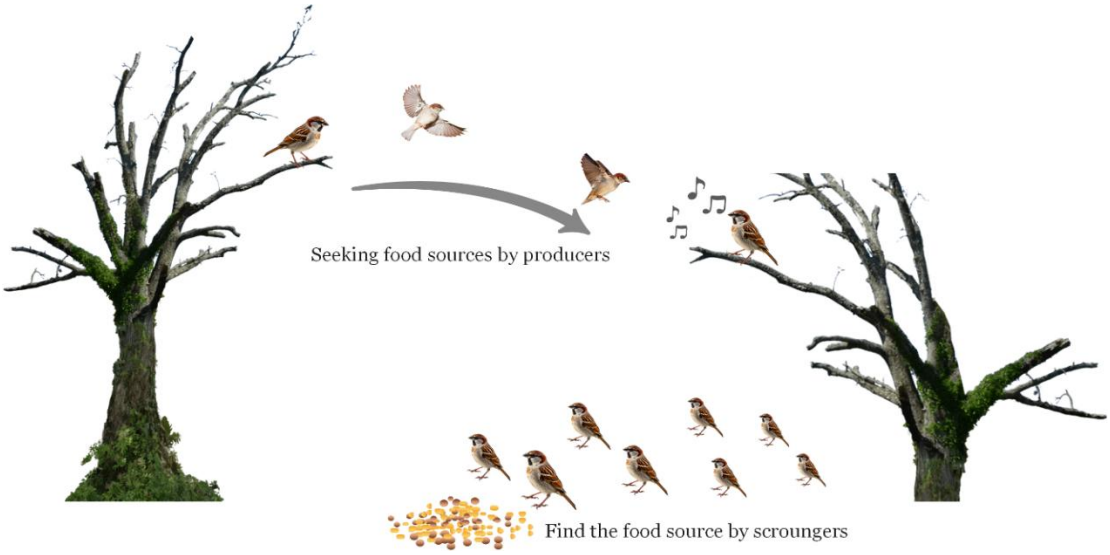


Figure 4: Behavioral diagram of the Sparrow Search Algorithm

4.3.3. Genetic algorithm

The genetic algorithm is a computational technique that mimics natural selection to solve optimization and search problems [30]. GA is composed of three fundamental components [31]. A set of textual or numeric symbols that each individual is assigned by the encoding entity is called a chromosome. The best encoding technique depends on the specific issue that needs to be resolved. The fitness function is also used to evaluate how accurately each person represented the answer. The fitness function was created specially to address the current problem. The evolutionary operators generate new individuals from preexisting ones through the processes of crossover, mutation, and selection. Crossover is a genetic process that combines the chromosomes of two distinct individuals to produce a new offspring. On the other hand, mutation causes random alterations to an individual's chromosomes. Selection is used to identify the individuals that reproduce the best.

4.3.4. Particle swarm optimization

Kennedy and Eberhart developed the PSO technique to address optimization issues. This method is based on the collective behavior observed in a swarm of particles [32]. Because the PSO technique tends to converge quickly and requires fewer parameters, it lowers computational overhead. Additionally, the likelihood of coming across a local solution that is not ideal is decreased by the thorough exploration carried out by several particles in pursuit of an ideal solution. Additionally, the algorithm features an efficient global search mechanism and is independent of derivatives. Each particle in the PSO searches a large search space for the optimal response. The search process begins with the random generation of candidate solutions, often called particles, in the search space. Particle velocities and fitness scores are often computed using a weighted mean of the number of features in the feature subset and the classification accuracy. This computation aids in updating the heading and velocity of their trajectories following the initial iteration, and the procedure is repeated until the stopping criterion is satisfied.

4.3.5. Northern goshawk optimization

NGO is a population-based metaheuristic inspired by the hunting mode of the northern goshawk (Accipiter gentilis), a very agile and efficient bird in hunting prey. The NGO algorithm replicates two principal stages of the goshawk's hunting process [33].

Major stages of the goshawk's hunting sequence:

(1) Prey identification and attack, and

(2) Pursuing and capturing prey through evasive flights.

Every agent (goshawk) of the population is a candidate solution to the optimization problem. The solutions are evaluated by an objective function and iteratively updated to converge to the optimum.

- Identifying and attacking prey.

As seen in Fig. 5, the northern goshawk haphazardly selects a victim during the first hunting phase and then quickly attacks it. This step improves the NGO's capability to explore by haphazardly selecting prey in the search domain. To choose the best site, this stage involves conducting a thorough search of the entire search domain. One way to articulate this is:

$$P_i = X_k, i = 1,2,\dots,N, k = 1,2,\dots\dots,i-1,i+1,\dots\dots,N \quad (13)$$

$$x_{i,j}^{\text{new},P1} = \begin{cases} x_{i,j} + r\left(P_{i,j} - Ix_{i,j}\right), F_{P_i} < F_i \\ x_{i,j} + r\left(x_{i,j} - P_{i,j}\right), F_{P_i} \geq F_i \end{cases} \quad (14)$$

$$X_i = \begin{cases} X_i^{\text{new},P1}, F_i^{\text{new},P1} < F_i \\ X_i, F_i^{\text{new},P1} \geq F_i \end{cases} \quad (15)$$

where: P_i displays the prey location of the i^{th} Northern goshawk, F_{P_i} is the value of its objective function, k is a haphazardly selected natural number from the range [1, N], $X_i^{\text{new},P1}$ depicts the updated status of the i^{th} recommended resolution, with its i^{th} dimension being $x_{i,j}^{\text{new},P1}$, $F_i^{\text{new},P1}$ indicates the value of the goal function based on the NGO's starting stage, and r and I are random numbers utilized to generate random behavior for the NGO during search and update. The variable (r) displays a haphazardly created number within $[0, 1]$, whereas the variable (I) displays a haphazardly generated number within [1, 2].

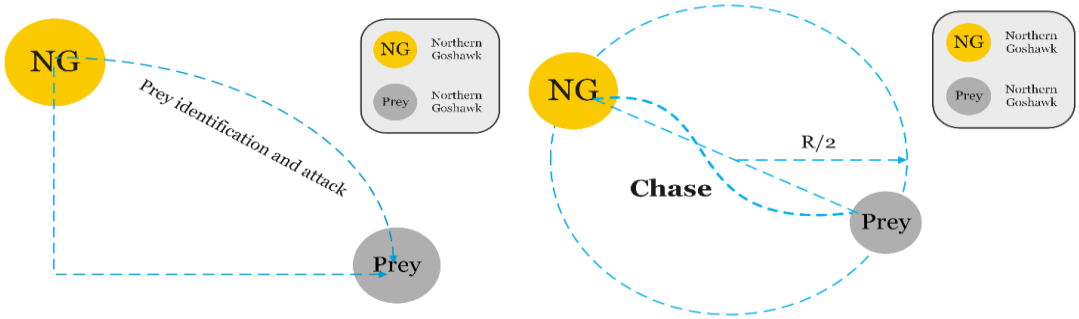


Figure 5: Prey identification and attack mechanism in Northern Goshawk Optimization

Phase (2): Engaging in pursuit and evasive maneuvers.

Following the attack, the goshawk pursues its escaping prey. This behavior is mimicked to make use of the search space locally to improve convergence. The mathematical representation of the ideas communicated in the second step may be expressed as Equations (16) - (18) in the following manner:

$$x_{i,j}^{\text{new},P2} = x_{i,j} + R(2r-1)x_{i,j} \quad (16)$$

$$R = 0.02\left(1 - \frac{t}{T}\right) \tag{17}$$

$$X_i = \begin{cases} X_i^{\text{new},P2}, F_i^{\text{new},P2} < F_i \\ X_i \quad\quad , F_i^{\text{new},P2} \geq F_i \end{cases} \tag{18}$$

The maximum cycle number is depicted as T, whereas the cycle counter is depicted as t. The parameter $X_i^{\text{new},P2}$ depicts the updated value of the i^{th} solution, and its dimension is $x_{i,j}^{\text{new}\,P2}$. The objective function derived from the second phase is depicted as $F_i^{\text{new,P2}}$. Fig. 6 illustrates the whole procedure of this optimizer.

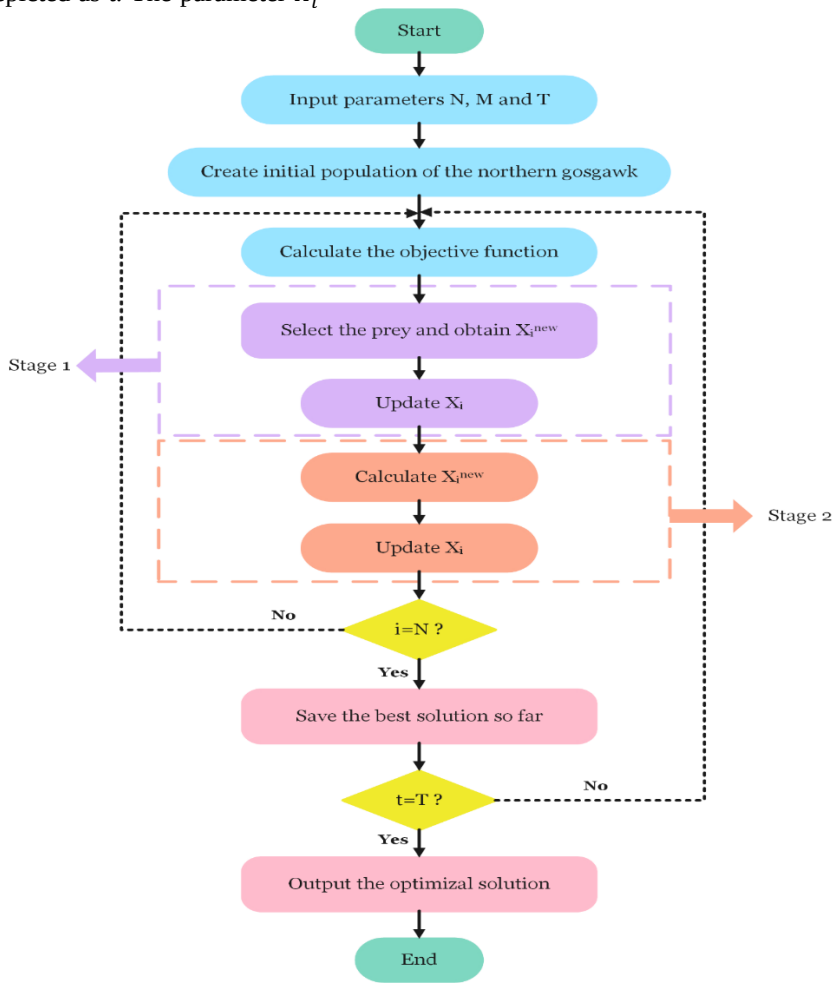


Figure 6: The NGO flowchart

The selection of the NGO was ultimately based on its biologically inspired two-phase optimization strategy, which successfully traverses the solution space by combining local exploitation (pursuit and attack) and global exploration (prey identification). This structure helps NGOs avoid overfitting and premature convergence, which are common issues in high-dimensional optimization. A scalable and suitable optimizer for nonlinear financial forecasting tasks that necessitate prompt decisions, like predicting the price of copper, an NGO can also benefit from its lightweight design, fewer control parameters, and fast convergence.

4.4 Model performance metrics

Metrics, or measurements, are utilized to review the productivity of a scheme, system, or process. The accuracy of the future estimates was evaluated based on many performance factors. Following a comprehensive investigation, these criteria were created to offer a thorough assessment of the estimates' dependability and correctness. Holdout data is required for trained machine learning schemes to perform properly to assess the difference between predicted and observed labels using various metrics. The metrics utilized to appraise the recommended scheme's prediction execution include R^2, relative square error (RSE), MAE, and MAPE.

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{19}$$

$$\text{MAPE} = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) \times 100 \tag{20}$$

$$\text{RSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(\bar{y} - \hat{y}_i)} \tag{21}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{22}$$

5 Outcomes

The primary goal of this study is to create and assess the best hybrid algorithm currently in use for precise copper

price prediction. The objective of this endeavor was to provide dependable data to analysts and investors, empowering them to make knowledgeable financial assessments. This paper combined AdaBoost and optimizers like FOA, SSA, and NGO to review the volatility traits of international copper futures prices and the flexible link of influencing factor variables. The goal of this combo is to identify an algorithm that predicts copper prices more accurately and efficiently. This article uses daily international copper futures price data from January 2014 to October 2023. The data set used in this article was gathered utilizing an OHLC price and volume. Included in the assessment criteria are RSE, MAE, MAPE, and R^2. An Intel Core i7-10700KF processor operating at 3.80 GHz, paired with an NVIDIA GeForce GTX 3080ti graphics processing unit and 32 GB of RAM access memory, was used to implement the model in an advanced computing environment. Complex financial forecasting tasks can be handled efficiently with this setup. To improve computational efficiency and model performance, Python version 3.12.3 was used for the implementation, along with sophisticated libraries like NumPy (1.26.4), Pandas (2.2.3), Scikit-learn (1.5.2), and TensorFlow (2.17.0). Table 4 showcases the numerical assessment of the scheme's predictive precision.

Table 4: The outcomes of the proposed models' evaluation criteria.

Schemes/Metrics		AdaBoost	GA-Adaboost	PSO-Adaboost	FOA-AdaBoost	SSA-AdaBoost	NGO-AdaBoost
Train	R^2	0.9857	0.9864	0.9873	0.9893	0.9925	0.9978
	MAPE	1.6457	1.6261	1.5868	1.6860	1.2753	0.7138
	MAE	0.0519	0.0504	0.0498	0.0480	0.0371	0.0209
	RSE	0.1014	0.0984	0.0952	0.0872	0.0730	0.0400
Test	R^2	0.9836	0.9843	0.9856	0.9865	0.9906	0.9919
	MAPE	0.9655	0.9541	0.9468	0.9426	0.8124	0.6733
	MAE	0.0381	0.0376	0.0370	0.0368	0.0298	0.0268
	RSE	0.0702	0.0687	0.0659	0.0633	0.0542	0.0494

6 Discussion

Positive outcomes are obtained by using the recommended NGO-AdaBoost model in this work. Metaheuristic optimization is crucial for improving the performance of ensemble learning models, especially for financial time series forecasting, where data is often nonlinear, volatile, and prone to sudden regime changes. Five hybrid models that integrated AdaBoost with various metaheuristic optimization algorithms were assessed in this study: Particle Swarm Optimization (PSO), FOA, SSA, Genetic Algorithm (GA), and the proposed NGO. A detailed comparison of these hybrid models across training and testing datasets using four standard performance metrics R^2, MAPE, MAE, and RSE is given in Table 4 and fig. 7. The results show that, although with differing degrees of success, all optimizers enhanced AdaBoost's performance by modifying its hyperparameters. Though their optimization dynamics were less stable in high-dimensional search spaces, GA-AdaBoost, PSO-AdaBoost, and FOA-AdaBoost all achieved moderate gains, particularly in lowering MAE and RSE values. SSA-AdaBoost demonstrated superior exploration and convergence capabilities compared to the previously mentioned models, resulting in significantly better generalization on the test set. On the other hand, NGO-AdaBoost was the most dependable and effective. It achieved the best R^2 scores (0.9978 on training and 0.9919 on testing) and the lowest error values across all metrics. By combining local exploitation (pursuit and attack) with global exploration (prey identification), the NGO's biologically inspired two-phase optimization process successfully traverses the solution space, as evidenced by these results. This dynamic allows for precise adjustment of AdaBoost's parameters, resulting in dependable forecasting even in highly variable and noisy market conditions. In conclusion, the NGO-AdaBoost hybrid model exhibits remarkable predictive accuracy, stability, and efficiency in addition to outperforming conventional optimization-enhanced AdaBoost variants. Because of these characteristics, it is an extremely useful and appropriate tool for strategic decision-making in industries that rely heavily on commodities. Having a distinct AdaBoost model incorporated allows for the separation and measurement of the unique contribution of each optimizer to enhancing model accuracy. AdaBoost, as strong alone, was found to have limitations when hyperparameter tuning was performed, and its relatively lower prediction accuracy on all the metrics—R^2, MAE, MAPE, and RSE—was the outcome. The hybrid models did their work with observable gains, and NGO-AdaBoost achieved the best result, which indicated that blending NGO facilitated a more efficient search of the hyperparameter space. The performance table demonstrates that NGO-AdaBoost outperforms all other hybrid variants in terms of accuracy in both training and testing. FOA's search mechanism, based primarily on sensory flight, lacks adaptive ways to avoid being trapped in local minima and so is not suited for complicated parameter tuning tasks. SSA, with higher competitiveness

compared to FOA, possessed stronger exploration but weaker exploitation. SSA exhibited a lower convergence rate and a more erratic search path, which led to poor tuning of ensemble components in noisy or highly nonlinear environments. Such inconsistency undermined its capacity for exploration-exploitation trade-off to achieve a balance between generalization and fit. Compared to it, the NGO, however, maintained a finer exploration-exploitation trade-off through its utilization of its two-stage (attack and pursuit) approach to both explore potential spots in the solution space and also optimize them in great detail. With this capacity, the NGO-AdaBoost hybrid model became a better generalizer and could perform more consistently as a price-predicting machine for copper prices. In conclusion, the NGO-AdaBoost hybrid model possesses excellent prediction accuracy, stability, and efficiency, as well as outperforms conventional optimization-augmented AdaBoost models. Because of these characteristics, it is an extremely practical and appropriate tool for strategic decision-making in highly commodity-reliant sectors. Additionally, an NGO's simple structure and fewer control parameters, which accelerate convergence and reduce computational complexity, make it more effective for real-time forecasting applications.

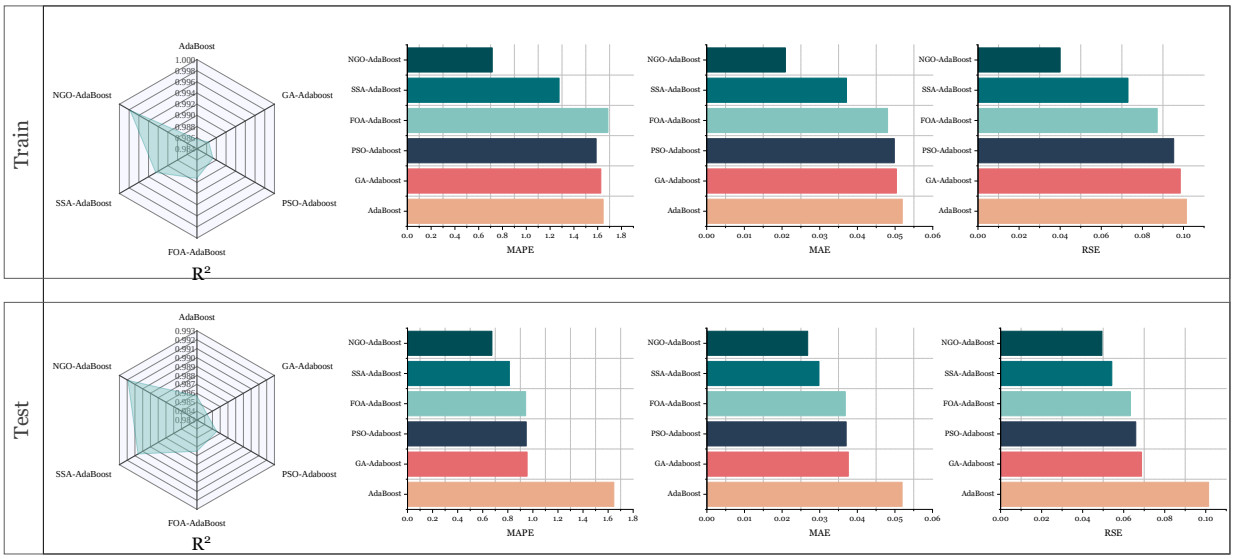


Figure 7: The model evaluation criteria's outcomes.

Figs. 8 and 9 show the actual and predicted value plots of NGO-AdaBoost on the training set and test set, respectively. The plots highlight the capacity of the model to accurately trace trends and ensure stability for learning as well as generalization purposes. The tight correspondence between predicted and actual curves, particularly on unseen test data, highlights the power of the model and its predictive accuracy.

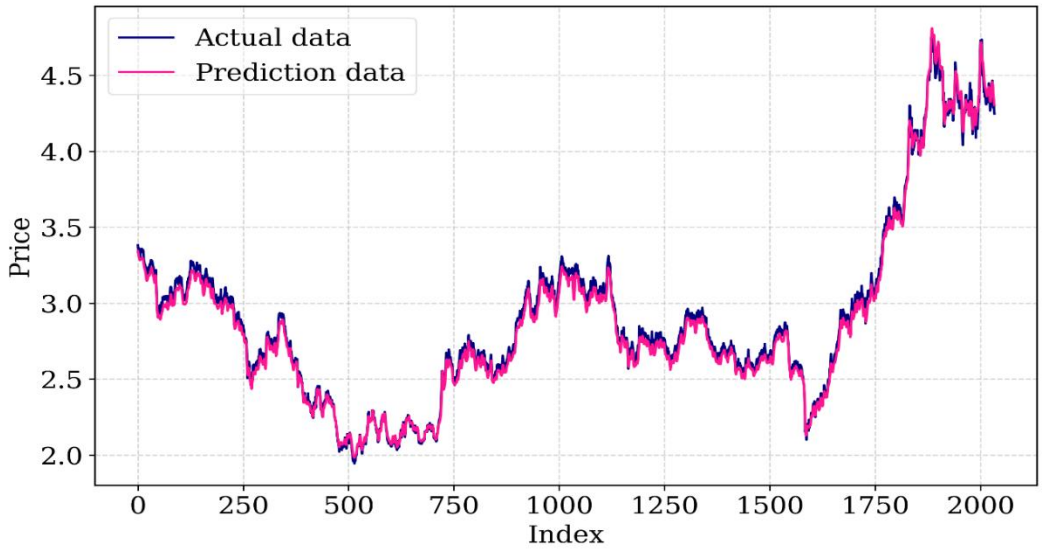


Figure 1: The train data prediction curve for NGO-AdaBoost

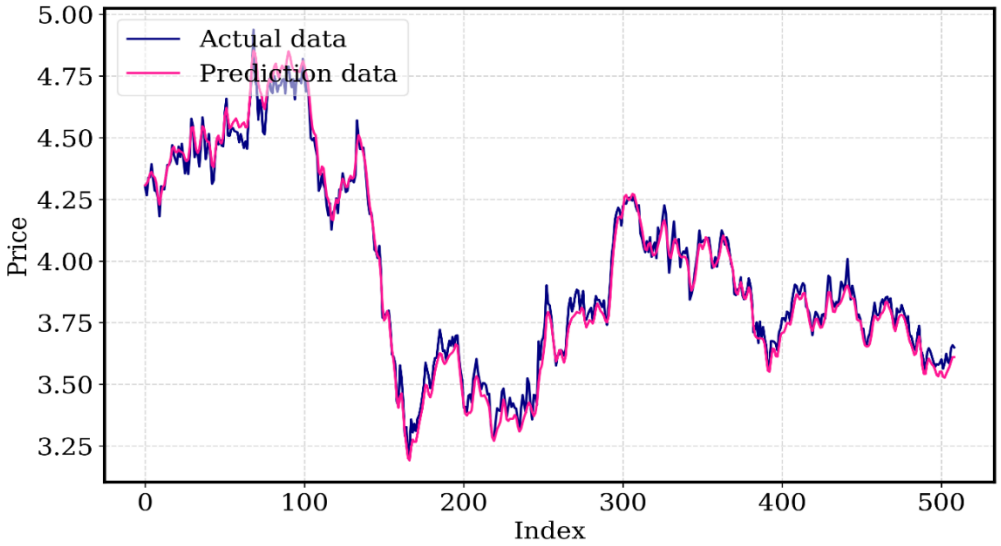


Figure 2: The test data prediction curve for NGO-AdaBoost

This research suggests many schemes such as AdaBoost, FOA-AdaBoost, SSA-AdaBoost, and FOA-AdaBoost to predict future copper prices. The findings reveal that NGO-AdaBoost surpasses other schemes regarding MAPE, RSE, MAE, and R^2 scores. Optimization of the NGO algorithm in connection with AdaBoost seems to be a better approach for copper price prediction. Because of the NGO-AdaBoost model's stability and computational efficiency, the model is well-suited for forecasting, which improves decision-making in sectors of the economy that depend heavily on copper; its precise prediction of copper prices gives investors and policymakers useful information that enables them to make informed decisions in a volatile market; and for copper producers, the model offers a useful tool for scheduling production and setting prices.

The individual models (ARIMA, SVR, GRU, Bi-LSTM, and XGBoost) have a relatively lower complexity as seen in Table 5. For example, models with relatively lower computational costs, like AdaBoost and ARIMA, are faster and more effective during training, but they usually perform worse for complex tasks where volatility and non-linearity are important considerations, like forecasting the price of copper. The AdaBoost model was chosen because of its robust ensemble learning feature, which can greatly enhance prediction quality and resilience when combined with optimization algorithms such as NGO. The hybrid models (GA-AdaBoost, PSO-AdaBoost, FOA-AdaBoost, SSA-AdaBoost, and NGO-AdaBoost) all show increased computational complexity as a result of the optimization step integrated with AdaBoost. Metaheuristic optimizers used to adjust the hyperparameters of the AdaBoost model also increase and improve training time significantly. Particularly, with greater training complexity than less complex models, NGO-AdaBoost achieves improved performance in recognizing complex copper price prediction patterns. Improved accuracy, reduced error rates, and overall NGO-AdaBoost model performance in copper price prediction compensated for the increased complexity of hybrid models. In short, the NGO-AdaBoost model provides a trade-off between precision and computational expense. Though more computationally expensive than conventional models, it is a very potent model for forecasting copper prices because of its superior predictive power, reduced error margins, and enhanced performance on unforeseen data. Its computational expense is justified in terms of the more accurate and stable performance of the model in an extremely volatile market.

Table 5. Training complexity of various models.

Training complexity (FLOPs*)	ARIMA	SVR	GRU	Bi-LSTM	XGBoost	AdaBoost	GA-Adaboost	PSO-Adaboost	FOA-AdaBoost	SSA-AdaBoost	NGO-AdaBoost
	4.03E+11	4.15E+11	4.86E+11	7.67E+11	5.55E+11	4.61E+11	1.87E+12	1.42E+12	2.68E+12	2.63E+12	1.49E+12

6.1 Comparative evaluation of the proposed model with state-of-the-art techniques

To confirm the performance of the proposed AdaBoost model, this paper compared it with five current state-of-the-art prediction methods: Autoregressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR), Gated Recurrent Unit (GRU), Bidirectional Long Short-Term Memory (Bi-LSTM), and Extreme Gradient Boosting (XGBoost). Table 6 and fig 10 show the summarization results using four traditional evaluation metrics: R^2, MAPE, MAE, and RSE, for training and test sets.

Table 6: Comparing the outcomes of the proposed Adaboost model's evaluation criteria with the state-of-the-art methods.

Schemes/Metrics		ARIMA	SVR	GRU	Bi-LSTM	XGBoost	AdaBoost
Train	R^2	0.9399	0.9534	0.9703	0.9766	0.9805	0.9857
	MAPE	4.4475	3.6715	2.7182	2.4089	1.9231	1.6457
	MAE	0.1314	0.0919	0.0778	0.0714	0.0620	0.0519
	RSE	0.2080	0.1825	0.1455	0.1293	0.1185	0.1014
Test	R^2	0.9372	0.9502	0.9675	0.9751	0.9785	0.9836
	MAPE	2.2071	2.0387	1.3553	1.1973	1.2118	0.9655
	MAE	0.0888	0.0767	0.0521	0.0499	0.0466	0.0381
	RSE	0.1375	0.1217	0.0988	0.0866	0.0804	0.0702

It produced the greatest R^2 values (training 0.9857, testing 0.9836), which indicates a superior ability to account for the variance of copper price trends. It produced the lowest MAPE and MAE, high precision, and the lowest variation. The lowest RSE values indicate a more central distribution of errors and a superior fit to actual values. These results validate the fact that AdaBoost is a highly effective and stable model for predicting copper prices, better than traditional statistical models (ARIMA, SVR) and deep learning models (GRU, Bi-LSTM). Based on this strong baseline performance, AdaBoost was selected as the baseline ensemble learner to be further optimized through metaheuristic optimization. Its combination with an NGO at the level of optimization enabled hyperparameter tuning (learning rate, number of estimators), further enhancing generalization performance in especially noisy or nonlinear market conditions.

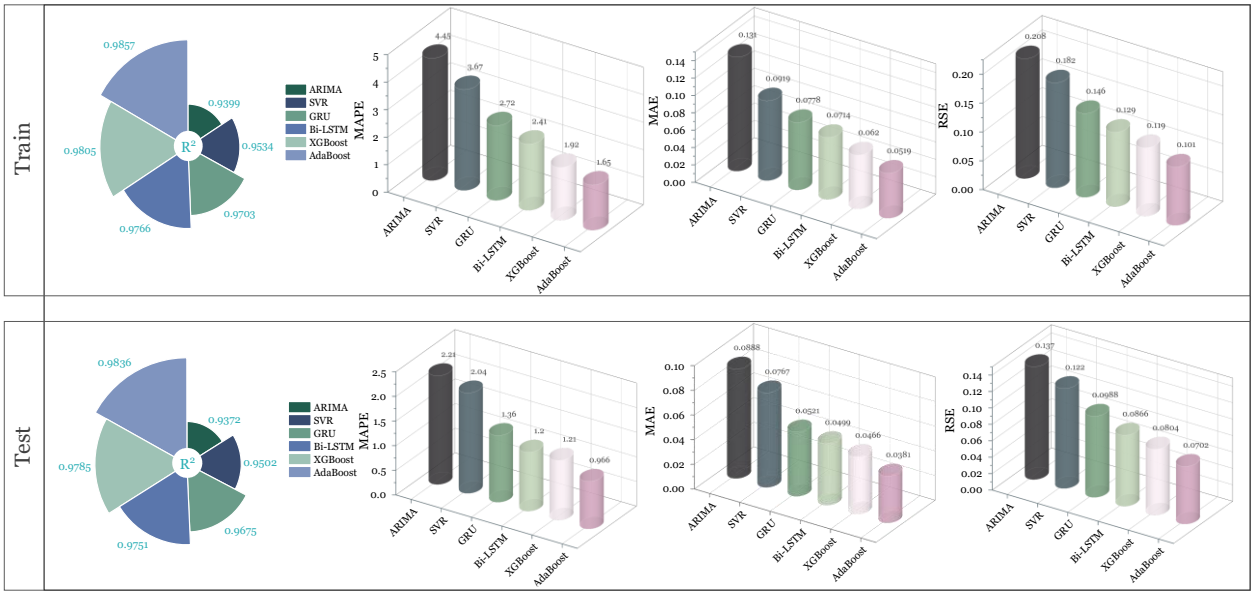


Figure 10: Contrasting the results of the suggested Adaboost model with state-of-the-art techniques.

6.2 Statistical Significance tests

The models' statistical significance was assessed by comparing the NGO-AdaBoost model's performance to that of other benchmark models using the Wilcoxon signed-rank test. Table 7 displays the Wilcoxon test results, test statistics (Wilcoxon values), and corresponding p-values for each model. The p-values indicate the statistical significance, whereas the Wilcoxon values show the performance differences between the paired models. A p-value of less than 0.05 indicates a statistically significant difference in performance. The table shows that the NGO-AdaBoost model performs better than AdaBoost, GA-AdaBoost, PSO-AdaBoost, FOA-AdaBoost, and SSA-AdaBoost in terms of predictive accuracy. Additionally, its p-value (1.36E-67) is the lowest.

Table 7: Wilcoxon Test Statistics and P-values for Model Comparison.

Models	**Wilcoxon**	**P-value**
AdaBoost	18506	2.07E-17
GA-Adaboost	20233	1.95E-13
PSO-Adaboost	19518	1.39E-09
FOA-AdaBoost	31227	2.72E-07
SSA-AdaBoost	2493	1.17E-65
NGO-AdaBoost	1358	1.36E-67

6.3 Comparison with literature reviews

The literature reviews for copper price prediction are compared in Table 8 and Fig.11, where the accuracy of various models and approaches is compared with the model suggested in this study. These findings indicate that the suggested model is significantly has more accuracy for predictions. These findings offer insightful information that helps investors make well-informed investment decisions.

Table 8: Performance comparison between the suggested model in this study and the methods and models offered in literature reviews.

Methods	**Results**	**Year**	**Reference**
RBF	R^2: 0.727	2024	[34]
RF	R^2: 0.608	2024	
SVM	R^2: 0.805	2024	
Deep Extreme Learning Machine (DELM)	R^2 : 0.959	2024	[35]
XGB	R^2: 0.935	2024	
GEP	R^2: 0.947	2024	
SSO-XGB	R^2 : 0.962	2024	
HHO-XGB	R^2: 0.959	2024	
LSTM	R^2 : 0.99012	2023	[19]
ANFIS-ACO	R^2 : 0.956	2021	[36]
NGO-Adaboost	**R^2: 0.9919**		**Proposed model**

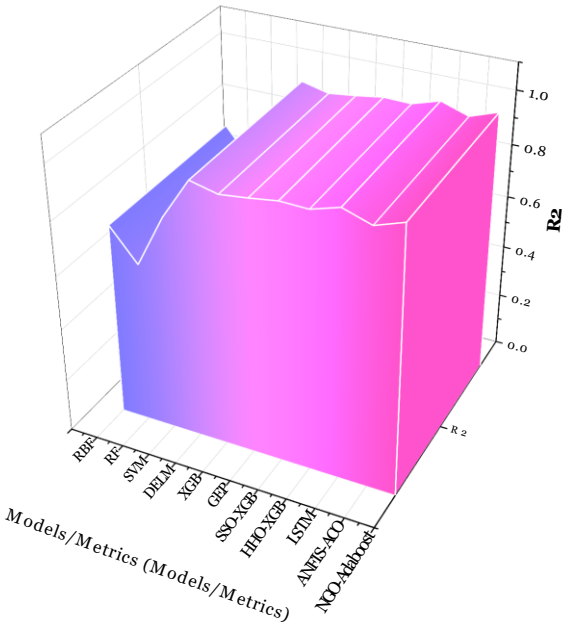


Figure 11: Comparison of the performance of the proposed model with other models presented in the literature reviews for predicting the price of copper.

7 Conclusions

Accurately predicting copper prices in both the spot and futures markets provides significant benefits to several stakeholders. Increased price predictability may allow policymakers to control the market more effectively. Copper producers can increase the efficiency and responsiveness of their production processes. As a result, investors may more effectively develop profitable investment plans for the short, medium, or long term. The purpose of this research is to forecast copper prices using a hybrid forecasting strategy that combines optimal schemes and ensemble learning algorithms. This work aims to fill the lack of research on copper price anticipation by providing a hybrid technique that overcomes the constraints of conventional time series schemes.

The main findings obtained from this investigation are as follows:

To forecast copper prices, this study introduced a new hybrid forecasting model that combines NGO and AdaBoost. This research uses historical data, including OHLC price and volume. The January 2014–October 2023 dataset period is used. The findings demonstrated that the NGO-AdaBoost model outperformed other metaheuristic-based models like PSO and GA-AdaBoost as well as more conventional machine learning models like SVR and ARIMA. Out of all the models tested, the NGO-AdaBoost model produced the best predictions with the highest R2 values (0.9978 training and 0.9919 testing) and the lowest error levels (MAPE: 0.6733, MAE: 0.0268). This resulted from the NGO's ability to successfully modify AdaBoost's hyperparameters to trade off local exploitation and global search through its biologically inspired two-phase search process.

. The outcomes display the robustness of the model in even adverse, noisy economic climates like predicting copper prices. The data clearly shows that the NGO-AdaBoost model works better than any other model. It also produces precise forecasts. Forecasts might be helpful to investors, policymakers, and manufacturers.

Notwithstanding its high predictive accuracy, the NGO-AdaBoost model has a few limitations. The model's sensitivity to AdaBoost and NGO hyperparameter settings may necessitate considerable fine-tuning to attain peak performance. NGOs help automate this process, but improper configurations can still result in less-than-ideal results. As with most ensemble methods, there is also a risk of overfitting, particularly when applied to small or noisy datasets. The model's performance has only been validated using data on copper prices; its applicability to other commodities or financial assets has not yet been established. Ultimately, using metaheuristic optimization increases computational complexity, which may limit scalability for real-time or large-scale applications.

Subsequent studies will build on this research by assessing the performance of various meta-heuristic optimization algorithms, such as Simulated Annealing (SA), Artificial Bee Colony (ABC), and Ant Colony Optimization (ACO), in enhancing AdaBoost for copper price forecasting. The application of external factors like macroeconomic factors (such as inflation rates, interest rates, and commodity prices) and geopolitical events (such as trade wars and policy changes) will also be critical in

enhancing the robustness of the model. Taking these factors into account is most likely going to increase the model's ability to predict due to their crucial role in variations in copper price. Additionally, learning more advanced ensemble methods like Gradient Boosting Machines (GBM) will give a well-rounded understanding of the best techniques for predicting prices. Finally, using techniques like SHAP or LIME, the explainability of NGO-AdaBoost can be improved such that stakeholders can understand the driving factors behind the predictions of the model and have faith in its outcomes in real-world applications.

Acknowledgment

This investigation was backed by a teaching research project of Hubei Polytechnic University: "Exploration and Practice on the Cultivation of the 'Three Creations' Ability of Local College Talents Based on the Industrial College under the Background of New Engineering" (No.2024004).

Authorship contribution statement

Xianming Liu: 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] Dehghani, H (2018). Forecasting copper price using gene expression programming. *Journal of Mining and Environment*, Shahrood University of Technology, 9(2), pp. 349–360. https://doi.org/10.22044/jme.2017.6195.1435.

[2] Dehghani, H. and D. Bogdanovic (2018). Copper price estimation using bat algorithm. *Resources Policy*, Elsevier, 55, pp. 55–61. https://doi.org/10.1016/j.resourpol.2017.10.015.

[3] Lasheras, F.S., F.J. de Cos Juez, A.S. Sánchez, A. Krzemień and P.R. Fernández (2015). Forecasting the COMEX copper spot price by means of neural networks and ARIMA models. *Resources Policy*, Elsevier, 45, pp. 37–43. https://doi.org/10.1016/j.resourpol.2015.03.004.

[4] Liu, C., Z. Hu, Y. Li and S. Liu (2017). Forecasting copper prices by decision tree learning. *Resources Policy*, Elsevier, 52, pp. 427–434. https://doi.org/10.1016/j.resourpol.2017.05.007.

[5] Hu, C., X. Liu, B. Pan, H. Sheng, M. Zhong, X. Zhu and F. Wen (2017). The impact of international price shocks on China's nonferrous metal companies: A case study of copper. *Journal of Cleaner Production*, Elsevier, 168, pp. 254–262. https://doi.org/10.1016/j.jclepro.2017.09.035.

[6] Cheng, C., A. Sa-Ngasoongsong, O. Beyca, T. Le, H. Yang, Z. Kong and S.T.S. Bukkapatnam (2015). Time series forecasting for nonlinear and non-stationary processes: a review and comparative study. *Iie Transactions*, Taylor & Francis, 47(10), pp. 1053–1071. https://doi.org/10.1080/0740817X.2014.999180.

[7] Kriechbaumer, T., A. Angus, D. Parsons and M.R. Casado (2014). An improved wavelet–ARIMA approach for forecasting metal prices. *Resources Policy*, Elsevier, 39, pp. 32–41. https://doi.org/10.1016/j.resourpol.2013.10.005.

[8] García, D. and W. Kristjanpoller (2019). An adaptive forecasting approach for copper price volatility through hybrid and non-hybrid models. *Applied Soft Computing*, Elsevier, 74, pp. 466–478. https://doi.org/10.1016/j.asoc.2018.10.007.

[9] Li, G. and Y. Li (2015). Forecasting copper futures volatility under model uncertainty. *Resources Policy*, Elsevier, 46, pp. 167–176. https://doi.org/10.1016/j.resourpol.2015.09.009.

[10] Chehal, D., P. Gupta and P. Gulati (2023). Predicting the Usefulness of E-Commerce Products' Reviews using Machine Learning Techniques. *Informatica*, Slovenian Society Informatika, 47(2). https://doi.org/10.31449/inf.v47i2.4155.

[11] Zhang, T. and X. Yang (2024). Energy-Saving Design of Smart City Buildings Based on Deep Learning Algorithms and Remote Sensing Image Scenes. *Informatica*, Slovenian Society Informatika, 48(19). https://doi.org/10.31449/inf.v48i19.6049.

[12] Jelenčič, J. and D. Mladenić (2022). Improving modeling of stochastic processes by smart denoising. *Informatica*, Slovenian Society Informatika, 46(1). https://doi.org/10.31449/inf.v46i1.3875.

[13] Barstuğan, M. and R. Ceylan (2020). The effect of dictionary learning on weight update of AdaBoost and ECG classification. *Journal of King Saud University-Computer and Information Sciences*, Elsevier, 32(10), pp. 1149–1157. https://doi.org/10.1016/j.jksuci.2018.11.007.

[14] Ferguson-Lees, J (2001). *Raptors of the World*. Houghton Mifflin Company.

[15] El-Dabah, M.A., R.A. El-Sehiemy, H.M. Hasanien and B. Saad (2023). Photovoltaic model parameters identification using Northern Goshawk Optimization algorithm. *Energy*, Elsevier, 262, pp. 125522. https://doi.org/10.1016/j.energy.2022.125522.

[16] García-Gonzalo, E., P.J. García Nieto, J. Gracia Rodríguez, F. Sánchez Lasheras and G. Fidalgo Valverde (2023). A support vector regression model for time series forecasting of the COMEX copper spot price. *Logic Journal of the IGPL*, Oxford Academic, 31(4), pp. 775–784. https://doi.org/10.1093/jigpal/jzac039.

[17] Chen, J., J. Yi, K. Liu, J. Cheng, Y. Feng and C. Fang (2023). Copper price prediction using LSTM recurrent neural network integrated simulated annealing algorithm. *Plos One*, Public Library of Science (PLOS), 18(10), e0285631. https://doi.org/10.1371/journal.pone.0285631.

[18] Astudillo, G., R. Carrasco, C. Fernández-Campusano and M. Chacón (2020). Copper price prediction using support vector regression technique. *Applied Sciences*, MDPI, 10(19), p. 6648. https://doi.org/10.3390/app10196648.

[19] Zhao, Y., J. Chen, H. Shimada and T. Sasaoka (2023). Non-Ferrous Metal Price Point and Interval Prediction Based on Variational Mode Decomposition and Optimized LSTM Network. *Mathematics*, MDPI, 11(12), p. 2738. https://doi.org/10.3390/math11122738.

[20] Luo, H., D. Wang, J. Cheng and Q. Wu (2022). Multi-step-ahead copper price forecasting using a two-phase architecture based on an improved LSTM with novel input strategy and error correction. *Resources Policy*, Elsevier, 79, p. 102962. https://doi.org/10.1016/j.resourpol.2022.102962.

[21] Khoshalan, H.A., J. Shakeri, I. Najmoddini and M. Asadizadeh (2021). Forecasting copper price by application of robust artificial intelligence techniques. *Resources Policy*, Elsevier, 73, p. 102239. https://doi.org/10.1016/j.resourpol.2021.102239.

[22] Wang, L., Y. Guo, M. Fan and X. Li (2022). Wind speed prediction using measurements from neighboring locations and combining the extreme learning machine and the AdaBoost algorithm. *Energy Reports*, Elsevier, 8, pp. 1508–1518. https://doi.org/10.1016/j.egyr.2021.12.062.

[23] Obiedat, R. and S.A. Toubasi (2022). A combined approach for predicting employees' productivity based on ensemble machine learning methods. *Informatica*, Slovenian Society Informatika, 46(5). https://doi.org/10.31449/inf.v46i5.3839.

[24] Lu, J., H. Hu and Y. Bai (2015). Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and AdaBoost algorithm. *Neurocomputing*, Elsevier, 152, pp. 305–315. https://doi.org/10.1016/j.neucom.2014.10.065.

[25] Peng, T., J. Zhou, C. Zhang and Y. Zheng (2017). Multi-step ahead wind speed forecasting using a hybrid model based on two-stage decomposition technique and AdaBoost-extreme learning machine. *Energy Conversion and Management*, Elsevier, 153, pp. 589–602. https://doi.org/10.1016/j.enconman.2017.10.021.

[26] Wu, X. and S. Meng (2016). E-commerce customer churn prediction based on improved SMOTE and AdaBoost, In *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, IEEE, Kunming, pp: 1–5. https://doi.org/10.1109/ICSSSM.2016.7538581.

[27] Zhao, H., H. Yu, D. Li, T. Mao and H. Zhu (2019). Vehicle accident risk prediction based on AdaBoost-SO in VANETs. *IEEE Access*, IEEE, 7, pp. 14549–14557. https://doi.org/10.1109/ACCESS.2019.2894176.

[28] Sutradhar, S., S. Karforma, R. Bose and S. Roy (2023). A dynamic step-wise tiny encryption algorithm with fruit fly optimization for quality of service improvement in healthcare. *Healthcare Analytics*, Elsevier, 3, p. 100177. https://doi.org/10.1016/j.health.2023.100177.

[29] Xue, J. and B. Shen (2020). A novel swarm intelligence optimization approach: sparrow search algorithm. *Systems Science & Control Engineering*, Taylor & Francis, 8(1), 22–34. https://doi.org/10.1080/21642583.2019.1708830.

[30] Mitchell, M (1998). *An introduction to genetic algorithms*. MIT press.

[31] Alkafaween, E., A.B.A. Hassanat and S. Tarawneh (2021). Improving initial population for genetic algorithm using the multi linear regression based technique (MLRBT). *Communications-Scientific Letters of the University of Zilina*, Žilinská univerzita v Žilině, 23(1), E1–E10.

https://www.ceeol.com/search/article-detail?id=1120555.

[32] Kennedy, J. and R. Eberhart (1995). Particle swarm optimization, In *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, Perth, WA, Australia, pp: 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

[33] Sadeeq, H.T. and A.M. Abdulazeez (2022). Improved northern Goshawk optimization algorithm for global optimization, In *2022 4th International Conference on Advanced Science and Engineering (ICOASE)*, IEEE, Zakho, Iraq, pp: 89–94. https://doi.org/10.1109/ICOASE56293.2022.10075576.

[34] Li, N., J. Li, Q. Wang, D. Yan, L. Wang and M. Jia (2024). A novel copper price forecasting ensemble method using adversarial interpretive structural model and sparrow search algorithm. *Resources Policy*, Elsevier, 91, pp. 104892. https://doi.org/10.1016/j.resourpol.2024.104892.

[35] Nabavi, Z., M. Mirzehi and H. Dehghani (2024). Reliable novel hybrid extreme gradient boosting for forecasting copper prices using meta-heuristic algorithms: A thirty-year analysis. *Resources Policy*, Elsevier, 90, p. 104784. https://doi.org/10.1016/j.resourpol.2024.104784.

[36] Khoshalan, H.A., J. Shakeri, I. Najmoddini and M. Asadizadeh (2021). Forecasting copper price by application of robust artificial intelligence techniques. *Resources Policy*, Elsevier, 73, pp. 102239. https://doi.org/10.1016/j.resourpol.2021.102239.

DQN-Raft+: A Deep Reinforcement Learning-Optimized Lightweight Consensus Algorithm for Secure Edge Storage in IoT Environments

Pengyuan Wang, Qiaonian Xu*, Huaiqing Zhang
School of Wuwei Vocation College School of Information Technology, Wuwei Gansu, 733000, China
E-mail: wwocxqn66@163.com
*Corresponding author

Keywords: internet of things, digitization, information security, privacy protection, deep learning

Received: April 14, 2025

The rapid development of the Internet of Things (IoT) has intensified security and privacy challenges across data generation, transmission, and storage. This study introduces a blockchain-based secure edge storage model tailored for IoT environments and presents a lightweight consensus algorithm, Deep Q-Network (DQN)-Raft+, which incorporates deep reinforcement learning. By combining the decentralized features of edge computing and blockchain, the model enables automated data access control through smart contracts. Furthermore, it optimizes leader node selection in the Raft consensus process using a DQN, formulating the consensus as a Markov Decision Process to enhance responsiveness and privacy protection in dynamic network conditions. Experiments were performed in a simulated environment using TensorFlow 2.6 and a MySQL database. The performance of DQN-Raft+ was compared against traditional consensus algorithms, including Proof of Work, Proof of Stake, Practical Byzantine Fault Tolerance, and Delegated Byzantine Fault Tolerance. Results indicate that DQN-Raft+ significantly reduces block generation delay (175.77 ms) and achieves a high privacy protection score (0.95). It also maintains a low data loss rate of 0.01%, demonstrating enhanced robustness and real-time capability. These findings indicate that DQN-Raft+ effectively strengthens data security and privacy in IoT systems, offering a technically sound and efficient mechanism for secure data exchange. The study provides both a theoretical framework and practical direction for future research in secure IoT deployment.

Povzetek: Študija uvaja DQN-Raft+, lahkoten konsenzni algoritem, optimiziran z globokim učenjem za izboljšano varnost podatkov in zaščito zasebnosti v IoT omrežjih z izboljšanjem latence in robustnosti.

1 Introduction

The rapid advancement of Internet of Things (IoT) technology has led to the widespread adoption of smart devices, increasing the frequency of digital information generation, transmission, and storage [1]. Concurrently, concerns regarding secure storage and privacy protection have become increasingly critical. According to Statista, the number of connected IoT devices worldwide is projected to reach billions in the near future. While this expansion enhances data accessibility, it also elevates the risk of breaches, tampering, and other security threats [2]. Traditional data storage solutions often fail to meet the stringent requirements for confidentiality and integrity in IoT environments [3]. In response, blockchain technology has emerged as a promising alternative due to its decentralized, tamper-resistant, and transparent nature. Numerous studies have examined blockchain's application in data protection, noting its potential to mitigate hacking attempts and unauthorized data alterations [4, 5]. Nonetheless, blockchain alone does not fully address privacy concerns. In complex IoT environments, ensuring data security while safeguarding user privacy remains a formidable challenge [6, 7]. Current privacy protection mechanisms are frequently limited by performance constraints. As data volume increases, computational overhead and response time often become significant barriers to efficiency [8, 9].

This study addresses the challenges of high latency and inefficient privacy protection in IoT environments. It investigates the following research question: In IoT scenarios with frequent terminal node state fluctuations, can a Deep Q-Network (DQN) be used to optimize the Raft consensus algorithm to reduce block generation latency and enhance the efficiency of data access control and privacy protection in edge blockchain systems? To explore this question, a blockchain-based secure storage model is deployed at the network edge. A lightweight consensus mechanism, DQN-Raft+, is developed by integrating deep reinforcement learning into the Raft protocol. The proposed approach is evaluated in terms of security, efficiency, and adaptability, demonstrating improved performance under dynamic and resource-constrained IoT conditions. This study introduces a secure digital information storage model for blockchain-based IoT environments and designs a lightweight consensus algorithm incorporating deep learning (DL) to enhance data privacy protection. The blockchain model improves data confidentiality and integrity, addressing the limitations of traditional storage methods in IoT

systems. The lightweight consensus algorithm reduces computational overhead and boosts data processing efficiency, supporting the real-time handling of large-scale IoT data. This approach not only strengthens privacy protection but also enhances system performance, offering a novel framework and technical direction for secure storage and privacy management in IoT applications.

The remaining sections of this study are structured as follows. Section 2 reviews related work on secure information storage and privacy protection in IoT environments, with a focus on the strengths and limitations of existing consensus algorithms and DL-based privacy-preserving techniques. Section 3 describes the architecture of the proposed secure storage system, which integrates blockchain and edge computing. This section covers the data acquisition mechanism, blockchain framework, security model design, and the implementation of the DQN-Raft+ consensus algorithm. Section 4 presents the experimental setup and performance evaluation, including platform construction, data collection methods, and evaluation metrics. A comparative analysis is conducted across several dimensions-latency, privacy protection, and access control-highlighting differences between DQN-Raft+ and baseline algorithms in terms of system complexity, learning convergence, and data security. Section 5 concludes the study by summarizing key findings, discussing current limitations, and suggesting directions for future research.

2 Related work

The rapid advancement of information technology has heightened concerns regarding the secure storage and privacy protection of digital information, particularly within cloud computing and IoT environments. Khan and Por pointed out that the rise of information technology has exacerbated the challenges of secure storage, particularly in cloud computing and IoT environments, where the risks of data leakage and tampering have grown significantly [10]. To address these challenges, Pazhani et al. proposed an improved memory-efficient distributed algorithm architecture for implementing two-dimensional discrete wavelet transform, to mitigate the high computational cost associated with image and video compression. In portable devices and high-speed communication systems, conventional multiplier-based computation is no longer feasible due to constraints on chip area and power consumption. The distributed algorithm architecture replaces multipliers with shift operations and lookup tables, thereby improving computational speed and reducing power usage. However, the increase in filter coefficients often leads to significant expansion of the lookup table size, which remains a concern [11]. In parallel, Dwivedi et al. highlighted the necessity of encrypted data storage in edge computing environments to reduce security risks during data transmission [12]. Cao et al. observed that current privacy protection mechanisms primarily rely on encryption, access control, and anonymization. However, these approaches often struggle with high-dimensional and dynamic data, especially in IoT contexts, where the growing number of devices and data volume leads to reduced efficiency and reliability [13]. Wen et al. highlighted the ongoing challenge of balancing privacy protection with data usability in the process of data sharing and utilization [14]. In response, blockchain has gained attention for its decentralized, tamper-resistant, and transparent properties. Chen et al. noted its widespread application in the secure storage of digital information [15]. Ren et al. introduced a blockchain-based solution that uses smart contracts to automate data management and access control [16]. Furthermore, Emami et al. explored the role of consortium blockchains in enabling secure data sharing among multiple participants while protecting privacy [17]. In the domain of DL, Rodríguez et al. proposed its application in privacy protection by learning user behavior patterns. Their study suggested that DL-based mechanisms could enhance data analysis accuracy while reducing reliance on raw data [18]. Valencia-Arias et al. further investigated the integration of DL and blockchain, indicating that this combination improves security and flexibility in both privacy protection and data sharing [19].

Table 1 presents a structured comparison of representative approaches. This table illustrates the advantages and limitations of existing work-especially in relation to consensus algorithms and DL-based privacy protection mechanisms.

Table 1: Comparison of consensus algorithms and DL-based privacy protection approaches

Method Type	Representative Model	Main Advantages	Main Limitations	DQN Integration Potential
Consensus Algorithm	Raft	Simple design; low communication overhead	Poor fault tolerance; lacks adaptability in dynamic settings	Compatible
	Practical Byzantine Fault Tolerance (PBFT)	Strong consistency; tolerates Byzantine faults	High communication complexity; poor scalability	Difficult
	Delegated Byzantine Fault	Better performance than PBFT	Complex architecture; sensitive to	Theoretically

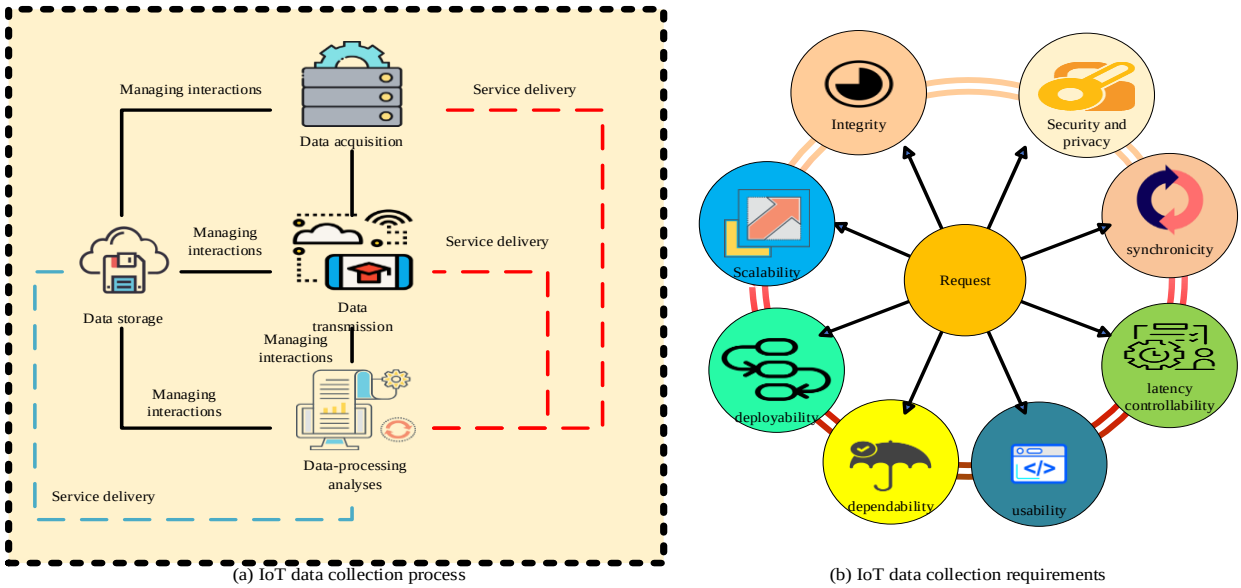
	Tolerance (DBFT)		node expansion	feasible
	Proof of Work (PoW)	High security; widely used	High energy usage; high latency	Not suitable
	Proof of Stake (PoS)	Energy-efficient; faster block generation than PoW	Lower decentralization; risk of centralization	Not suitable
DL Privacy Methods	CNN-based privacy model	Effective at capturing spatial features	Rigid architecture; limited generalization across diverse data	Partially compatible
	Recurrent Neural Network + Access Control	Suitable for modeling temporal privacy patterns	Poor scalability in real-time IoT environments	Compatible
	Federated Learning	Data remains local; strong privacy guarantees	High communication overhead; slow convergence	Can be combined with DQN
	DQN-based privacy scoring	Adaptive learning; responsive in dynamic settings	Complex modeling; high training cost	Core of this study

In summary, while substantial progress has been made in secure storage and privacy protection for digital information, key challenges persist. Most existing consensus algorithms lack the flexibility to handle dynamic IoT environments, and DL techniques, though effective in learning complex patterns, have not been fully integrated into consensus mechanisms. In particular, the application of DQNs in consensus optimization remains limited. Existing research often suffers from vague state space definitions and unstable training processes. To address these gaps, the DQN-Raft+ algorithm proposed in this study constructs a lightweight and self-adaptive consensus model, enhancing both privacy protection efficiency and system robustness in IoT environments.

3 The Resource Sharing System for Vocational Education Based on Blockchain and edge computing

3.1 IoT

IoT, a core technology linking the physical and digital worlds, is characterized by high-frequency, multi-source, and real-time data collection. These attributes create favorable conditions for integrating edge intelligence and blockchain technologies to address the demands of secure, efficient, and privacy-aware data storage [20]. To meet these requirements, this study develops a secure digital information storage model based on a collaborative architecture that combines blockchain with edge computing. Before detailing the model design, it is necessary to outline the typical data acquisition process in IoT environments. As shown in Figure 1, this process forms the theoretical and technical foundation of the proposed architecture [21].



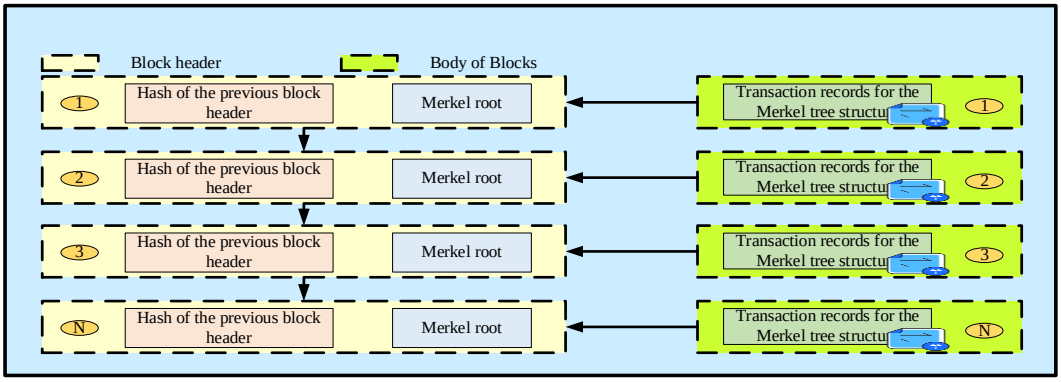
(a) IoT data collection process (b) IoT data collection requirements

Figure 1: Data acquisition and processing flow in an IoT environment

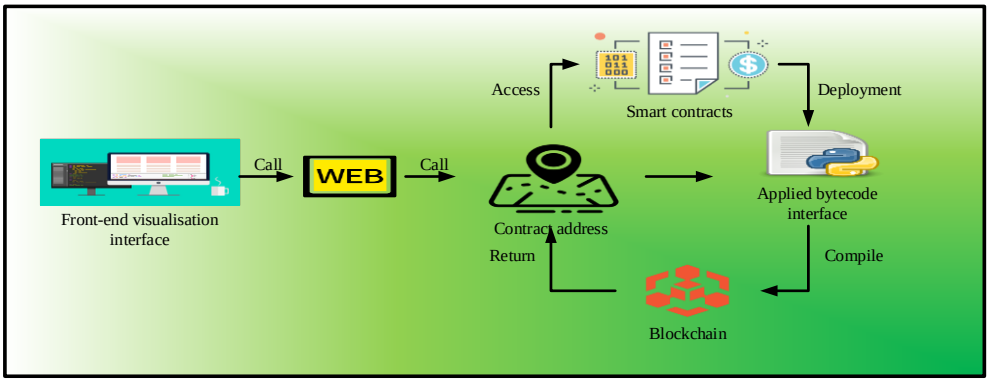
Figure 1 illustrates a standard data flow in IoT systems, including stages such as data sensing by edge devices, preprocessing, transmission, and storage. This workflow provides a reference framework for designing the blockchain–edge computing-based secure storage model.

3.2 Blockchain technology

Blockchain is a decentralized, distributed ledger technology that records and stores data securely and transparently. It organizes data into linked "blocks," with each block containing the hash of the preceding block, thereby forming an immutable chain. The core features of blockchain-transparency, immutability, and security-have enabled its widespread application in areas such as financial transactions, supply chain management, and identity authentication. Furthermore, blockchain supports the deployment of smart contracts, which automate transactions without intermediaries, reducing operational costs and improving efficiency [22]. The fundamental structure of blockchain and the design principles of smart contracts are presented in Figure 2 [23].



(a) Basic structure of blockchain technology



(b) Principles of smart contract development

Figure 2: Basic structure of blockchain and development principles of smart contracts

Cryptography serves as the cornerstone of blockchain security and privacy. Hash functions are used to maintain data integrity and prevent tampering, while public–private key cryptography supports identity authentication and transaction signing. Digital signatures ensure transaction authenticity, and both symmetric and asymmetric encryption schemes safeguard data confidentiality. Additionally, zero-knowledge proofs allow users to validate claims without exposing sensitive data. The integration of these cryptographic techniques significantly enhances the security and trustworthiness of blockchain-based data storage and sharing systems [24]. Key cryptographic mechanisms are shown in Figure 3 [25].

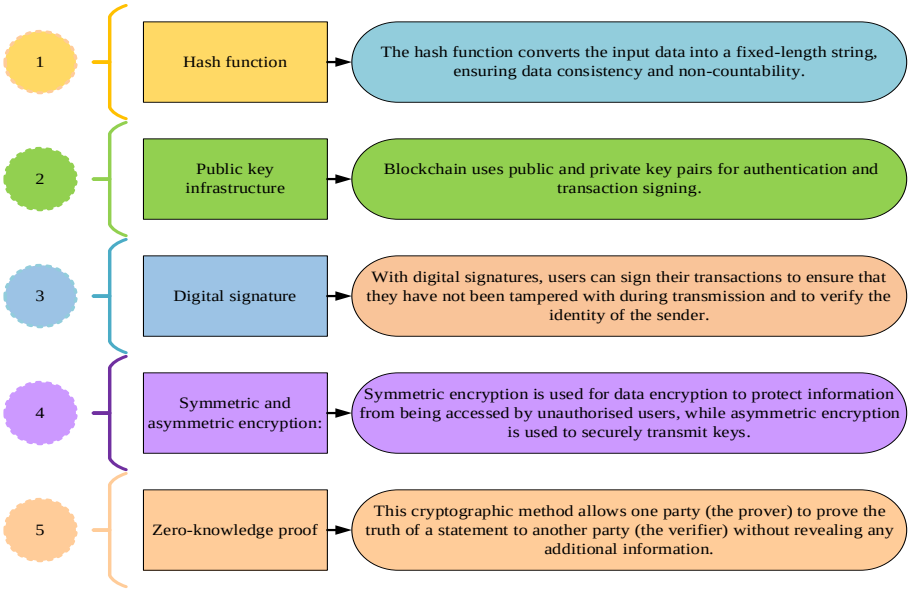


Figure 3: Cryptographic techniques

3.3 Digital information security storage model for network edge-based blockchain IoT

This study proposes a digital information security storage model for blockchain-enabled IoT environments based on network edge computing. The model integrates the decentralized architecture of blockchain with the low-latency, high-responsiveness advantages of edge computing to ensure data security, availability, and privacy protection. The model architecture comprises three primary components: edge devices, edge computing nodes, and a blockchain network connected to cloud servers. At the edge device level-such as sensors, smart home systems, and industrial controllers-data is generated and preprocessed locally. This reduces latency and bandwidth usage by limiting the volume of raw data transmitted over the network. Data is then securely sent to edge computing nodes via encrypted communication protocols. Edge nodes play a key role in processing the data, including encryption, deduplication, and compression, to enhance both transmission efficiency and security. The processed data is subsequently uploaded to the blockchain for decentralized storage. Blockchain ensures data traceability and immutability, effectively preventing tampering and forgery while preserving data integrity. Each data block embeds corresponding access control policies to manage permissions precisely during access operations. To strengthen privacy protection, the model incorporates encryption techniques and multi-factor authentication. Authorized access is enforced through smart contracts, which automate access control, policy execution, and verification. These contracts not only streamline the access process but also create a secure environment for data sharing among heterogeneous devices and users. Moreover, the model adopts a modular architecture, allowing for dynamic scalability and adaptability. As the number of IoT devices and data volume continues to grow, the system can flexibly reallocate storage and computational resources to accommodate varying demands. Overall, this network edge-based blockchain IoT model significantly enhances digital information security while improving system performance and user experience. By implementing this framework, users can securely store, manage, and share data in a highly responsive and privacy-preserving environment, thereby supporting the sustainable development and application of IoT technologies. The structural design of the proposed model is illustrated in Figure 4.

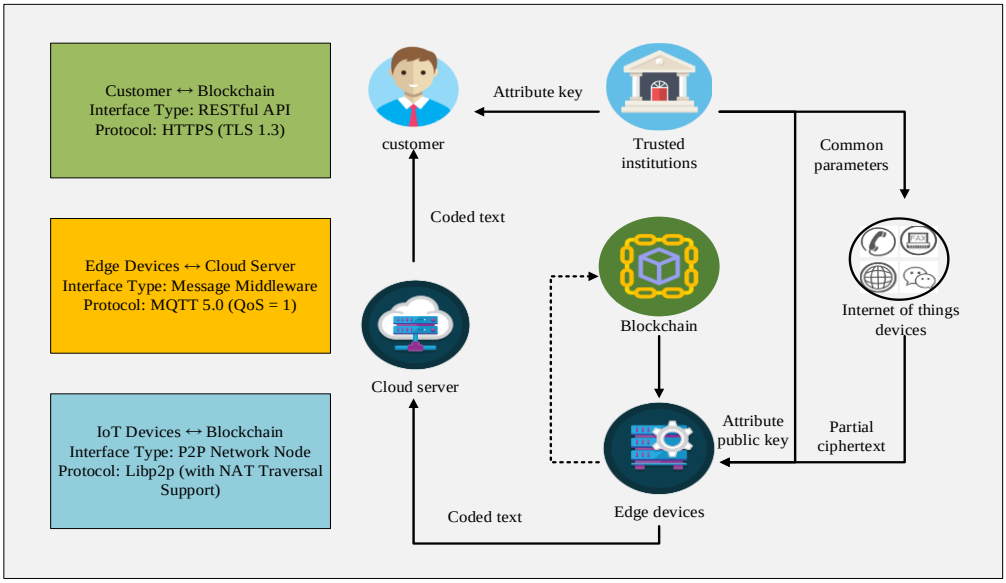


Figure 4: Structure of the blockchain-based digital information security storage model at the network edge

As shown in Figure 4, the blockchain-based information security storage model deployed at the network edge clearly defines the interfaces and protocols among its components through technical annotations. Clients interact with the blockchain via Representational State Transfer Application Programming Interface over Hypertext Transfer Protocol Secure and Transport Layer Security 1.3. Data requests are submitted using attribute-based encryption, and access rights are verified accordingly. Edge devices and cloud servers communicate via the Message Queuing Telemetry Transport 5.0 protocol to transmit fragmented and encrypted data. Reed-Solomon coding is used to improve transmission efficiency under low-bandwidth conditions. IoT devices act as lightweight blockchain nodes and participate in a lightweight PBFT consensus mechanism through the Libp2p-based peer-to-peer networking protocol, enabling block header synchronization and transaction validation. The model integrates several key security mechanisms, including trusted execution environments for key protection, zero-knowledge proofs to conceal transaction attributes, and randomized sampling verification to reduce the computational burden on edge nodes. Through layered encryption, protocol optimization, and lightweight consensus design, the model establishes a secure and efficient data storage and verification pipeline from edge to cloud.

On edge devices, data is encrypted using symmetric encryption algorithms. The ciphertext is computed as Equation (1):

$$C = E(K, D) \tag{1}$$

In Equation (1), C represents the encrypted ciphertext; D denotes the original data; E means the encryption function; K refers to the encryption key. To ensure data integrity and immutability, the hash value H and digital signature S of the data are generated as follows:

$$H = H(D) \tag{2}$$

$$S = Sign(K_{priv}, H) \tag{3}$$

Here, K_{priv} private key of the edge device used for digital signing. Each block B_i on the blockchain contains the following information:

$$B_i = (C, H, S, T, A) \tag{4}$$

In Equation (4), T is a timestamp indicating when the data is generated or stored, and A an access control policy specifying which users or devices can access the data. The access control policy A is represented as a set of conditions, where each condition C_i corresponds to a specific access rule. This can be expressed as Equation (5):

$$A = \{C_1, C_2, \ldots, C_m\} \tag{5}$$

Smart contracts are used to automate data access management based on these conditions. The access control logic is defined by the function:

$$F(U, A) \rightarrow Access_{result} \tag{6}$$

F denotes the access control function of the smart contract, U signifies the user requesting access, and $Access_{result}$ indicates whether the user is granted access to the data. When a user requests data, the access rights are verified by invoking the smart contract, which can be expressed as Equation (7):

$$R = F(U, A) \text{if} R = \text{true} \rightarrow D = D_{retrieved} \tag{7}$$

In Equation (7), R represents the result of the permission check, and $D_{retrieved}$ denotes the raw data retrieved from the blockchain if access is granted.

To further clarify the integration of encryption, access control mechanisms, and the blockchain transaction process, this study designs the data on-chaining procedure to comprise five sequential stages: preprocessing, transaction generation, smart contract triggering, data on-chaining, and access verification. After data is encrypted by edge devices-as outlined in Equations (1) to (3)-and its hash and digital signature are generated, the edge computing nodes encapsulate the encrypted content into a transaction and broadcast it across the blockchain network. Consensus nodes subsequently verify the digital signature and access policy, and upon successful validation, the transaction is committed to the blockchain via the underlying consensus mechanism. Each transaction contains the fields cipher_data, hash_val, signature, timestamp, and access_policy, corresponding to the data block structure defined in Equation (4). This ensures the integrity of the data and the synchronous on-chaining of the associated access control policy. For access control enforcement, this study implements smart contracts using the Solidity programming language, deploying them on a private blockchain based on the Ethereum architecture. Ethereum is selected due to its mature smart contract ecosystem and strong support for complex access control logic. The access control smart contract is deployed on-chain and includes a core function, checkPermission(address user, bytes32 data_id), which evaluates a user's address U against the predefined access policy A, returning a Boolean value that determines whether access is permitted. Access verification is executed through an on-chain contract call. Smart contracts are automatically triggered upon a user's data request, eliminating the need for manual intervention or intermediary middleware. This mechanism facilitates autonomous enforcement of access control directly within the blockchain environment. The proposed framework achieves seamless integration of encryption, permission management, and blockchain-based auditability. Furthermore, it establishes a practical foundation for the future integration of privacy-enhancing technologies and secure multi-party data sharing in IoT environments.

3.4 Design of lightweight consensus algorithms based on DL

DL, a subfield of machine learning, employs multi-layer neural networks to extract complex features from large-scale data and automate decision-making in intricate environments [26]. To enhance the efficiency and privacy-preserving capabilities of blockchain applications in IoT environments, this study proposes a lightweight consensus algorithm named DQN-Raft+, which integrates deep reinforcement learning with the classical Raft consensus protocol. Built upon the Raft algorithm, DQN-Raft+ introduces a policy-learning agent that utilizes a DQN to dynamically optimize the leader election process. In contrast to centralized scheduling strategies, DQN-Raft+ deploys lightweight proxy models in parallel across multiple candidate nodes. Each node independently determines whether to participate in the election based on its locally observed state, including metrics such as resource availability, communication delay, and recent election outcomes. This decentralized decision-making framework enhances adaptability to network fluctuations and improves fault tolerance, while maintaining the underlying distributed nature of the blockchain system. Through this approach, DQN-Raft+ significantly improves the efficiency, scalability, and robustness of the consensus process in highly dynamic IoT environments, addressing key limitations of conventional consensus mechanisms such as fixed leader selection and lack of real-time responsiveness [27, 28].

This study first establishes a terminal error model and adopts a Markov Decision Process (MDP) to represent the system's state transitions. Among various reinforcement learning methods, the DQN is selected for this application based on several key considerations. Although policy gradient algorithms such as Proximal Policy Optimization and Actor-Critic approaches offer advantages in handling continuous action spaces and ensuring stable policy updates, they generally exhibit slower convergence and require high sample efficiency, especially in high-dimensional state spaces. In contrast, DQN, as a value-based method, is particularly effective in environments with discrete action spaces, such as leader node selection in this study. Its relative simplicity, faster convergence, and training stability make DQN well-suited for deployment in resource-constrained IoT edge environments. In the MDP formulation, the state space S is defined to include system metrics such as the current leader node's hardware failure rate, communication latency, terminal response rate, and node load status. The action space Acomprises the set of candidate leader nodes. To direct the learning agent toward optimizing both privacy preservation and system performance, the reward function R is structured as follows: a positive reward (+1) is issued when consensus is achieved and latency remains below a predefined threshold, while a negative reward (–1) is assigned when consensus fails or the latency exceeds the threshold. Additionally, the reward is positively correlated with the security score of the elected leader node and the effectiveness of privacy enforcement. To further stabilize convergence-particularly during the later stages of training-this study replaces the traditional ε-greedy exploration strategy with a combination of entropy-regularized exploration and the Upper Confidence Bound approach. This hybrid strategy enhances the exploration–exploitation balance and improves the agent's generalization ability. Coupled with experience replay and target network updates, the proposed DQN framework facilitates stable policy learning and effectively supports performance optimization in dynamic IoT edge computing environments.

A standard MDP is formally defined as a quadruple (S, A, P, R), where S denotes the set of all possible states in the environment, A represents the set of all actions, P specifies the transition probability between states upon taking a given action, and R is the reward function, which quantifies the immediate gain associated with performing a specific action in a given state [29, 30]. In this study, an agent-modeled as a base station-interacts with the environment to select actions that maximize cumulative rewards. The state space is constructed using terminal device information collected by the base station, along with relevant blockchain network parameters. The action space is defined as the set of possible leading terminal selections within the local blockchain network. To improve the efficiency of data aggregation and dissemination during the block consensus process, the proposed Raft+ algorithm designates a central node (the base station) to collect data from peer nodes and elect a leading terminal. If the selected terminal fails to receive a sufficient number of valid responses within the predefined consensus time, the block is considered invalid, triggering a new round of leader selection. The lightweight consensus algorithm, enhanced with DQN, solves the MDP by continuously interacting with the environment and updating its policy to converge towards an optimal decision strategy that maximizes the expected cumulative reward. In this context, DQN algorithm utilizes deep neural networks to approximate the action-value function $Q(s, a)$, effectively overcoming the limitations of traditional Q-learning in high-dimensional state and action spaces. The algorithm incorporates an experience replay mechanism to eliminate correlations among sequential training samples, ensuring stable and efficient learning. By learning the Q-values, the agent evaluates the expected utility of actions in each state, enabling more effective and informed decision-making. The model operates in a finite and discrete state space, and the action space size corresponds to the number of robust (i.e., strong and reliable) terminals in the network. The DQN-Raft+ algorithm is further refined through targeted training procedures. In each training iteration, the system updates the hardware failure probabilities of each terminal based on a preset random seed queue and simulates potential communication failures. When such failures occur, a new leading terminal is selected according to the ε-greedy policy, which balances exploration and exploitation, helping to avoid convergence to local optima. Through continuous iterative training, the DQN agent converges to an optimal action-value function and learns a corresponding optimal policy, thereby establishing a lightweight and efficient consensus mechanism. This approach not only enhances consensus reliability and scalability but also strengthens the system's privacy protection capabilities in dynamic IoT environments.

For model deployment, the policy training of the DQN-Raft+ algorithm is conducted during a dedicated simulation phase. In this stage, a simulation environment is designed to emulate diverse node states and network feedback conditions, thereby enabling comprehensive training of the Q-network. Upon completion of training, the optimized policy model is embedded into each edge node within the blockchain system. During actual operations, at the onset of each leader election round, nodes autonomously execute the pretrained policy model to evaluate their local state and calculate the action value, based on which they determine whether to participate in the election. This decentralized decision-making mechanism reduces communication overhead and minimizes leader candidate conflicts, thereby significantly enhancing the efficiency and scalability of the leader selection process. The training workflow and model interaction process are illustrated in Figure 5.

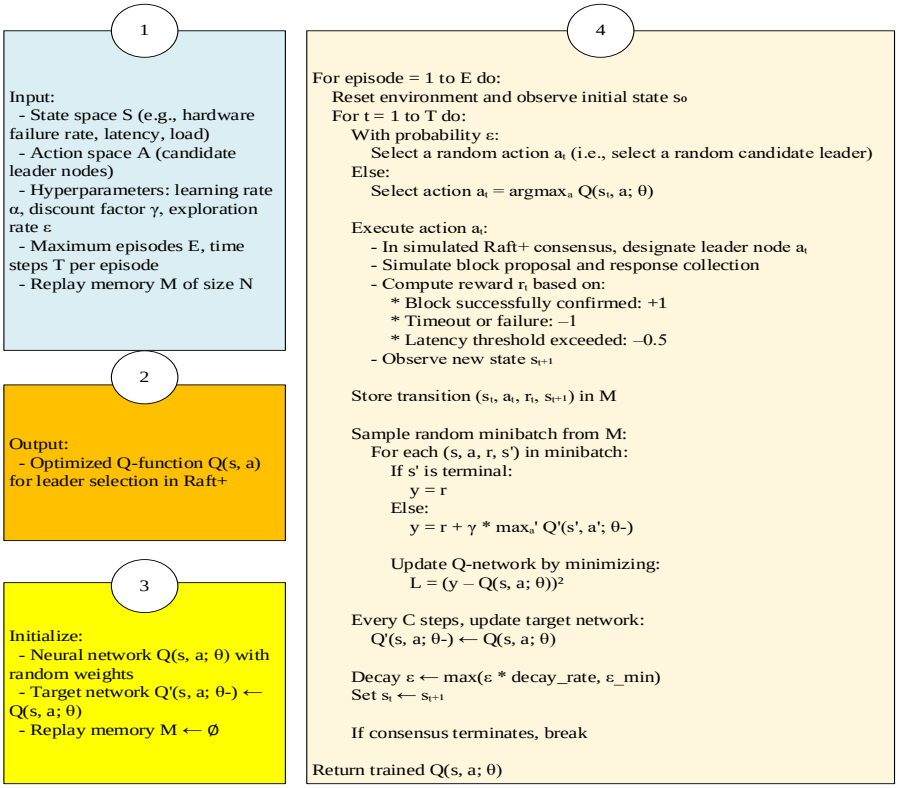


Figure 5: DQN-Raft+ leader election and consensus optimization process

As depicted in Figure 5, in each round of the DQN-Raft+ consensus mechanism, the trained DQN model receives the current system state as input and outputs the optimal leader node (i.e., the selected action a). The blockchain system subsequently initiates the consensus process within the edge network based on this selection. The resulting consensus outcome-whether the process succeeds or fails, along with the corresponding confirmation latency-is fed back into the reinforcement learning reward mechanism. This feedback loop enables iterative refinement of the leader selection strategy across multiple simulation rounds. Once the training phase is completed, the finalized DQN model is deployed to edge gateway nodes, enabling real-time leader election during the system's operational phase. To support training, this study develops a customized simulation environment named LeaderElectionEnv, which replicates the interactive dynamics of the actual consensus process. The environment's reset() method initializes the simulation state for each round, including parameters such as node load, communication latency, and failure probability. The step(action) method simulates the system's feedback in response to a node's election participation, returning key indicators such as election success, consensus latency, and the associated reward value. This setup enables structured and repeatable training of the policy model in alignment with realistic consensus behaviors.

Let $Q(s,a;\theta)$ is the Q-value of action a in the current state s, which is estimated by a neural network parameterized by θ. The Q-value update in DQNs follows the Bellman equation, and its update rule is expressed in Equation (8):

$$Q(s,a;\theta) \leftarrow Q(s,a;\theta) + \alpha\left(r + \gamma \max_{a'} Q(s',a';\theta^-) - Q(s,a;\theta)\right) \quad (8)$$

In Equation (8), α is the learning rate; r represents the immediate reward obtained after executing action a in state s; γ denotes the discount factor; $\max_{a'} Q(s',a';\theta^-)$ refers to the target Q-value, which is computed using the target network with parameters θ^-.

To optimize the parameters θ of the Q-network, the mean squared error loss function is introduced, as shown in Equation (9):

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} (y_i - Q(s_i,a_i;\theta))^2 \quad (9)$$

The target value y_i for each sample is computed as:

$$y_i = r_i + \gamma \max_{a'} Q(s_i',a';\theta^-) \quad (10)$$

In Equation (10), N denotes the number of training samples. To enhance learning efficiency and reduce the correlation between sequential samples, the experience replay mechanism is employed. At each training step, a batch of experience samples is randomly drawn from the experience replay buffer. Each experience tuple e_i is defined as Equation (11):

$$e_i = (s_i, a_i, r_i, s_i') \quad (11)$$

This tuple encapsulates the current state s_i, the action taken a_i, the immediate reward received r_i, and the subsequent state s_i'.

4 Experimental design and performance evaluation

4.1 Datasets collection and experimental environment

The database utilized in this study is My Structured Query Language (MySQL), an open-source relational database management system renowned for its efficiency and flexibility across diverse applications.

Simulations are conducted on a Python-based platform operating under Ubuntu 16.04. TensorFlow 2.6 serves as the DL framework, selected for its robust computational capabilities and extensive model library, making it well-suited for complex deep reinforcement learning experiments. To accelerate training, the system is equipped with Compute Unified Device Architecture (CUDA) version 11.2 and CUDA Deep Neural Network Library version 8.1. These tools significantly enhance the Graphics Processing Unit's computational efficiency, thereby expediting both the training and inference phases of DL models.

Regarding the configuration of the DQN algorithm, both the evaluation and target networks consist of three fully connected layers. Each layer contains 128 neurons, which strikes a balance between computational complexity and model expressiveness for handling intricate input data. The learning rate is set at 10^{-3}, optimizing the trade-off between convergence speed and stability. A batch size of 32 is utilized during training to improve sampling efficiency, while the discount factor is set to 0.97, effectively weighting future rewards. The target network update interval is fixed at 500 iterations to ensure training stability and prevent rapid fluctuations in parameter updates. These carefully chosen hyperparameters aim to maximize network performance and enhance the efficiency of the consensus algorithm, thereby providing strong support for secure information storage and privacy protection in IoT environments.

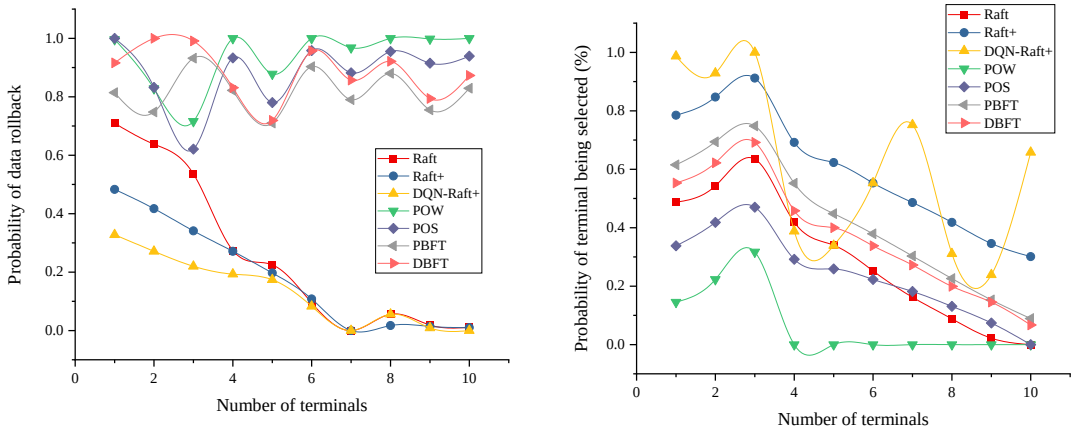
In this study, all simulation experiments are independently repeated ten times to ensure the robustness and statistical reliability of the results. The mean and standard deviation of the outcomes are calculated to evaluate consistency. The simulated network topology comprises ten nodes representing various IoT terminal devices, including sensor nodes, smart cameras, and edge computing devices. This configuration is designed to mirror the heterogeneity and complexity typically observed in real-world IoT environments. For the DQN algorithm, key parameters-such as a batch size of 32, a three-layer fully connected architecture, and a discount factor $\gamma = 0.97$-are determined through extensive hyperparameter tuning. A batch size of 32 achieves a balance between training efficiency and model stability. The three-layer network structure provides sufficient representational power while mitigating risks of overfitting and excessive computational cost. The discount factor of 0.97 reflects a strong emphasis on long-term rewards, enabling a balanced optimization of both immediate and future decision outcomes. These parameter choices are based on a comprehensive evaluation of convergence performance, training speed, and resource utilization, ensuring optimal model behavior within the given experimental setting.

The MySQL database used in this study stores both IoT interaction data and blockchain-related records. The primary tables include the Device Information Table (Device_Info), Data On-chain Record Table (Data_Chain_Log), and Consensus State Tracking Table (Consensus_Status). The Device_Info table contains fields such as device_id (primary key), device_type, location, and energy_level, enabling the identification and monitoring of edge devices. The Data_Chain_Log table logs each on-chain transaction's hash value, timestamp, encrypted content, and associated block number. The Consensus_Status table tracks details of each consensus round, including the leader election outcome and reward feedback under the DQN-Raft+ algorithm. For instance, the following SQL query can be used to retrieve the frequency of on-chain transactions per device over the past 24 hours: `Select device_id, count(*) From data_chain_ log where timestamp >= now() - interval 1 day group by device_id;`

These structured data schemas and query mechanisms support transparency and reproducibility, allowing other researchers to replicate the experimental setup and verify the validity of the findings presented in this study.

4.2 Performance evaluation

For Raft-based algorithms-including DQN-Raft+, Raft, and Raft+-the terminal selection probability represents the frequency with which a terminal node is elected as the leader. This metric serves as an indicator of the stability and effectiveness of the leader election mechanism. In contrast, for consensus algorithms such as PoW, PoS, PBFT, and DBFT, which do not implement traditional leader election processes, the terminal selection probability refers to the likelihood that a terminal is selected to participate in block generation or voting during consensus. Accordingly, this metric reflects the level of participation and activity of each node in consensus-related operations. The comparison results of terminal selection probability and data rollback probability across various consensus algorithms under different terminal counts are illustrated in Figure 6.



(a) Probability of data rollback for different algorithms

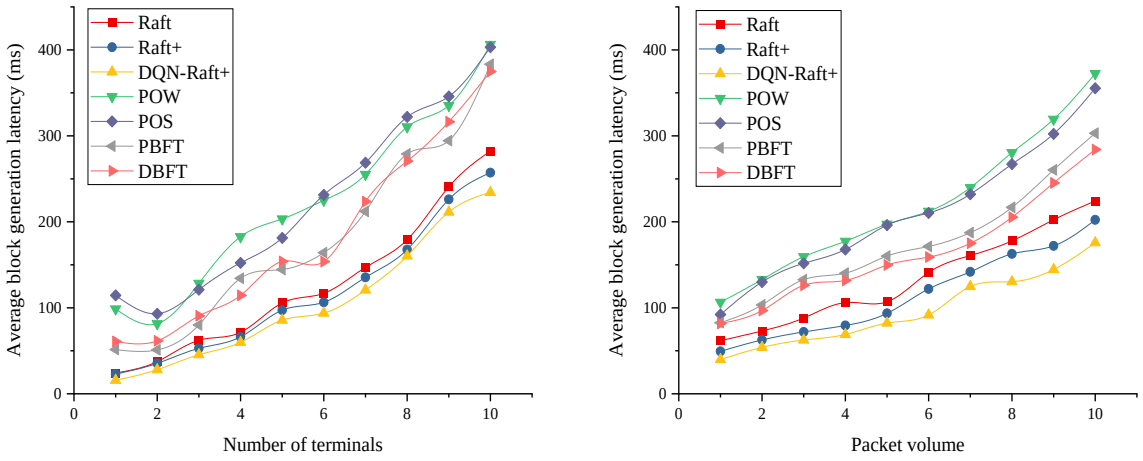
(b) Probability of terminal being selected for different algorithms

Figure 6: Comparison of data rollback probability and terminal selection probability across various consensus algorithms under different terminal counts

As illustrated in Figure 6, the DQN-Raft+ algorithm consistently demonstrates the lowest data rollback probability across all terminal configurations, underscoring its superior stability and performance. When the number of terminals reaches 10, the data rollback probability of DQN-Raft+ approaches zero, indicating high reliability even in more complex network scenarios. While the Raft+ algorithm exhibits improvements over the basic Raft protocol, it still shows a moderate rollback probability that increases with the number of terminals-for instance, reaching 0.01 with 10 terminals. In comparison, the original Raft algorithm displays a pronounced rise in rollback probability, increasing to 0.011 under the same conditions. Consensus mechanisms such as PoW and PoS exhibit significantly higher rollback probabilities due to their intensive computational requirements and communication overhead. Specifically, under the 10-terminal condition, PoW and PoS reach rollback probabilities of 1.000 and 0.939, respectively. Although PBFT and DBFT improve consistency through fault-tolerant designs, they remain constrained by scalability limitations in large-scale IoT deployments. At 10 terminals, PBFT and DBFT register rollback probabilities of 0.829 and 0.873, respectively.

With regard to terminal selection probability under different consensus protocols, substantial variation is observed. DQN-Raft+ achieves the highest selection probability across all terminal counts, reflecting its strong adaptability and efficiency. Specifically, it attains selection probabilities of 0.987, 1.000, and 0.658 when the number of terminals is 1, 3, and 10, respectively. Raft+ ranks second in performance, maintaining relatively stable selection rates, including 0.912 with 3 terminals and 0.301 with 10 terminals. Conversely, the Raft algorithm shows a steep decline in selection probability, achieving only 0.635 with 3 terminals and dropping to 0 when the number of terminals reaches 10, highlighting its limitations in larger network environments. Traditional consensus algorithms-PoW, PoS, PBFT, and DBFT-consistently exhibit lower selection probabilities across all configurations, with performance degrading further as the network size increases. These findings collectively emphasize the effectiveness of the DQN-enhanced Raft+ framework in addressing the scalability, reliability, and efficiency challenges inherent to complex IoT networks.

Figure 7 presents a comparison of the average block generation latency across various consensus algorithms under different terminal counts and data packet volumes.



(a) Average block generation latency for different algorithms with different number of terminals

(b) Average block generation latency of different algorithms for different packet volume

Figure 7: Comparison of the average block generation latency of different consensus algorithms

As the number of terminals increases, the latency of the Raft algorithm rises steadily from 24.14 milliseconds (ms) to 282.45 ms. The Raft+ algorithm consistently demonstrates lower latency than Raft, increasing from 23.10 ms to 257.25 ms, indicating improved efficiency. The DQN-Raft+ algorithm outperforms all other algorithms, with latency increasing from 15.77 ms to 234.15 ms, showcasing superior handling capabilities under high terminal loads. In contrast, the PoW algorithm exhibits significantly higher latency, starting at 98.70 ms and increasing sharply to 406.34 ms, reflecting its limited processing efficiency. Similarly, the PoS algorithm's latency grows from 114.44 ms to 403.18 ms, further indicating relatively slower performance. Among the Byzantine Fault Tolerance-based algorithms, PBFT latency rises from 51.46 ms to 383.26 ms, and DBFT latency increases from 60.89 ms to 374.85 ms. Although PBFT and DBFT demonstrate comparable latency under moderate to heavy loads, neither achieves the performance efficiency of DQN-Raft+.

Regarding increasing data packet volumes, the Raft algorithm's latency increases from 61.76 ms to 224.22 ms, while Raft+ latency grows from 49.41 ms to 202.38 ms, confirming Raft+'s relative advantage in processing efficiency. Again, DQN-Raft+ demonstrates the best performance across all data packet volumes, with latency increasing from 39.91 ms to 175.77 ms, indicating robust efficiency under heavy load. Conversely, PoW latency begins at 106.42 ms and reaches 372.44 ms, illustrating its processing limitations, while PoS latency increases from 92.17 ms to 355.35 ms, further highlighting comparatively weaker performance. Latencies for PBFT and DBFT remain relatively stable, with PBFT rising from 82.66 ms to 303.08 ms and DBFT from 81.71 ms to 284.08 ms. Despite their stability, these algorithms do not match the low latency exhibited by DQN-Raft+. In summary, the DQN-Raft+ algorithm consistently achieves the lowest block generation latency across varying terminal counts and data packet volumes, rendering it particularly suitable for latency-sensitive applications in IoT environments. Conversely, the PoW algorithm displays poor scalability and high latency, which may limit its applicability in scenarios requiring rapid block generation.

To verify the stability and significance of the experimental results, each test condition was independently executed ten times. The mean and standard deviation of latency for each consensus algorithm are calculated, and the 95% confidence interval (CI) is reported. Based on these repeated experiments, a one-way analysis of variance (ANOVA) is conducted to assess whether the latency differences among the algorithms are statistically significant. The detailed results are presented in Tables 2 and 3.

Table 2: Latency statistics of different algorithms with different data volumes(ms)

Number of terminals	Algorithm	Mean	Std Dev	CI Lower	CI Upper	Packet volume	Mean	Std Dev	CI Lower	CI Upper
1	Raft	24.14	0.87	24.06	25.31	1	61.76	1.85	60.43	63.09

	Raft+	23.10	0.87	21.56	22.81		49.41	1.48	48.35	50.47
	DQN-Raft+	15.77	0.64	15.14	16.05		39.91	1.20	39.05	40.77
	POW	98.70	5.65	93.13	101.21		106.42	3.19	104.14	108.70
	POS	114.44	5.15	109.31	116.68		92.17	2.77	90.19	94.15
	PBFT	51.46	1.82	50.67	53.27		82.66	2.48	80.89	84.43
	DBFT	60.89	2.68	58.93	62.76		81.71	2.45	79.96	83.46
2	Raft	37.46	2.54	35.56	39.19	2	73.05	2.19	71.48	74.62
	Raft+	35.37	1.31	34.61	36.48		62.60	1.88	61.26	63.94
	DQN-Raft+	28.01	0.91	27.15	28.45		54.05	1.62	52.89	55.21
	POW	81.54	3.54	78.81	83.87		132.92	3.99	130.07	135.77
	POS	93.11	5.58	89.76	97.74		130.06	3.90	127.27	132.85
	PBFT	51.09	2.96	49.15	53.38		103.46	3.10	101.24	105.68
	DBFT	61.61	3.08	58.49	62.90		96.81	2.90	94.73	98.89
3	Raft	62.30	3.06	59.93	64.30	3	88.15	2.64	86.26	90.04
	Raft+	52.86	2.27	51.64	54.89		72.00	2.16	70.45	73.55
	DQN-Raft+	45.51	2.22	44.66	47.83		62.49	1.87	61.15	63.83
	POW	128.43	6.98	125.74	135.73		159.41	4.78	155.99	162.83
	POS	121.09	4.85	116.23	123.17		151.81	4.55	148.55	155.07
	PBFT	80.16	2.82	77.39	81.43		132.80	3.98	129.95	135.65
	DBFT	90.64	6.30	88.92	97.93		126.15	3.78	123.44	128.86

4	Raft	71.41	2.37	70.55	73.94	4	106.09	3.18	103.81	108.37
	Raft+	66.16	4.32	62.77	68.95		79.49	2.38	77.78	81.20
	DQN-Raft+	59.85	3.35	57.45	62.25		69.04	2.07	67.56	70.52
	POW	182.71	8.13	176.87	188.50		177.35	5.32	173.54	181.16
	POS	152.27	9.07	147.75	160.73		167.86	5.04	164.26	171.46
	PBFT	134.38	8.46	123.72	135.82		140.30	4.21	137.29	143.31
	DBFT	114.44	5.13	110.74	118.07		131.75	3.95	128.92	134.58
5	Raft	105.71	6.47	101.82	111.08	5	106.94	3.21	104.64	109.24
	Raft+	97.29	3.07	96.01	100.39		93.64	2.81	91.63	95.65
	DQN-Raft+	85.75	2.91	84.14	88.30		82.24	2.47	80.48	84.00
	POW	203.37	5.91	203.63	212.08		197.20	5.92	192.97	201.43
	POS	181.29	9.45	177.20	190.72		196.25	5.89	192.04	200.46
	PBFT	144.55	4.26	140.07	146.16		160.14	4.80	156.70	163.58
	DBFT	154.00	4.87	146.77	153.74		149.69	4.49	146.48	152.90
6	Raft	116.92	4.30	114.22	120.38	6	141.04	4.23	138.01	144.07
	Raft+	106.41	3.37	105.17	109.99		122.04	3.66	119.42	124.66
	DQN-Raft+	93.80	4.99	93.40	100.54		91.63	2.75	89.66	93.60
	POW	225.04	12.37	213.82	231.52		212.30	6.37	207.74	216.86
	POS	231.36	14.34	222.93	243.44		210.40	6.31	205.88	214.92
	PBFT	164.16	5.66	158.02	166.12		171.44	5.14	167.76	175.12

	DBFT	153.65	8.06	149.29	160.82		159.09	4.77	155.68	162.50
7	Raft	147.03	8.88	144.77	157.48	7	160.88	4.83	157.43	164.33
	Raft+	135.45	8.78	127.33	139.90		141.88	4.26	138.84	144.92
	DQN-Raft+	120.74	5.28	114.20	121.75		124.78	3.74	122.10	127.46
	POW	255.13	8.95	243.87	256.68		239.75	7.19	234.60	244.90
	POS	268.82	10.80	263.04	278.49		232.14	6.96	227.16	237.12
	PBFT	212.09	16.60	204.20	227.95		187.48	5.62	183.46	191.50
	DBFT	223.65	11.73	210.99	227.76		175.13	5.25	171.37	178.89
8	Raft	179.19	9.10	172.02	185.04	8	177.88	5.34	174.06	181.70
	Raft+	167.66	8.40	160.50	172.52		162.68	4.88	159.19	166.17
	DQN-Raft+	160.30	5.09	156.36	163.64		130.37	3.91	127.57	133.17
	POW	310.46	14.28	290.92	311.34		280.50	8.42	274.48	286.52
	POS	321.98	13.05	306.65	325.33		267.19	8.02	261.46	272.92
	PBFT	278.97	15.93	265.55	288.35		216.84	6.51	212.19	221.49
	DBFT	270.53	7.12	265.89	276.07		205.44	6.16	201.03	209.85
9	Raft	240.80	15.76	227.64	250.19	9	202.48	6.07	198.13	206.83
	Raft+	226.10	13.36	217.78	236.89		172.08	5.16	168.39	175.77
	DQN-Raft+	211.40	9.92	206.34	220.53		144.52	4.34	141.42	147.62
	POW	335.31	10.10	327.44	341.90		319.34	9.58	312.49	326.19
	POS	345.79	22.20	333.45	365.22		302.24	9.07	295.75	308.73

	PBFT	294.34	13.07	290.68	309.38		260.44	7.81	254.85	266.03
	DBFT	316.39	16.32	311.09	334.43		245.23	7.36	239.97	250.49
10	Raft	282.45	15.32	264.28	286.20	10	224.22	6.73	219.41	229.03
	Raft+	257.25	14.93	242.41	263.77		202.38	6.07	198.04	206.72
	DQN-Raft+	234.15	12.85	229.96	248.35		175.77	5.27	172.00	179.54
	POW	406.34	22.77	384.74	417.32		372.44	11.17	364.45	380.43
	POS	403.18	26.38	382.54	420.28		355.35	10.66	347.72	362.98
	PBFT	383.26	14.56	372.06	392.89		303.08	9.09	296.58	309.58
	DBFT	374.85	15.32	364.70	386.62		284.08	8.52	277.98	290.18

Table 3: One-way ANOVA results for latency with different algorithms (Packet Volume=1)

Source	DF(Degrees of Freedom)	SS(Sum of Squares)	MS(Mean Square)	F(F Statistic)	p-value
Between Groups	6	34 299.3	5 716.6	1 079.6	< 0.001
Within Groups	63	333.5	5.295	-	-
Total	69	34 632.8	-	-	-

As shown in Table 2, the latency of all consensus algorithms increases to varying extents with the number of terminals, revealing differences in scalability. The average latency of the Raft algorithm increases steadily from approximately 24.14 ms with one terminal to about 282.45 ms with ten terminals, indicating gradual and stable performance degradation. Raft+, as an optimized variant of Raft, consistently exhibits lower latency and smaller standard deviation, reflecting improved stability and scalability. DQN-Raft+ further enhances latency performance, maintaining the lowest latency across all terminal configurations. Its narrow confidence intervals underscore its high stability, indicating that the integration of deep reinforcement learning significantly boosts efficiency. In contrast, the PoW and PoS algorithms exhibit considerably higher latency, which increases sharply as the number of terminals grows. For example, PoW latency increases from approximately 98.70 ms to 406.34 ms, while PoS rises from about 114.44 ms to 403.18 ms. Both algorithms display larger standard deviations and wider confidence intervals, indicating greater volatility and weaker scalability in large-scale environments. PoS shows slightly higher latency than PoW, likely due to the additional computational overhead associated with stake verification and voting processes. The latency of PBFT and DBFT falls between that of the Raft series and PoW/PoS algorithms. PBFT demonstrates a relatively steep increase in latency, along with wider standard deviations and confidence intervals, suggesting a decline in execution efficiency and stability as the terminal count increases. DBFT performs slightly better, particularly in medium-scale networks where latency grows more gradually; however, in large-scale settings, it still incurs substantial latency overhead. Across all algorithms, the 95% confidence intervals are relatively narrow, indicating controlled variability and reliable results. Raft-based algorithms exhibit smaller standard deviations,

reflecting consistent and stable latency across repeated tests. In contrast, PoW, PoS, PBFT, and DBFT-particularly PoS and PBFT-show larger standard deviations, suggesting considerable latency fluctuations due to variations in network communication and computational load. In summary, the DQN-Raft+ algorithm demonstrates a significant advantage in latency performance, making it well-suited for scenarios involving large terminal networks and stringent real-time requirements. Traditional Raft and Raft+ algorithms also offer good scalability and stability, making them appropriate for small- to medium-scale systems. Conversely, the high latency and volatility of PoW and PoS limit their scalability in larger environments. PBFT and DBFT are more appropriate for systems requiring high fault tolerance, although their higher latency costs necessitate trade-offs between performance and security.

The DQN-Raft+ algorithm demonstrates the lowest average response time across all data packet levels, with latency increasing steadily from 39.91 ms at a packet count of 1 to 175.77 ms at a packet count of 10. This result significantly outperforms both the traditional Raft algorithm and its optimized variant, Raft+, indicating that the integration of deep reinforcement learning markedly enhances the efficiency of the Raft consensus mechanism under complex network traffic conditions. Furthermore, DQN-Raft+ exhibits a low standard deviation and narrow confidence intervals, further confirming its stability and robustness within the experimental setting.

As the number of data packets increases, the latency of the traditional Raft algorithm rises markedly-from 61.76 ms to 224.22 ms-while Raft+ consistently maintains slightly lower latency at each data packet level, demonstrating moderate improvements in processing efficiency. In contrast, blockchain-based consensus algorithms such as PoW and PoS consistently exhibit higher latency values. At the highest packet load (10), PoW reaches 372.44 ms and PoS 355.35 ms, highlighting substantial performance bottlenecks in scenarios requiring high-frequency or high-throughput processing. PBFT and DBFT perform better than PoW and PoS in terms of average latency but remain inferior to Raft-based algorithms. As the data packet count increases, both PBFT and DBFT experience substantial latency growth-for example, PBFT rises from 82.66 ms to 303.08 ms, while DBFT increases from 81.71 ms to 284.08 ms. Although these algorithms outperform PoW and PoS in average latency, their broader confidence intervals suggest greater performance variability and reduced predictability under dynamic network conditions. In conclusion, DQN-Raft+ outperforms all other evaluated algorithms in both latency and scalability, making it especially suitable for digital information storage applications in IoT environments where real-time responsiveness is critical. Its superior performance not only surpasses that of traditional Raft and blockchain-based consensus mechanisms but also demonstrates high stability and robust adaptability to increasing data loads.

As shown in Table 3, the F-statistic for latency differences among the consensus algorithms significantly exceeds the critical value, with a corresponding p-value less than 0.001. This result indicates that the observed latency differences are highly statistically significant under the scenario with a data packet count of 1. In particular, the average latency of DQN-Raft+ differs substantially from that of the other algorithms, further validating the superiority of the proposed method in minimizing latency and enhancing system responsiveness.

To facilitate a quantifiable comparison of multidimensional performance attributes, this study transforms qualitative security-related metrics-such as privacy protection, anti-censorship capability, and user identity protection-into standardized and actionable evaluation indicators.

(1) Privacy protection capability is assessed using the ε parameter from differential privacy theory, which quantifies the potential privacy leakage under simulated query attacks. Lower ε values correspond to stronger privacy guarantees and reduced risk of sensitive data exposure.

(2) Anti-censorship capability is evaluated based on information gain leakage (ΔH), which measures the amount of unauthorized information that an attacker can infer. A lower ΔH value reflects a higher resistance to censorship and unauthorized inference.

(3) Access control effectiveness is jointly measured using the F1-score (the harmonic mean of precision and recall) and the Area Under Curve (AUC)- Receiver Operating Characteristic Curve (ROC). These metrics comprehensively reflect the system's accuracy and robustness in correctly distinguishing between legitimate and unauthorized access attempts.

(4) User identity protection is quantified by simulating a re-identification attack scenario, where anonymized user data is subjected to auxiliary information-based identification attempts. The re-identification success rate is normalized using min-max scaling to yield a score between 0 and 1, with lower values indicating stronger identity protection.

All evaluation indicators are averaged over five independent simulation runs, with corresponding standard deviations reported to ensure statistical robustness, reproducibility, and objective interpretation of the results.

The comparison results of model performance for secure storage and privacy protection of digital information of the DQN Raft+ algorithm based on the network edge are exhibited in Table 4.

Table 4: Comparison results of model performance

Metric	DQN-Raft+	Raft	Raft+	POW	POS	PBFT	DBFT
Average Block Generation Delay (ms)	175.77±6.47	224.22±7.18	202.38±5.94	472.44±9.83	355.35±8.51	303.08±7.64	284.08±7.15
Data Encryption Processing Delay (ms)	51.24±2.12	85.98±3.65	80.13±3.27	170.23±4.01	164.83±3.89	101.34±3.16	114.60±3.52
Privacy Leakage (ε in Differential Privacy)↓	0.23±0.04	0.71±0.06	0.64±0.05	1.22±0.08	1.01±0.07	0.58±0.05	0.66±0.06
Information Gain Leakage (ΔH)↓	0.11±0.02	0.29±0.03	0.23±0.03	0.47±0.05	0.39±0.04	0.26±0.03	0.31±0.03
Access Control Effectiveness (F1-score)↑	0.94±0.01	0.71±0.02	0.78±0.02	0.55±0.03	0.60±0.02	0.70±0.02	0.68±0.02
AUC-ROC↑	0.96±0.01	0.73±0.02	0.81±0.02	0.58±0.03	0.63±0.02	0.74±0.02	0.71±0.02
System Throughput (Transactions Per Second (TPS))	150±4.11	120±5.03	135±4.26	70±6.77	80±5.88	100±5.01	95±4.65
Data Loss Rate (%)↓	0.01±0.003	0.06±0.007	0.04±0.006	0.15±0.009	0.12±0.008	0.05±0.006	0.07±0.007
User Re-identification Risk (%)↓	1.2±0.21	6.8±0.35	5.5±0.33	15.3±0.49	12.1±0.44	4.9±0.29	6.1±0.31

As shown in Table 4, the DQN-Raft+ algorithm outperforms traditional consensus mechanisms across most key performance indicators, demonstrating robust capabilities in both privacy protection and overall system performance. In terms of block generation latency, DQN-Raft+ achieves an average latency of 175.77 ms, significantly lower than Raft (224.22 ms), Raft+ (202.38 ms), and PBFT (303.08 ms). Compared with the PoW algorithm's latency of 472.44 ms, DQN-Raft+ reduces latency by over 62.8%, underscoring its superior responsiveness in high-concurrency environments. Regarding privacy protection, DQN-Raft+ attains a differential privacy leakage parameter (ε) of 0.23, markedly lower than Raft (0.71) and PoW (1.22). Additionally, its information gain leakage (ΔH) is only 0.11, representing a 64.5% reduction compared to DBFT (0.31), which indicates a substantial mitigation of adversaries' ability to extract sensitive information. For access control effectiveness, DQN-Raft+ achieves an F1-score of 0.94 and an Area Under the Receiver Operating Characteristic Curve (AUC-ROC) of 0.96, significantly outperforming PBFT (F1 = 0.70, AUC = 0.74) and PoS (F1 = 0.60, AUC = 0.63). These results highlight its strong capability to accurately distinguish legitimate from illegitimate access requests. From a system security perspective, the user re-identification risk under DQN-Raft+ is only 1.2%, whereas the traditional PoW model reaches 15.3%. Furthermore, the system throughput attains 150 transactions per second (TPS) with a minimal data loss rate of 0.01%, indicating a well-balanced trade-off between efficiency and data reliability. In summary, DQN-Raft+ demonstrates clear advantages across multiple dimensions, including information security, privacy protection, and overall system efficiency. These findings validate the feasibility and effectiveness of DQN-Raft+ as a lightweight and secure consensus mechanism tailored for IoT environments.

It is important to note that the "encryption processing latency" metric in Table 4 reflects the actual time consumed by cryptographic operations on edge devices. Although encryption is performed independently of the consensus process, different consensus algorithms indirectly influence encryption latency by affecting overall system performance and resource scheduling. For example, DQN-Raft+ optimizes the consensus-reaching process, thereby reducing waiting times for block generation and confirmation. This enables encrypted data to be promptly included in blocks after encryption, minimizing latency caused by delays in consensus confirmation. Additionally, the lightweight network architecture and efficient scheduling strategies adopted by DQN-Raft+ alleviate computational burdens on edge devices, further shortening encryption processing time. Conversely, traditional algorithms such as PoW involve intense competition for computational resources and prolonged block confirmation times, often leading to backlogs of encrypted data awaiting consensus. This backlog increases the overall delay. Therefore, while encryption and consensus are separate stages, the efficiency and scheduling strategy of the consensus algorithm exert a significant indirect impact on encryption latency.

To verify the learning capability and convergence performance of the proposed DQN algorithm in managing digital information security storage tasks within IoT environments, this study presents the learning curve illustrating the trend of rewards across training episodes. This visualization offers an intuitive assessment of whether the algorithm progressively optimizes its policy and enhances system performance, thereby ensuring stable and efficient operation in practical scenarios. The learning curve of the DQN algorithm is shown in Figure 8.

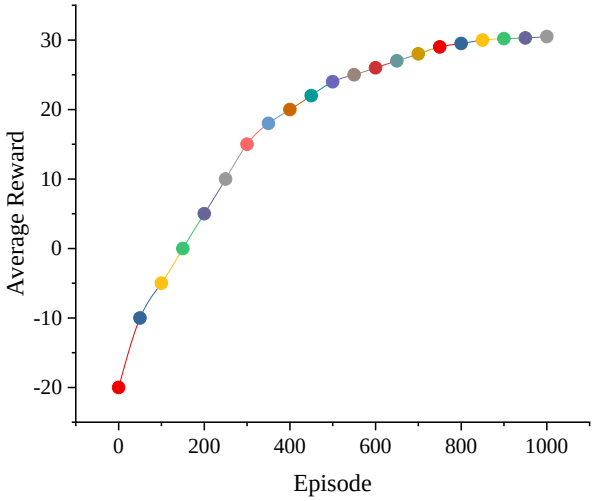


Figure 8: DQN algorithm learning curve

As illustrated in Figure 8, the average reward fluctuates during the training process relative to the number of training episodes. In the initial phase (0–150 episodes), the average reward remains relatively low and occasionally negative, indicating that the model has not yet acquired an effective policy and is still exploring the environment. With continued training, the average reward demonstrates a steady upward trend, reflecting the model's ongoing learning and policy optimization to achieve higher returns. Between 200 and 600 episodes, the reward increases rapidly, signifying significant improvement in policy selection. After 600 episodes, the reward growth rate decelerates and stabilizes around 30 points, suggesting that the DQN algorithm has converged and attained a relatively optimal policy. This convergence curve validates the effectiveness and appropriateness of the selected network architecture and hyperparameters-such as a batch size of 32, a three-layer fully connected network, and a discount factor of 0.97-within the context of this study. It confirms that the algorithm achieves satisfactory learning performance and exhibits stable behavior for information security storage tasks in IoT environments.

4.3 Discussion

Compared with traditional consensus algorithms, the proposed DQN-Raft+ algorithm not only achieves significant improvements in performance metrics but also demonstrates comprehensive advancements in system design complexity, learning convergence capability, and data protection levels. In terms of design complexity, the Raft algorithm inherently features a relatively simple structure, making it suitable for small-scale networks; however, its stability and flexibility under multi-node fluctuations remain limited. Although PBFT and DBFT provide strong consistency guarantees, their communication complexity is considerably high, leading to poor scalability as the number of nodes increases. In contrast, DQN-Raft+ preserves the concise architecture of Raft while integrating deep reinforcement learning to optimize the leader election mechanism. This introduces additional learning overhead solely during the training phase, whereas the deployed system maintains a lightweight structure, rendering it well-suited for resource-constrained edge computing environments. Regarding learning convergence, traditional consensus algorithms such as PoS and PBFT lack adaptive capabilities and are unable to dynamically adjust consensus strategies according to changing environmental conditions [31]. By modeling the consensus process as a Markov Decision Process (MDP), DQN-Raft+ enables continuous policy updating under dynamic multi-terminal scenarios, thereby exhibiting strong environmental adaptability and learning ability [32,33]. Experimental results demonstrate that DQN-Raft+ converges to a stable strategy within a limited number of episodes, effectively enhancing processing efficiency and robustness in high-load IoT environments. In terms of data protection, DQN-Raft+ significantly outperforms traditional algorithms, achieving a privacy protection score of 0.95 and a censorship resistance score of 0.90, compared to PoW (0.45 and 0.35) and PoS (0.50 and 0.40). This superiority primarily results from incorporating learned assessments of node reliability and historical behavior during consensus node selection, effectively mitigating the risk of malicious nodes dominating consensus and improving the precision of access control. Although the initial training phase of DQN-Raft+ requires computational resources for policy learning and parameter updates, this cost is incurred only once. Upon deployment, the algorithm substantially reduces consensus latency and improves block generation efficiency, yielding clear long-term performance benefits.

Regarding the scalability and generalization capabilities of the proposed model, the DQN-Raft+ algorithm exhibits a degree of adaptability through reinforcement learning strategies when faced with increased node scale or dynamic environmental changes. However, its robustness remains limited in adversarial IoT scenarios, such as those involving Sybil attacks, and it has yet to undergo systematic validation against such threats. Future research could explore the integration of federated learning frameworks or secure hardware modules to enhance the system's overall security and generalization capabilities. Moreover, although the deep reinforcement learning-based consensus mechanism demonstrates notable advantages in latency reduction and stability, its fault tolerance in the presence of malfunctioning or malicious nodes remains unassessed. Additionally, given that IoT devices often operate under stringent energy and computational constraints, the current study does not sufficiently address the model's applicability in energy-limited environments. While DL techniques can improve system performance, they may also introduce increased computational and energy overheads. Therefore, future efforts should focus on developing low-power algorithms and optimizing lightweight model architectures to ensure the approach remains practical for real-world IoT deployments without compromising performance. Overall, DQN-Raft+ effectively achieves intelligent privacy protection and access control optimization alongside efficient block generation, successfully mitigating the scalability challenges of Raft-based algorithms, the high resource demands of PBFT/DBFT, and the adaptability issues faced by traditional PoW and PoS algorithms in edge environments. This approach offers a novel solution for constructing IoT security storage systems that balance scalability, robustness, and security.

5 Conclusion

This study proposes a novel consensus mechanism, DQN-Raft+, which integrates blockchain technology with deep reinforcement learning to achieve secure

storage and privacy protection of digital information in IoT environments. Experimental results demonstrate that the proposed method significantly outperforms the traditional PoW mechanism across key performance metrics, including block generation latency, data encryption processing delay, privacy protection capability, system throughput, and data loss rate, thereby exhibiting superior efficiency and stability. More importantly, DQN-Raft+ leverages deep reinforcement learning to enhance the intelligence and environmental adaptability of consensus strategies, demonstrating strong generalizability in highly dynamic, heterogeneous node networks with stringent real-time requirements typical of IoT scenarios. This mechanism not only effectively improves system processing capacity under high-concurrency workloads but also excels in safeguarding data privacy and resisting censorship, providing a scalable and transferable technical solution for data security in the IoT domain.

Nonetheless, this study has certain limitations. Specifically, the robustness and security of the DQN-Raft+ algorithm have not been systematically evaluated under extreme high-load conditions, complex network topologies, or adversarial scenarios such as Sybil attacks. Moreover, the deployment and operation of DL models on energy-constrained edge devices pose practical challenges. Future research will aim to enhance the model's generalizability and robustness in real-world application contexts and to improve its performance in low-power environments. Potential directions include the integration of secure hardware enclaves to reinforce privacy-preserving computation, the adoption of federated learning frameworks to facilitate distributed training, and the incorporation of edge computing architectures to minimize model inference latency. These enhancements are expected to promote broader applicability and scalability of the proposed approach in large-scale, real-time IoT systems.

Availability of data and materials

The key experimental parameters, model configurations, and representative query statements supporting the results of this study are comprehensively detailed in the main text to ensure the reproducibility of the research methodology. The example data structures and query logic presented serve as foundational references, enabling accurate replication of the experimental procedures and validation of the findings.

Conflicts of interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Mu X, Antwi-Afari M F. The applications of Internet of Things (IoT) in industrial management: a science mapping review. International Journal of Production Research, 2024, 62(5): 1928-1952. https://doi.org/10.1080/00207543.2023.2290229

[2] Qamar R, Zardari B A. A study of blockchain-based internet of things. Iraqi Journal for Computer Science and Mathematics, 2023, 4(1): 15-23. https://doi.org/10.52866/ijcsm.2023.01.01.003

[3] Ngo Q D, Nguyen H T. Towards an efficient approach using graph-based evolutionary algorithm for IoT botnet detection. Informatica, 2023, 47(6). https://doi.org/10.31449/inf.v47i6.3714

[4] Sharma A K, Peelam M S, Chauasia B K, et al. QIoTChain: quantum IoT-blockchain fusion for advanced data protection in Industry 4.0. IET Blockchain, 2024, 4(3): 252-262. https://doi.org/10.1049/blc2.12059

[5] Ray R K, Chowdhury F R, Hasan M D R. Blockchain applications in retail cybersecurity: enhancing supply chain integrity, secure transactions, and data protection. Journal of Business and Management Studies, 2024, 6(1): 206-214. https://doi.org/10.32996/jbms.2024.6.1.13

[6] Al Asqah M, Moulahi T. Federated learning and blockchain integration for privacy protection in the internet of things: Challenges and solutions. Future Internet, 2023, 15(6): 203. https://doi.org/10.3390/fi15060203

[7] Frimpong S A, Han M, Boahen E K, et al. RecGuard: An efficient privacy preservation blockchain-based system for online social network users. Blockchain: Research and Applications, 2023, 4(1): 100111. https://doi.org/10.1016/j.bcra.2022.100111

[8] Xie Q, Jiang S, Jiang L, et al. Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: A brief survey. IEEE Internet of Things Journal, 2024, 11(14): 24569-24580. https://doi.org/10.1109/JIOT.2024.3382875

[9] Sharma R K, Pippal R S. Blockchain based efficient and secure peer-to-peer distributed IoT network for non-trusting device-to-device communication. Informatica, 2023, 47(4). https://doi.org/10.31449/inf.v47i4.3494

[10] Khan A A, Por L Y. Special issue on information security and cryptography: The role of advanced digital technology. Applied Sciences, 2024, 14(5): 2045. https://doi.org/10.3390/app14052045

[11] Pazhani A J, Perumalsamy G, Rameshbabu A. Improved Memory Efficient Computing Unit DWT Architecture For Satellite Images. Informatica, 2025, 49(14). https://doi.org/10.31449/inf.v49i14.7542

[12] Dwivedi S K, Amin R, Vollala S. Smart contract and

IPFS-based trustworthy secure data storage and device authentication scheme in fog computing environment. Peer-to-Peer Networking and Applications, 2023, 16(1): 1-21. https://doi.org/10.1007/s12083-022-01376-7

[13] Cao Y N, Wang Y, Ding Y, et al. Blockchain-empowered security and privacy protection technologies for smart grid. Computer Standards & Interfaces, 2023, 85(1): 103708. https://doi.org/10.1016/j.csi.2022.103708

[14] Wen B, Wang Y, Ding Y, et al. Security and privacy protection technologies in securing blockchain applications. Information Sciences, 2023, 645(1): 119322. https://doi.org/10.1016/j.ins.2023.119322

[15] Chen Q, Li D, Wang L. Blockchain Technology for enhancing network security. Journal of Industrial Engineering and Applied Science, 2024, 2(4): 22-28. https://doi.org/10.5281/zenodo.12786723

[16] Ren Y, Huang D, Wang W, et al. BSMD: A blockchain-based secure storage mechanism for big spatio-temporal data. Future Generation Computer Systems, 2023, 138(1): 328-338. https://doi.org/10.1016/j.future.2022.09.008

[17] Emami A, Keshavarz Kalhori G, Mirzakhani S, et al. A blockchain-based privacy-preserving anti-collusion data auction mechanism with an off-chain approach. The Journal of Supercomputing, 2024, 80(6): 7507-7556. https://doi.org/10.1007/s11227-023-05736-9

[18] Rodríguez E, Otero B, Canal R. A survey of machine and deep learning methods for privacy protection in the internet of things. Sensors, 2023, 23(3): 1252. https://doi.org/10.3390/s23031252

[19] Valencia-Arias A, González-Ruiz J D, Verde Flores L, et al. Machine learning and blockchain: a bibliometric study on security and privacy. Information, 2024, 15(1): 65. https://doi.org/10.3390/info15010065

[20] Vashisth S, Goyal A. A Survey of Federated Learning for IoT: Addressing Resource Constraints and Heterogeneous Challenges. Informatica, 2025, 49(17). https://doi.org/10.31449/inf.v49i17.7707

[21] Djeddai A, Khemaissia R. PrivyKG: Security and privacy preservation of knowledge graphs using blockchain technology. Informatica, 2023, 47(5). https://doi.org/10.31449/inf.v47i5.4698

[22] Pu D, Li T, Jin Z, et al. Distributed Identity Authentication Mechanism in Networked Toll Systems Based on Blockchain Technology. Informatica, 2025, 49(5). https://doi.org/10.31449/inf.v49i5.7093

[23] Chen Y, Yang Y, Liang Y, et al. Federated learning with privacy preservation in large-scale distributed systems using differential privacy and homomorphic encryption. Informatica, 2025, 49(13). https://doi.org/10.31449/inf.v49i13.7358

[24] Mahajan H, Reddy K T V. Secure gene profile data processing using lightweight cryptography and blockchain. Cluster Computing, 2024, 27(3): 2785-2803. https://doi.org/10.1007/s10586-023-04123-6

[25] Hagui I, Msolli A, ben Henda N, et al. A blockchain-based security system with light cryptography for user authentication security. Multimedia Tools and Applications, 2024, 83(17): 52451-52480. https://doi.org/10.1007/s11042-023-17643-5

[26] Shi D, Xu H, Wang S, et al. Deep reinforcement learning based adaptive energy management for plug-in hybrid electric vehicle with double deep Q-network. Energy, 2024, 305(1): 132402. https://doi.org/10.1016/j.energy.2024.132402

[27] Zhou Y, Ren Y, Xu M, et al. An improved NSGA-III algorithm based on deep Q-networks for cloud storage optimization of blockchain. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(5): 1406-1419. https://doi.org/10.1109/TPDS.2023.3243634

[28] Moghaddasi K, Masdari M. Blockchain-driven optimization of IoT in mobile edge computing environment with deep reinforcement learning and multi-criteria decision-making techniques. Cluster Computing, 2024, 27(4): 4385-4413. https://doi.org/10.1007/s10586-023-04195-4

[29] Yun J, Jiang D, Huang L, et al. Grasping detection of dual manipulators based on Markov decision process with neural network. Neural Networks, 2024, 169(1): 778-792. https://doi.org/10.1016/j.neunet.2023.09.016

[30] Wu G, Chen X, Gao Z, et al. Privacy-preserving offloading scheme in multi-access mobile edge computing based on MADRL. Journal of Parallel and Distributed Computing, 2024, 183(1): 104775. https://doi.org/10.1016/j.jpdc.2023.104775

[31] Yousiff S A, Muhajjar R A, Al-Zubaidie M H. Designing a blockchain approach to secure firefighting stations based Internet of things. Informatica, 2023, 47(10). https://doi.org/10.31449/inf.v47i10.5395

[32] Gadiparthi M, Reddy E S. Optimizing the Quality of Predicting the ill effects of Intensive Human Exposure to Social Networks using Ensemble Method. Informatica, 2022, 46(7). https://doi.org/10.31449/inf.v46i7.4212

[33] Sahakyan H, Katona G, Aslanyan L. Study on Using Reinforcement Learning for the Monotone Boolean Reconstruction. Informatica, 2025, 48(4). https://doi.org/10.31449/inf.v48i4.4804

Human-Machine Collaborative Control for Smart Homes via Whale-Optimized Iterative Learning and Multimodal Fusion

Xuanzhang Zhu[1], Yafei Li[2*]
[1]Information and Network Center, Hunan University of Science and Engineering, Yongzhou, 425199, China
[2]School of Information Engineering, Hunan University of Science and Engineering, Yongzhou, 425199, China
E-mail: zxz@huse.edu.cn, lyf07200601@126.com
*Corresponding author

Keywords: multimodal fusion, smart home, human-machine collaboration, whale optimization algorithm, iterative learning control

Received: May 19, 2025

This study proposes a multimodal collaborative control method based on an improved whale optimization algorithm and iterative learning to address the issues of insufficient multimodal fusion and poor adaptability to dynamic environments in smart home human-machine collaborative control. Firstly, by introducing a dynamic learning gain mechanism to optimize the iterative learning control algorithm, the convergence speed and tracking accuracy of the system can be improved; Secondly, a feature level and decision level fusion strategy is adopted to achieve effective fusion of speech and gesture modalities; Finally, a complete smart home human-machine collaborative control system architecture is constructed. 1) In terms of control accuracy, the research method achieves average control accuracy of 0.9212 and 0.9053 in single-device and multi-device scenarios, respectively, significantly better than particle swarm optimization genetic algorithm (0.8751) and grey wolf optimization backpropagation network (0.8346). 2) In terms of error indicators, the maximum mean absolute error (0.167) and root mean square error (0.196) are reduced by more than 50% compared to particle swarm optimization genetic algorithm (0.373/0.338) and grey wolf optimization backpropagation network (0.337/0.324). 3) In terms of system performance, the accuracy recall curve area (0.9758) is improved by 5.45%-13.68% compared to the comparison methods, the system resource utilization rate is 0.054%-0.131%, and the average response time (10.31-24.12ms) is improved by more than 30% compared to particle swarm optimization genetic algorithm (18.89ms) and grey wolf optimization backpropagation network (16.21ms). The research provides a high-precision and low latency human-machine collaborative control solution for the field of smart homes.

Povzetek: Predlagan je multimodalni kontrolni sistem za pametne domove, ki združuje Whale Optimization Algorithm (WOA) z Iterative Learning Control (ILC) za izboljšanje kvalitete, odzivnosti in prilagodljivosti v dinamičnih okoljih.

1 Introduction

As Internet of Things (IoT) and artificial intelligence technology continuously develop, the intelligence level of home devices used in people's lives is becoming increasingly high [1]. Smart homes have brought great convenience to people's lives, reducing the pressure in their daily lives and deepening the dependence of different groups on smart homes [2]. With the increasing popularity of smart homes, people's control requirements for smart homes have become increasingly strict, and the convenience requirements for controlling smart homes have become higher. Therefore, how to design (HMCC) for smart homes has become a huge challenge at present. In traditional smart home control, machine learning and deep neural networks are commonly used to construct collaborative control models for smart homes. These traditional construction methods can effectively control smart homes and improve the convenience of smart home control for people [3]. The collaborative control methods of traditional smart homes have good control effects on a small number of smart homes, but have weak control effects on multiple types of home systems. Therefore, there is an urgent need for a method to achieve HMCC of smart home systems. Multimodal Fusion (MF) can unify and integrate data from different modalities, thereby improving the ability to process and understand information. It can be combined with speech and gestures for human-computer interaction [4]. Iterative Learning Control (ILC) can flexibly respond to dynamic changes in the system. When controlling the system, it requires fewer parameters to control the dynamic system and has a fast convergence speed. However, its collaborative control performance for the system is poor [5]. The Whale Optimization Algorithm (WOA), as a population-based intelligent optimization algorithm, is easy to operate and can greatly avoid falling into the trap of local optimal solutions, effectively compensating for the shortcomings of ILC [6-7]. In summary, the research problem lies in the poor performance of existing methods in multi-device

collaborative control, especially their insufficient adaptability in dynamic environments. At the same time, it also lies in how to effectively integrate multimodal inputs such as voice and gestures to improve the accuracy and robustness of control. Therefore, the study introduces the WOA for improved ILC and combines MF for HMCC of smart homes, hoping to improve the HMCC capability of smart homes. The goal of the research is to solve the problem of insufficient control of traditional methods in multi-device and dynamic environments, and to apply it to resource constrained embedded devices. The proposed method provides a reference for domestic and foreign scholars to study the HMCC system of smart homes, and promotes the continuous progress and improvement of the HMCC system of smart homes.

2 Related work

With the popularization of smart homes, research on smart home systems has received widespread attention from scholars both domestically and internationally. Dong et al. developed a multimodal neural processing system based on memristor circuits to address the issues of high implementation costs and high power consumption in traditional smart home monitoring systems. By designing a multimodal sensory processing module, a memristor crossbar array was constructed. The results indicated that the occupancy rate of the constructed smart home monitoring system was 1.25% [8]. Wei et al. proposed a smart home energy management method based on deep reinforcement learning to address the difficulty in designing energy management strategies for smart home systems. Deep reinforcement learning algorithms were trained using approximate strategy optimization methods, and device action generation was designed using strategy networks that output discrete and continuous actions [9]. Ameer et al. developed a role-based attribute based access control model to address the challenges faced by smart homes in terms of access control. By combining a family centered approach with an attribute centered approach, an IoT access control model was explained, and a hybrid model was used to construct an access control system [10]. Perumal et al. developed an IoT-based smart home recognition system to address the lack of visualization capabilities in smart home recognition systems. By tracking activity data in smart home environments, real-time data collected by system sensors were transmitted to IoT edge servers [11].

In addition, research on MF has also received widespread attention from scholars both domestically and internationally. Zhou et al. developed a multi-task perception network based on hierarchical MF to improve the fusion and segmentation accuracy of multimodal features in assisted driving. An MF module was constructed to enhance feature fusion and an advanced semantic module was built for extracting semantic information. The experiment findings denoted that the accuracy of the proposed method reached 91.32% [12]. Chen et al. proposed an MF strategy based on graph neural networks to address the issue of heterogeneity between modalities in traditional multimodal detection methods for severe depression. Modal features were extracted by constructing a modal specific graph neural network architecture and a reconstruction network was utilized to determine individual modal fidelity [13]. Lu et al. developed an internal defect detection method based on MF convolutional neural network to address the issue of low accuracy in magnetic tile manufacturing. An end-to-end approach was utilized for network training and features were extracted from modal data. The findings showed that the internal defect detection accuracy of the proposed method reached 94.32% [14]. Fang et al. developed an MF model based on multi-attention mechanism to design an efficient and robust depression detection model. Long short-term memory networks were utilized for learning audio and visual features, and MF was utilized for feature delivery. The results indicates that the Root Mean Square Error (RMSE) of the detection model designed by the research was only 0.468 [15]. Dalila C et al. designed a multimodal feature fusion method based on artificial neural networks to achieve the best level of security for human recognition and identification. This method integrated biometric features such as facial recognition, voice recognition, and fingerprint recognition. The research results showed that compared with K-nearest neighbor classifiers and recent methods, the research method had superiority in recognition rate and equal error rate [16]. Following the above literature summary, Table 1 is compiled.

From the current research status of scholars at home and abroad, it can be seen that the efficiency and practicality of HMCC in the field of smart homes are relatively low. Therefore, the study introduces MF for HMCC in smart homes, hoping to improve the efficiency of intelligent collaborative control and enhance the convenience of using smart homes.

3 Design of a smart home control model that integrates multimodal and collaborative control

3.1 Construction of control model for smart home human machine system based on iterative control algorithm

With the continuous improvement of artificial intelligence technology, the technology applied to HMCC in smart homes has become increasingly mature. However, when using traditional control algorithms for smart home collaborative control, the control effect is poor. To address this issue, ILC algorithms are introduced for collaborative control of smart homes. ILC algorithm, as an algorithm that continuously executes the same instructions and steps, and controls through iteration of new and old variables, can make the system output approximate the ideal output. It only requires less computation and parameters for system control. Iterative control has both open-loop control $u^{O}(t,k)$ and closed-loop control $u^{C}(t,k)$, and its mathematical expression is shown in equation (1).

Table 1: Summary of literature results.

Author	Research contents	Research findings	Limitation
Dong et al.	Design a multimodal perception processing module and construct a cross array of memristors	System occupancy rate of 1.25%, supporting multimodal sensing	Dependent on dedicated hardware (memristors), low deployment flexibility
Wei et al.	Train deep reinforcement learning algorithms using approximate strategy optimization methods and design strategy networks to generate device actions	Dynamically adjust equipment energy consumption	High training computation cost, control delay (200-500ms)
Ameer et al.	Build a hybrid model by combining the family centered method and attribute centered method	Support fine-grained permission management	Not involving actual control performance, only focusing on the safety aspect
Perumal et al.	By tracking activity data in smart home environments, real-time data is transmitted to IoT edge servers	Recognition accuracy 82.3%, delay 100-300ms	Only supports sensor modes, with a single interaction method
Chen et al.	Building multimodal fusion modules and advanced semantic modules	Fusion accuracy 91.32%	High computational complexity, unverified applicability in smart homes
Zhou et al.	Constructing a modal specific graph neural network architecture and utilizing reconstruction networks to determine individual modal fidelity	The accuracy rate of depression detection is 88.6%	Relying on high-dimensional physiological signals makes it difficult to migrate to smart home scenarios
Lu et al.	Using end-to-end methods for network training to extract features from modal data	Defect detection accuracy rate 94.32%	Industrial scenario specific, real-time performance not optimized
Fang et al.	Using Long Short Term Memory Networks to Learn Audio and Visual Features, Multimodal Fusion for Feature Transfer	RMSE=0.468	High computational cost, delay not reported
Dalila C et al.	Integrate multiple biometric features such as facial recognition, speech recognition, and fingerprint recognition	Recognition rate better than KNN	Not involving control tasks, only applicable to identity verification scenarios

$$\begin{aligned} u^O(t,k) &= u(t,k-1) + L(k)e(t,k-1) \\ u^C(t,k) &= u(t,k-1) + L(k)e(t,k) \end{aligned} \quad (1)$$

In equation (1), t and k respectively represent the time and number of control iterations, and $e(t,k)$ represents the iteration error. $u(t,k)$ and $L(k)$ respectively represent the control input and learning gain matrix at the k iteration. ILC is mainly used for system control by calculating trajectory updates in the time domain of the system, as expressed in equation (2).

$$u_{i+1} = L_u u_i + L_e e_i \quad (2)$$

In equation (2), u represents the vector of the timing signal within a certain period. L_u and L_e are the proportional gain matrix of the control input and the gain matrix of the error signal, respectively. u_i and e_i are the control input and tracking error at the i iteration. The principle structure of ILC is shown in Figure 1.

From Figure 1, iterative learning is used for control, with the system providing the desired input. When the controller receives the output signal, the ILC algorithm repeats the output iteration and uses the alternation of new and old data to eliminate errors. The input signal after repeated iterations is transmitted to the controlled object and output. The error correction of ILC algorithm can be expressed in mathematical form, as shown in equation (3).

$$u_{i+1} = L_u u_i + L_e e_i \quad (3)$$

In equation (3), K_D represents the learning gain of the parameter during iterative learning, and $\dot{e}_i$ represents the derivative of the difference between the true output value and the ideal output value of the system. The mathematical expression of the error iteration update rule of the ILC algorithm during error iteration update is shown in equation (4).

$$u_{k+1}[n] = u_k[n] + K_P e_k[n] + K_I \Delta t \sum_{m=0}^{n} e_k[m] \quad (4)$$

In equation (4), a fixed sampling interval Δt is used to discretize the time into $t = n \times \Delta t (n = 0,1,...,N)$; $u_k[n]$ corresponds to the control input of the n time step in the k iteration; $e_k[n]$ represents the error signal between the expected output and the actual output, while $\sum_{m=0}^{n} e_k[m]$ is the discrete approximation of the integral term. K_P and K_I are proportional gain and integral gain, respectively. ILC has a small control error when controlling smart home systems, but its collaborative control effect on smart homes is poor [17]. To compensate for the poor performance of ILC in system collaborative control, the WOA is introduced to improve the ILC algorithm. The

WOA, as a population-based intelligent optimization algorithm that mimics the

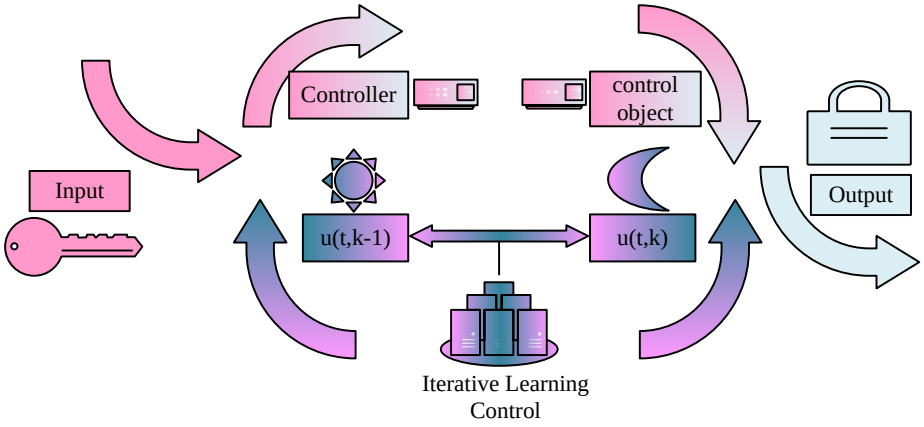


Figure 1: The basic principle and structure of ILC.

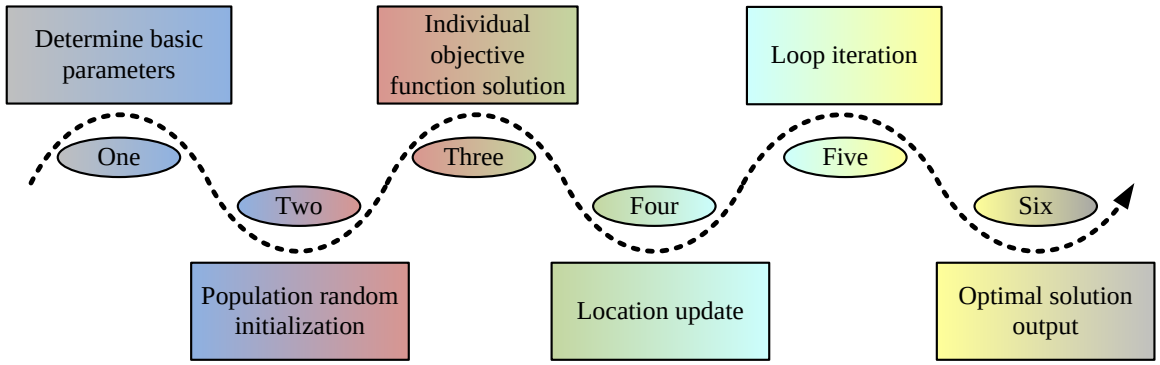


Figure 2: Implementation process of WOA.

hunting of humpback whale populations in nature, is easy to operate and less prone to falling into local optimal solution traps. The implementation process of the WOA is shown in Figure 2.

In Figure 2, the WOA first sets the initial input parameters and determines the population range and iteration times during optimization. Subsequently, the population is initialized and the objective function is solved for the independent individuals of the population. Finally, the population location is updated and the results are output. The WOA for population initialization can be expressed mathematically, as shown in equation (5).

$$X_{i,j}(initial) = Lb_j + rand_{i,j}(0,1)\cdot(Ub_j - Lb_j) \quad (5)$$

In equation (5), Lb means the lower bound position of the search space in the WOA, Ub means the upper bound position of the search space in the algorithm, and $rand_{i,j}(0,1)$ represents a random number with a value range of $rand_{i,j}(0,1) \in [0,1]$. The process of updating individual positions using the WOA can be expressed mathematically, as shown in equation (6).

$$X(t+1) = X(t) - \vec{A}\cdot\left|\vec{C}\cdot\vec{X}^*(t) - \vec{X}(t)\right| \quad (6)$$

In equation (6), $\vec{X}(t)$ means the vector of the position of the individual whale, $\vec{X}^*(t)$ means the position vector found by the leader whale of the whale population, and $\vec{A}$ and $\vec{C}$ denote the coefficient vectors. The WOA simulates the spiral movement of whales during hunting to complete the pursuit of prey, and its mathematical expression is shown in equation (7).

$$X(t+1) = D'e^{bl}\cos(2\pi l) + X^*(t) \quad (7)$$

In equation (7), D' means the distance between the individual whale and its prey, and b represents the constant coefficient of the whale's spiral motion. The ILC algorithm improved by the WOA can comprehensively control the system and effectively process the system control data. Therefore, the study utilized the WOA-ILC algorithm to construct a smart home human-machine control system and established a smart home human-machine control model. The specific structure is shown in Figure 3.

Figure 3 shows the specific implementation architecture of WOA-ILC in smart homes, whose core is to convert algorithm outputs into physical operation instructions for heterogeneous devices through a unified control bus. This architecture mainly consists of three parts: data acquisition module, controller based on WOA-ILC algorithm, and smart home devices. The data acquisition module is responsible for collecting various data in the smart home environment, such as indoor temperature, humidity, light intensity, etc., and sending the collected data to the WOA-ILC controller. The controller based on the WOA-ILC algorithm calculates the optimal control input and uses the optimized control signal to regulate the operating status of smart home devices.

Smart home devices adjust their operating status based on the output signal of the WOA-ILC controller to achieve control over the smart home environment. The

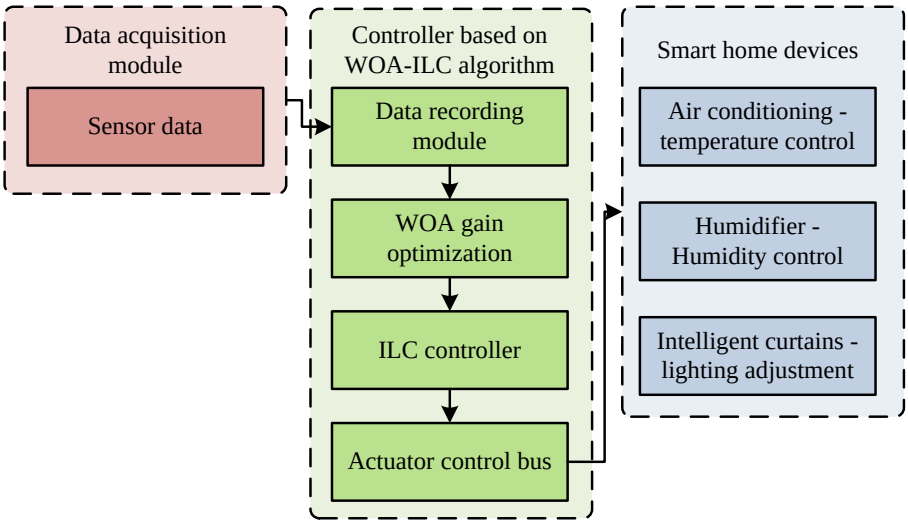


Figure 3: Smart home human-machine control system model.

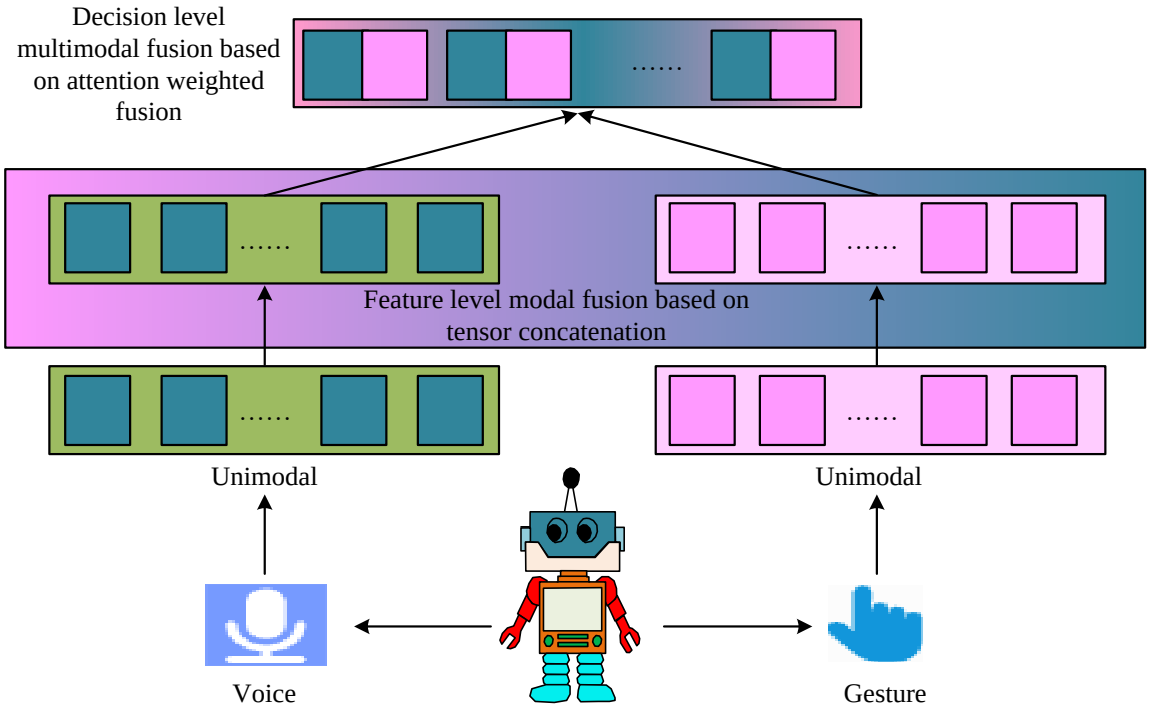


Figure 4: MF structure diagram.

controller is the core component of smart home devices, responsible for centralized management and coordination of various smart terminal devices, achieving automation and intelligent control of the home environment. Specifically, the integration mechanism of WOA and ILC is as follows: first, the parameters of the ILC algorithm and the WOA are initialized. Secondly, the WOA dynamically adjusts the learning gain in the ILC algorithm, namely K_P, K_I, and K_D, through its population search capability. In each iteration, the WOA updates the learning gain based on the current population position and objective function. Then the control input changes in the ILC algorithm are updated and finally the above process is repeated to guide the realization of a predetermined number of iterations or error convergence to a satisfactory value. Based on the above content, it can be seen that the design of the WOA-ILC algorithm mainly focuses on unimodal control performance, such as temperature control and humidity control. When voice commands and gesture commands are input simultaneously, the single error signal of this method cannot distinguish the modal source, resulting in control conflicts.

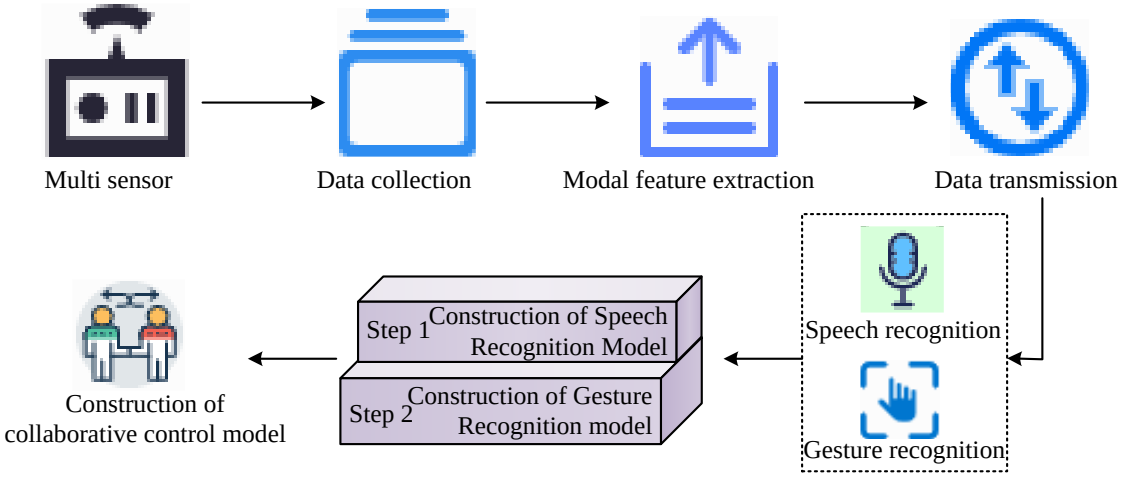


Figure 5: Optimization process of multimodal smart home collaborative control model.

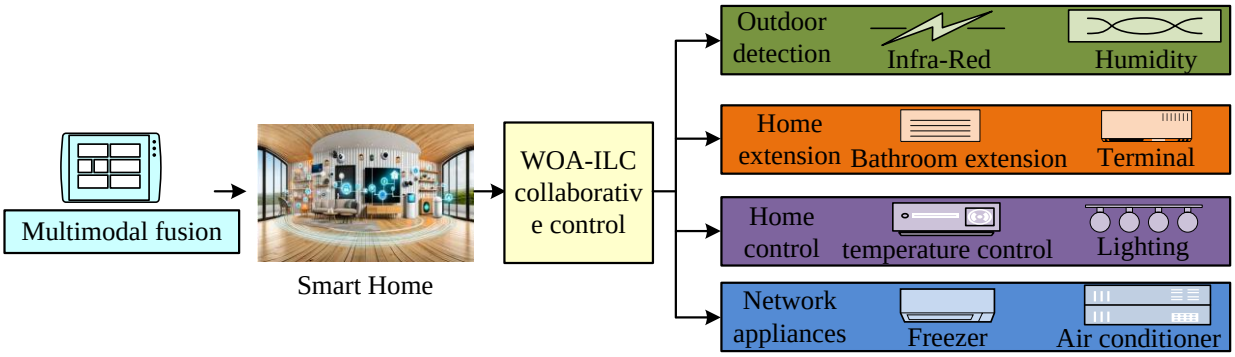


Figure 6: A human-machine collaborative control model combining MF.

3.2 Optimization of smart home collaborative control model combining MF

The smart home collaborative control model constructed using the WOA-ILC algorithm can ensure the intelligent control of the home system and is easy to operate. However, the control model constructed by the WOA-ILC algorithm has poor control effect on multiple modalities. Therefore, the study introduced MF to optimize the smart home collaborative control model. MF technology can fuse data through multiple sensors and eliminate the repulsion between different data modalities, thereby achieving data fusion. Its structure is shown in Figure 4.

In Figure 4, MF combines different single modalities and integrates data between them, mainly focusing on the fusion of features between speech and text data. After the feature extraction of speech and text data is completed, feature level fusion can be performed through tensor concatenation operation in equation (8), and then decision level MF can be performed using attention weighted fusion in equation (11). When performing MF, the weight tensor is used to linearly expand the modality, and the mathematical representation of vector generation is obtained using the input tensor, as shown in equation (8).

$$Z = \bigotimes_{m=1}^{M} z_m \tag{8}$$

In equation (8), Z represents the input tensor generated by MF. z_m represents the feature tensor of the m mode, and M represents the total number of modes. MF decomposes data weights into different modal factors, and its mathematical expression is shown in equation (9).

$$W = \sum_{k=1}^{R} \bigotimes_{m=1}^{M} (\phi_{m,k} \cdot W_{m,k}) \tag{9}$$

In equation (9), m represents the modal factor and R represents the minimum tensor rank. $\phi_{m,k}$ represents the contribution weight of the m modality in the k factor, and $W_{m,k}$ represents the k factor weight of the m modality. The optimization process of the smart home collaborative control model using multimodality is shown in Figure 5.

In Figure 5, MF technology mainly collects data through multiple data sensors and extracts features from different data modalities. After the feature extraction of different data modalities is completed, it can be processed through a collaborative control model, which can comprehensively process multiple inputs and output control decisions. In the smart home control model, data collection and transmission of speech recognition and gesture recognition are carried out through multiple sensors, and the data information of speech and gesture is preprocessed to reduce the loss of speech and gesture information, ensuring the integrity of speech and gesture information [18-20]. When collecting speech data

information, a speech recognition model is first established to preprocess the language information to improve the decoding efficiency of the decoder for language information. For language data collection in speech information, the probability of word sequence occurrence is calculated, and the calculation of word generation probability can be expressed mathematically, as shown in equation (10).

$$p(W) = p(w_1^K) = \prod_{k=1}^{K} p(w_k \left| w_1^{k-1} \right.) \tag{10}$$

In equation (10), K means the total number of words contained in the recognized speech information, and w_k represents the k th word string in the number of recognized words. The construction of a collaborative control model for smart home human-machine system using WOA-ILC combined with MF is shown in Figure 6.

As shown in Figure 6, firstly, the collected multimodal data is sent to the feature extraction module. Secondly, key features are extracted and transmitted to the MF module. The fused feature vectors are used for subsequent control or decision-making. Based on the fused feature data, the optimal control input is calculated using the WOA-ILC algorithm, and the optimized control signal is sent to the actuator. Finally, the operating status of the smart home device is adjusted based on the output signal of the controller. The MF for gesture and speech recognition can be expressed mathematically, as shown in equation (11).

$$h = Attention\left(Z_{voice}, Z_{gesture}\right) = \sum_{i=1}^{N} \alpha_i \cdot \left(W_v z_i^v \| W_g z_i^g\right) \tag{11}$$

In equation (11), Z_{voice} and $Z_{gesture}$ are the feature sequences of the speech modality and gesture modality, W_v and W_g are the projection matrices of the corresponding modalities, α_i represents the attention weight of the time step, h is the fused feature representation, and $\|$ represents vector concatenation. The data feature extraction of voice information and gesture recognition information can be represented by mathematical expressions, as shown in equation (12).

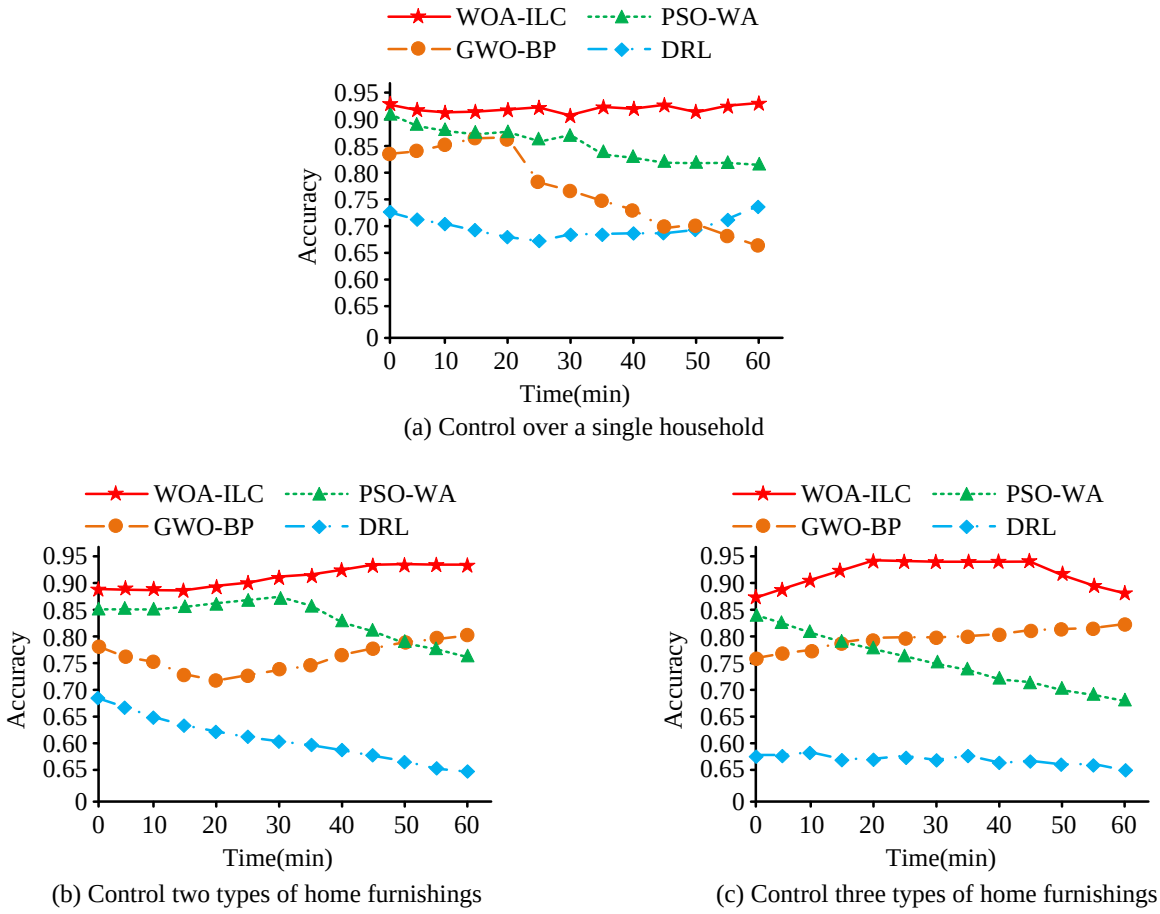
$$R_{ab}^{(n)} = \frac{\sum_{k=1}^{n} D_k^{(t)} S_k^{(t)} C_k^{(t)} Z_k^{(t)}}{\sum_{k=1}^{n} a_{rk}^{(1)} b_{rk}^{(1)} z_{rk}^{(1)}} \tag{12}$$

In equation (12), $R_{ab}^{(t)}$ represents the correlation between voice nodes and gesture nodes at different times, and n represents the number of intermediate nodes between gesture nodes and voice nodes. $D_k^{(t)}$, $S_k^{(t)}$, $C_k^{(t)}$, and $Z_k^{(t)}$ represent the data matrix, state matrix, context matrix, and feature vector of the k feature at time t. $a_{rk}^{(1)}$ and $b_{rk}^{(1)}$ are the weights of the r speech node and gesture node on the k feature, respectively. $z_{rk}^{(1)}$ represents the feature value of the r node on the k feature.

4 Empirical analysis of integrating multimodal smart home control models

4.1 Performance validation of improved iterative control model

To validate the performance of the improved iterative control model, model control experiments were conducted in Matlab software using Windows 11 system, Intel i5-12600KF processor, and NVIDIA GTX1070 graphics card model. Different types of smart homes, including smart air conditioners, smart refrigerators, and smart washing machines, were collected from the network, and the collected data was divided into training data and testing data. The voice commands were sourced from home environment recordings with a sample size of 8000, including background noise. The gesture trajectories were sourced from infrared depth cameras with a sample size of 6500, including 10 control gestures. The device status was sourced from smart home device logs with a sample size of 12000, featuring real-time recording of temperature, humidity, and power consumption. The dataset was divided into a training set and a testing set according to 7:3. The collection and use of personal data complied with privacy protection regulations and ethical requirements. The experimental parameters are set as follows: the population size, helix coefficient, and maximum iteration number of the WOA algorithm were set to 50, 1.0, and 200, respectively. The initial learning gain, differential gain, integral gain, and sampling interval in the ILC algorithm were 0.5, 0.05, 0.01, and 10ms, respectively. The backbone network of the speech recognition model adopted a two-layer bidirectional long short-term memory network and a 1-layer convolutional neural network, with the former having a hidden layer of 256 and the latter having 3 * 3, 64 channels. The training was 50 cycles (early stop), Batch=32, and the optimizer was AdamW. The architecture of the gesture recognition model is as follows: with a 1D convolutional neural network as the backbone network, Kernel=5, Stride=2, 64 → 128 → 256 channels, combined with the max pooling layer. The optimizer selected stochastic gradient descent. To verify the effectiveness of the WOA-ILC model, it was experimentally compared with the Particle Swarm Optimization-Genetic Algorithm (PSO-WA) model, the Grey Wolf Optimization-Back Propagation Neural Network (GWO-BP) model, and the Deep Reinforcement Learning (DRL) model. The DRL strategy network is a 3-layer fully connected network, and the value network adopts a symmetrical structure with the strategy network. The training cycle is 50000 steps (early stop strategy), and its end-to-end optimization characteristics are suitable as a performance upper limit reference. The input layer, hidden layer, and output layer of GWO-BP



(a) Control over a single household

(b) Control two types of home furnishings

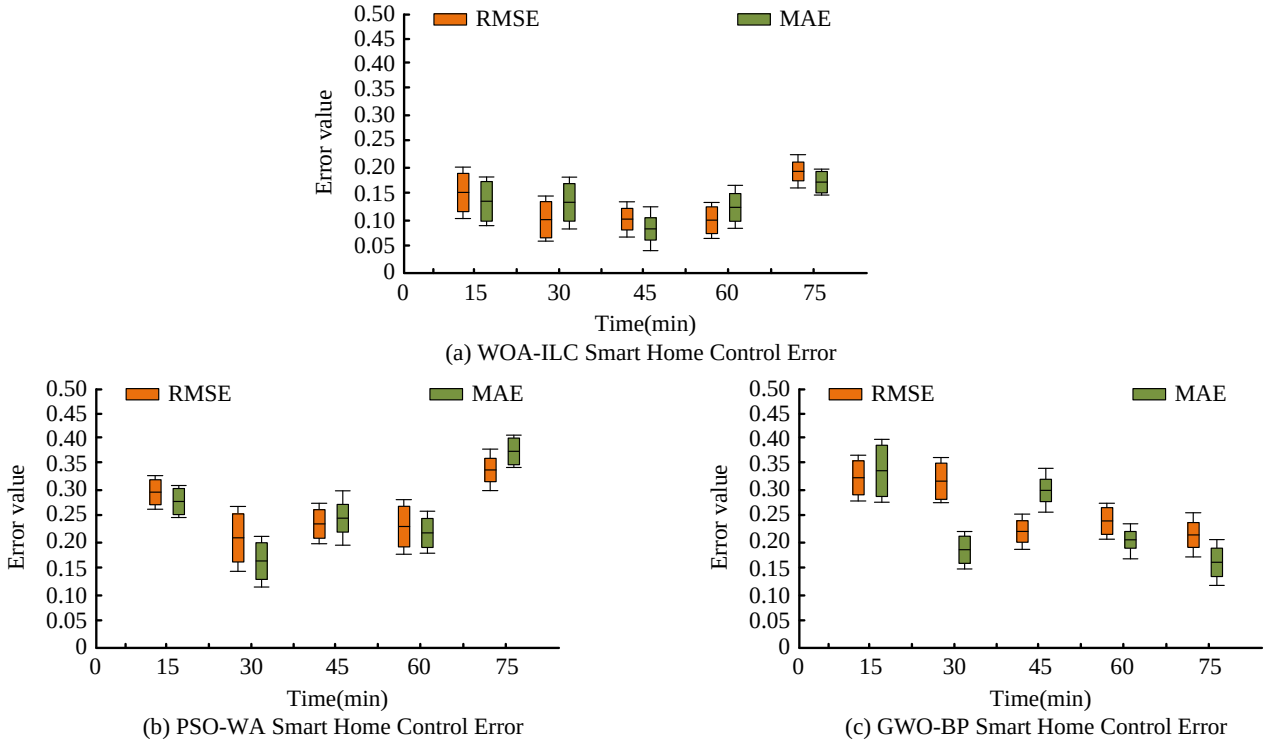
(c) Control three types of home furnishings

Figure 7: Comparison of control accuracy of different models.

were 8, 24, and 4, respectively. The population size and maximum iteration times were 50 and 200, respectively. Its interpretability and stability reflected the advantages and disadvantages of traditional methods. The crossover probability and mutation probability of the PSO-WA model were 0.85 and 0.02, respectively, making it suitable as a representative of metaheuristic algorithms. The comparison results are shown in Figure 7.

Figures 7 (a)-7 (c) correspond to the control accuracy tests of different models under single-device control, multi-device collaborative control, and noisy environments. According to Figure 7 (a), WOA-ILC had the highest control accuracy for smart homes, with an average control accuracy of 0.9212 and a maximum control accuracy of 0.9387. Moreover, the control accuracy for smart homes was not affected by changes in time span. The highest control accuracy of the PSO-WA model reached 0.8751, which was 0.0636 lower than the WOA-ILC model. From Figure 7 (b), the WOA-ILC model continuously improved its control accuracy for smart homes over time, with the highest control accuracy reaching 0.9412 and the average control accuracy being 0.9053. For the GWO-BP model, it could not perform well in controlling smart homes at the beginning, and over time, the control accuracy of smart homes slowly increased. Compared with the WOA-ILC model, the highest control progress was reduced by 0.1045. From Figure 7 (c), the control accuracy of the WOA-ILC model slowly increased with time and maintained a certain balance. Its average control accuracy was 0.8976, which was 0.2742 higher than that of the DRL model. The above outcomes denoted that the WOA-ILC model had higher precision in smart home control compared to other control models, and could perform high-precision control of smart homes well. The above results may be due to the fact that the WOA can dynamically adjust the search range and direction during the iteration process, which can effectively avoid getting stuck in local optima. To verify the control accuracy error of the WOA-ILC model on smart homes, RMSE and Mean Absolute Error (MAE) were used as experimental indicators. Different models were compared using the same dataset, and the experimental outcomes are indicated in Figure 8.

Figures 8 (a)-8 (c) correspond to the control error results of the WOA-ILC model, PSO-WA model, and GWO-BP model. From Figure 8 (a), the highest MAE value of the WOA-ILC model was 0.167, and the highest RMSE was 0.196. At the beginning of smart home control, the error value generated by controlling the home was relatively small, and over time, the control error of the smart home gradually decreased. When the control time reached 45 minutes, the MAE value for smart home control was the lowest, which was 0.094, and the lowest RMSE was 0.112. According to Figure 8 (b), the highest MAE of the PSO-WA model was 0.373, and the highest RMSE was 0.338. Overall, as time goes

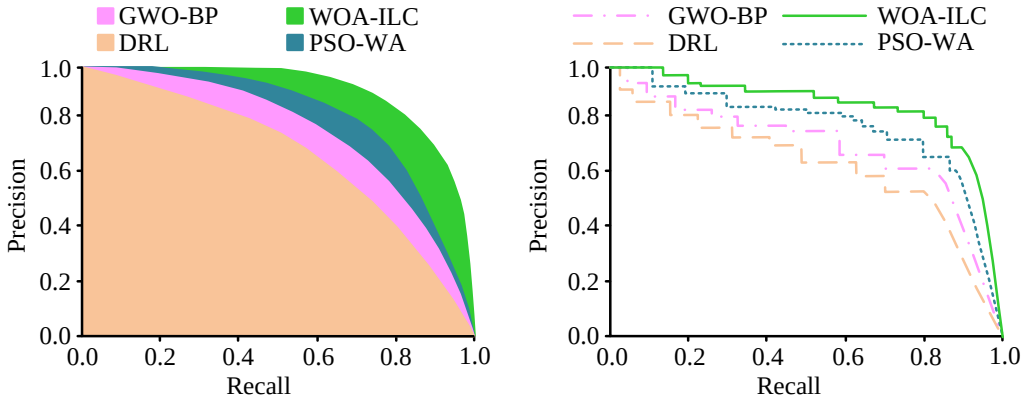


(a) WOA-ILC Smart Home Control Error

(b) PSO-WA Smart Home Control Error

(c) GWO-BP Smart Home Control Error

Figure 8: Comparison of control errors between different models.



(a) Control a small number of households

(b) Control large quantities of home furnishings

Figure 9: Comparison of PR curves for different control models.

on, the error value of the PSO-WA control model fluctuated to some extent and reached its minimum value at 30 minutes before increasing again. When the control time was 30 minutes, the RMSE reached a minimum of 0.216, which was 0.104 higher than the WOA-ILC model. The minimum MAE was 0.151, which was 0.057 higher than the WOA-ILC model. From Figure 8 (c), the control error of the GWO-BP model gradually decreased with time. The highest MAE was 0.337, which was 0.17 higher than the WOA-ILC model. The highest RMSE was 0.324, which was 0.128 higher than the WOA-ILC model. The above results indicated that the WOA-ILC model had a small control error for smart homes and performed well in controlling smart homes. This is because WOA can dynamically adjust the learning gain based on the current population position and objective function, so that it can better approximate the ideal output in each iteration. This dynamic adjustment mechanism significantly improves the convergence speed and control accuracy of the system. To further validate the control performance of the control model, an experimental comparison was conducted on its precision-recall (PR) curve, and the results are shown in Figure 9.

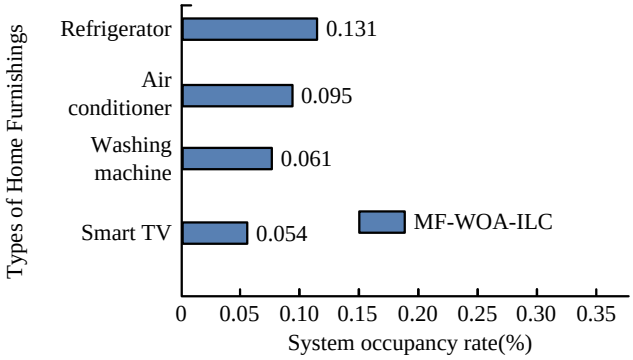
Comparison of PR curves for different methods in the cases of few-batch and multi-batch corresponds to Figure 9 (a) and Figure 9 (b). According to Figure 9 (a), when the WOA-ILC model was used to control small batch smart homes, the PR curve area was 0.9758, which was higher than the other three control models. The offline area of the PR curve of the PSO-WA control model was 0.9213, which was 0.0545 lower than that of the WOA-ILC model. Compared with the DRL model, the PR curve area of the WOA-ILC model increased by 0.2471. From Figure 9 (b), when performing multi-batch

Table 2: Statistical test results.

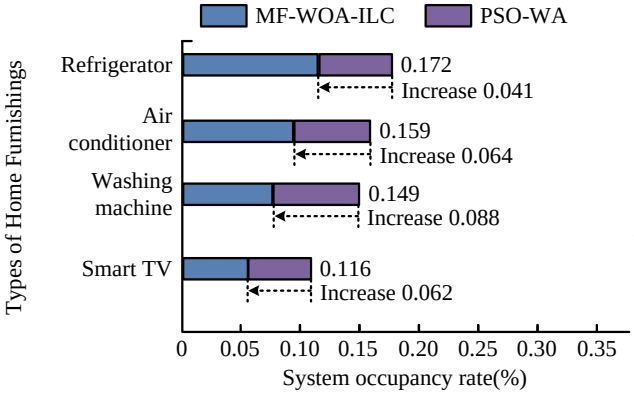
Comparison group	MF-WOA-ILC vs PSO-GA	MF-WOA-ILC vs DRL	MF-WOA-ILC vs GWO-BP
Experimental group mean	0.9387	0.9387	0.9387
Control group mean	0.8751	0.8234	0.8346
t-values	7.82	9.15	8.37
p-value	<0.001	<0.001	<0.001
Cohen's d	1.24	1.53	1.41

Table 3: Actual deployment of equipment.

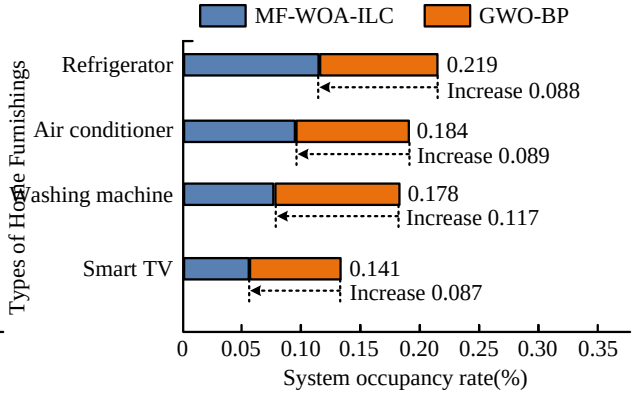
Types of home furnishings	Number of data	Model	Brand
Smart refrigerator	50	XQS70-128	Meiling refrigerator
Air conditioning	50	KFR-50GW/N1A1	Xiaomi air conditioner
Washing machine	50	EG100MATESL6	Haier washing machine
Smart TV	50	KD-75X9000F	Sony TV



(a) WOA-ILC control model occupancy rate



(b) PSO-WA control model occupancy rate



(c) GWO-BP control model occupancy rate

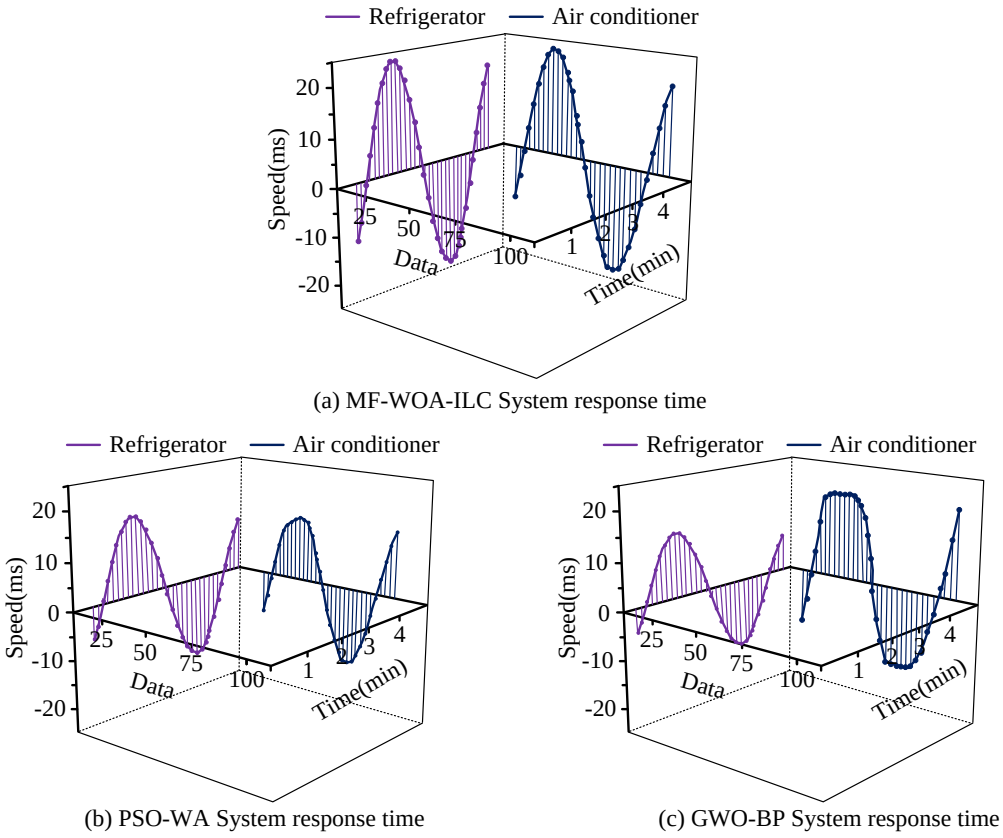
Figure 10: Comparison of system occupancy between different control models.

smart home control, the PR curve area of the WOA-ILC model decreased, but it was still significantly higher than the other three models at 0.9645, which was 0.0753 higher than the PSO-WA model and 0.1368 higher than the GWO-BP model. To further quantify the performance of the comparative methods, statistical tests were conducted, and the results are shown in Table 2.

According to Table 2, the experimental group showed a statistically significant advantage ($p<0.001$) in all comparisons, and an effect size exceeding 1.2 indicated that the differences in research methods were of practical significance. The F-value of the three groups' repeated cross validation was 46.37, ($p<0.001$).

4.2 Practical effect analysis of MF smart home collaborative control model

To analyze the practical effect of WOA-ILC smart home control combined with MF, different types of smart home control modes, including speech recognition and gesture recognition, were collected from different households for experimental verification. 200 sets of data were collected from the network. Due to the low complexity of research methods, there is no need for a large amount of training data to learn model parameters. In this case, less training data may already be sufficient, while more testing data can better evaluate the model's generalization ability. At the same time, to ensure that the testing set has sufficient sample size to evaluate the model's performance, 50 sets



(a) MF-WOA-ILC System response time

(b) PSO-WA System response time

(c) GWO-BP System response time

Figure 11: Comparison of response times of different models.

of data were used as the training set and 150 sets of data were used as the testing set. The actual deployed equipment is shown in Table 3.

To verify the system occupancy rate of the WOA-ILC control model combined with MF, different home control system occupancy rate experiments were compared with the PSO-WA model and GWO-BP model. The experimental results are shown in Figure 10.

According to Figure 10 (a), when the MF-WOA-ILC control model was used for intelligent refrigerator control, the system occupancy rate was 0.131%, the system occupancy rate for smart air conditioner control was 0.095%, the system occupancy rate for smart washing machine control was 0.061, and the system occupancy rate for smart air conditioner was 0.054. From Figure 10 (b), when the PSO-WA model was used for intelligent refrigerator control, the system occupancy rate reached 0.72%, which was 0.041% higher than the occupancy rate of the MF-WOA-ILC model. The system occupancy rate during smart TV control was 0.116, which was 0.088 higher than the MF-WOA-ILC model. The system occupancy rates of smart air conditioner and smart washing machine reached 0.159% and 0.149% respectively. From Figure 10 (c), the GWO-BP model achieved occupancy rates of 0.184%, 0.178%, 0.219%, and 0.141% for the four types of smart home control systems, respectively. Compared with the MF-WOA-ILC control model, the system occupancy rate for smart air conditioner control increased by 0.088%, and the system occupancy rate for smart washing machine control increased by 0.117. This is because MF technology can more comprehensively reflect the user's intention and environmental status by fusing data from different modalities, thereby improving the control accuracy of the system. Simultaneously, through tensor concatenation and attention weighted fusion, the features of speech and gesture modalities are effectively integrated. This can preserve more useful information, reduce information loss, and thus improve the quality of data representation. To verify the system response time of the MF-WOA-ILC HMCC model, it was experimentally compared with the PSO-WA model and GWO-BP model, and the results are shown in Figure 11.

In Figure 11 (a), the MF-WOA-ILC control model required less response time and had a fast response speed. Under 25 sets of data conditions, the average response speed was 10.31ms. As the data increased, the response time decreased and the response speed slowly accelerated. When the number of data reached 50, the average response speed reached a maximum of 24.12ms. The PSO-WA model necessitated a longer system reaction time than the suggested model, as seen in Figure 11(b). Its average response speed, when tested on 25 data sets, was 5.35 ms, 4.96 ms faster than the MF-WOA-ILC model. Its average response time, when 50 sets of data were included, was 18.89 ms, 5.23 ms faster than the MF-WOA-ILC model. The GWO-BP model had average response speeds of 4.48 ms and 16.21 ms, respectively, when there were 25 and 50 sets of data. This was 5.83 ms and 7.91 ms faster than the MF-WOA-ILC model, as shown in Figure 11 (c). To test the effectiveness of different parts of the MF-WOA-ILC model in practical

Table 4: The effectiveness of different parts of the MF-WOA-ILC model in practical applications.

Evaluation	Single voice	Single gesture	MF-WOA-ILC model
Inference time/ms	6.2	3.5	2.1
Memory usage/%	58.3	42.1	16.4
CPU utilization rate/%	12.7	8.9	5.2

applications, the study introduced inference time, memory usage, and CPU utilization for evaluation. The results are shown in Table 4.

Table 4 shows that the speech recognition module had the highest time consumption, at 6.2ms, mainly due to its bidirectional LSTM network needing to process temporal signals. The gesture recognition module has reduced its time consumption by 44%. After being processed by MF technology, the time consumption was only 2.1ms. In the fusion stage, WOA's population parallel computing was used to achieve an ultra-low occupancy of 5.2%. In addition, the research method could increase market share by 8% in terms of energy-saving benefits, maintenance costs, and annual single household benefits retained by users, which were 320-450%, 150-200%, and 12%-15%, respectively.

4.3 Discussion

The MF-WOA-ILC model was designed to address the challenges of HMCC in smart home environments. The experimental results showed that, firstly in terms of accuracy, the research model achieved high control accuracy in smart home control, with an average control accuracy of 0.9212 and a maximum control accuracy of 0.9387. Meanwhile, its maximum RMSE was 0.196 and maximum mean absolute error was 0.167, significantly better than the RMSE of 0.468 in reference [15]. This indicates that the research method can achieve more accurate HMCC when dealing with smart home control tasks in complex dynamic environments.

Secondly, in terms of controlling latency, the MF-WOA-ILC model had an average response time of 10.31ms under 25 data conditions and a response time of 24.12ms under 50 data conditions. This fast response capability enabled it to better adapt to tasks in smart home environments that require high real-time performance, such as real-time control and feedback of smart devices. Most existing literature has not quantitatively analyzed control delay.

In terms of system occupancy rate, the system occupancy rate in reference [8] was 1.25%. The MF-WOA-ILC model had significantly lower system occupancy rates in different smart home device controls, which were 0.131% for smart refrigerators, 0.095% for smart air conditioners, 0.061% for smart washing machines, and 0.054% for smart TVs. This indicates that the method proposed in this article has significant advantages in resource utilization efficiency, can effectively reduce system resource consumption, and is suitable for application on embedded devices with limited resources.

Finally, in terms of the scope of multimodal integration, the research results of references [12] - [16] focused on a single application scenario and did not involve the application scenario of collaborative control. However, the MF-WOA-ILC model could significantly improve the flexibility and adaptability of smart home control. In summary, the research methods have significant advantages in various aspects, providing new ideas and technical support for the research and application in this field.

5 Conclusion

Aiming at the problem that traditional smart home control methods cannot achieve good HMCC, a method was proposed to improve the construction of a smart home HMCC model by combining ILC with MF. ILC was improved to determine the input and output of the control system, and errors in home data were iteratively reduced, and finally MF was combined to fuse speech and gesture modalities for multimodal HMCC. The experimental results showed that the MF-WOA-ILC model had a maximum MAE of 0.167 and a maximum RMSE of 0.196 when performing smart home collaborative control, with a control accuracy of 0.9387 and an average control accuracy of 0.9212. In practical applications, the proposed control model had a system occupancy rate of 0.131%, demonstrating high accuracy and good control performance. In summary, the proposed model could effectively carry out HMCC of smart homes, with high control accuracy. However, there are still certain limitations in the research, as the research method only integrates speech and gesture modalities, which limits the system's adaptability in multiple scenarios. Therefore, in future research, facial expressions, EEG signals, eye tracking and other modalities can be introduced to enhance the robustness of the system.

References

[1] Guanglong Du, Linlin Zhang, Kang Su, Xueqian Wang, Shaohua Teng, and Peter X. Liu. A multimodal fusion fatigue driving detection method based on heart rate and PERCLOS. IEEE Transactions on Intelligent Transportation Systems, 23(11):21810-21820, 2022. https://doi.org/10.1109/TITS.2022.3176973

[2] Long Jin, Xin Zheng, and Xin Luo. Neural dynamics for distributed collaborative control of manipulators with time delays. IEEE/CAA Journal of Automatica Sinica, 9(5):854-863, 2022. https://doi.org/10.1109/JAS.2022.105446

[3] Yong Xu, and Zhengguang Wu. Data-based collaborative learning for multiagent systems under distributed denial-of-service attacks. IEEE Transactions on Cognitive and Developmental Systems, 16(1):75-85, 2022. https://doi.org/10.1109/TCDS.2022.3172937

[4] Genfeng Liu, and Zhongsheng Hou. Adaptive iterative learning fault-tolerant control for state constrained nonlinear systems with randomly varying iteration lengths. IEEE Transactions on Neural Networks and Learning Systems, 35(2):1735-1749, 2022. https://doi.org/10.1109/TNNLS.2022.3185080

[5] Ahmed S. Elkorany, Mohamed Marey, Khaled M. Almustafa, and Zeinab F. Elsharkawy. Breast cancer diagnosis using support vector machines optimized by whale optimization and dragonfly algorithms. IEEE Access, 10(3):69688-69699, 2022. https://doi.org/10.1109/ACCESS.2022.3186021

[6] Yu Liu, Yu Shi, Fuhao Mu, Juan Cheng, Chang Li, and Xun Chen. Multimodal MRI volumetric data fusion with convolutional neural networks. IEEE Transactions on Instrumentation and Measurement, 71:1-15, 2022. https://doi.org/10.1109/TIM.2022.3184360

[7] Mahlous A. Threat model and risk management for a smart home IoT system. Informatica, 47(1):51-63, 2023. https://doi.org/10.31449/inf.v47i1.4526

[8] Zhekang Dong, Xiaoyue Ji, Guangdong Zhou, Mingyu Gao, and Donglian Qi. Multimodal neuromorphic sensory-processing system with memristor circuits for smart home applications. IEEE Transactions on Industry Applications, 59(1):47-58, 2022. https://doi.org/10.1109/TIA.2022.3188749

[9] Guixi Wei, Ming Chi, Zhiwei Liu, Mingfeng Ge, Chaojie Li, and Xianggang Liu. Deep reinforcement learning for real-time energy management in smart home. IEEE Systems Journal, 17(2):2489-2499, 2023. https://doi.org/10.1109/JSYST.2023.3247592

[10] Safwa Ameer, James Benson, and Ravi Sandhu. Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT. IEEE Transactions on Dependable and Secure Computing, 20(5):4032-4051, 2022. https://doi.org/10.1109/TDSC.2022.3216297

[11] Thinagaran Perumal, E. Ramanujam, Sukhavasi Suman, Abhishek Sharma, and Harshit Singhal. Internet of Things centric-based multiactivity recognition in smart home environment. IEEE Internet of Things Journal, 10(2):1724-1732, 2022. https://doi.org/10.1109/JIOT.2022.3209970

[12] Wujie Zhou, Shaohua Dong, Jingsheng Lei, and Lu Yu. MTANet: Multitask-aware network with hierarchical multimodal fusion for RGB-T urban scene understanding. IEEE Transactions on Intelligent Vehicles, 8(1):48-58, 2022. https://doi.org/10.1109/TIV.2022.3164899

[13] Tao Chen, Richang Hong, Yanrong Guo, Shijie Hao, and Bin Hu. MS ^{2}-GNN: Exploring GNN-based multimodal fusion network for depression detection. IEEE Transactions on Cybernetics, 53(12):7749-7759, 2022. https://doi.org/10.1109/TCYB.2022.3197127

[14] Houhong Lu, Yangyang Zhu, Ming Yin, Guofu Yin, and Luofeng Xie. Multimodal fusion convolutional neural network with cross-attention mechanism for internal defect detection of magnetic tile. IEEE Access, 10(4):60876-60886, 2022. https://doi.org/10.1109/ACCESS.2022.3180725

[15] Fang M, Peng S, Liang Y, Hung C C, and Liu S. A multimodal fusion model with multi-level attention mechanism for depression detection. Biomedical Signal Processing and Control, 82(3):561-573, 2023. https://doi.org/10.1016/j.bspc.2022.104561

[16] Cherifi Dalila, El Affifi Omar Badis, Boushaba Saddek, and Amine Naït-ali. Feature Level Fusion of Face and voice Biometrics systems using Artificial Neural Network for personal recognition. Informatica, 44(1):85-96, 2020. https://doi.org/10.31449/inf.v44i1.2596

[17] Chi R, Lv Y, and Huang B. Distributed iterative learning temperature control for multi-zone HVAC system - ScienceDirect. Journal of the Franklin Institute, 357(2):810-831, 2020.

[18] Hsien-Pin Hsu, and Chia-Nan Wang. Hybridizing whale optimization algorithm with particle swarm optimization for scheduling a dual-command storage/retrieval machine. IEEE Access, 11(4):21264-21282, 2023. https://doi.org/10.1109/ACCESS.2023.3246518

[19] Ankita Gandhi, Kinjal Adhvaryu, Soujanya Poria, Erik Cambria, and Amir Hussain. Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions. Information Fusion, 91(5):424-444, 2023. https://doi.org/10.1016/j.inffus.2022.09.025

[20] Can Cui, Haichun Yang, Yaohong Wang, Shilin Zhao, Zuhayr Asad, Lori A Coburn, Keith T Wilson, Bennett A Landman, and Yuankai Huo. Deep multimodal fusion of image and non-image data in disease diagnosis and prognosis: A review. Progress in Biomedical Engineering, 5(2):201-213, 2023. https://doi.org/10.1088/2516-1091/acc2fe

Hybrid Multi-layer Perceptron and Metaheuristic Optimizers for Indoor Localization Error Estimation

Jia Liu[1], Jian Sun[2,*]
[1]CangZhou Kindergarten Teachers College, Cangzhou 061001, China
[2]Hebei Software Institute, Baoding 071000, China
E-mail: sunjian8035@163.com
[*]Corresponding author

Keywords: average localization error, pelican optimizer algorithm, global situational system, gold rush optimizer, multi-layer perceptron regression

Indoor localization is hindered by GPS signal weakening in indoor environments. This research formulates machine learning with Multi-layer Perceptron Regression (MLPR) algorithm supported by two metaheuristic optimizers, namely, Gold Rush Optimizer (GRO) and Pelican Optimizer (POA), to yield hybrid models MLGR and MLPO to forecast Average Localization Error (ALE). The dataset organized in a structured form was of size 107 samples with six significant features as follows: anchor ratio, transmission range, node density, trainings, standard deviation of ALE, and ALE as objective. The dataset was split into training (70%), validation (15%), and testing (15%) subsets. Experimental analysis in three prediction layers reveals that MLGR outperformed MLPO and MLPR models in every prediction layer. MLGR exhibited maximum performance at the third test layer with an RMSE of 0.036 and R^2 of 0.993, whereas MLPO and MLPR attained RMSE of 0.059 and 0.080 and values of R^2 of 0.981 and 0.966, respectively. The findings establish the validity of the introduced hybrid optimization technique to increase accuracy and convergence rate of prediction of ALE in wireless sensor networks.

Povzetek: Raziskava uvaja hibrid MLPR z optimizatorjema GRO in POA za napoved ALE v WSN. Model MLGR doseže najboljšo točnost (RMSE 0.036, R^2 0.993), presega MLPO in osnovni MLPR.

1 Introduction

A Wireless Sensor Network (WSN) is designed by several cheap, tiny sensors located all around an area to gauge various physical characteristics or monitor environmental situations [1,2]. Target tracking and precision agriculture are only two of the many industries where this technology finds applicable uses [3,4]. These sensors are often required to make accurate coordinate predictions while utilizing the minimum resources possible [5,6]. Although sensors equipped with an incorporated GPS can quickly determine their situations, cost and size constraints preclude embedding GPS in each sensor. Employing localization frameworks is one alternative strategy, whereby several anchor nodes equipped with incorporated GPS help the hidden nodes to identify their coordinates accurately [7,8].

Numerous localization frameworks have been developed to address diverse problems [9]. These frameworks should be flexible to operate well in different indoor and outdoor structures and topologies. These localization strategies fall into two groups: range-free and range-based frameworks. When using range-based strategies, the unfamiliar nodes' situation is identified by measuring the length between the anchor and unfamiliar sensor nodes [10,11]. Diverse factors are employed, including the Received Signal Strength Indication (RSSI), the angle, and time of arrival [12].

Conversely, range-free methods such as centroid [13] and ad-hoc situationing systems[14] use simple connectivity-related operations to locate hidden nodes. All these frameworks need is for the anchor node's beacon signal to be present in the medium. Range-based frameworks are preferred and used more frequently than range-free frameworks [15–17]. Several bioinspired frameworks have been recommended for the range-based method to help design a less complex algorithm [18]. A node localization technique utilizing Particle Swarm Optimization (PSO) [19] was first described by Gopakumar and Jacob [20], mimicking the behavior of a swarm of fish searching for food. While the framework first yielded promising results, it often became stuck in local optima, leading to premature convergence. Goyal and Patterh [21] showed significant results in reducing localization errors in 2014 when they used Cuckoo Search (CS) for node localization in WSNs. The CS algorithm's tuning settings, which simplified the computation procedure, are responsible for this result. Researchers recently recommended a modification to the CS technique: Cheng and Xia [22], which improved the traditional CS algorithm's rate of convergence. They changed the mutation probability and random walk stage size to improve the search strategy [23,24].

A Bayesian approach for node localization in WSNs was introduced by Morelande et al. [25]. The suggested algorithm, known as progressive correction, improved their previous work [26]. Both strategies are compared

under different situations, with the Cramer-Rao bound (CRB) as the benchmark. Outperforming its predecessor, the newly recommended algorithm achieves superior accuracy. Furthermore, Ghargan et al. [27] recommended a tactic in which three enhancement frameworks, PSO, Backtracking Search Algorithm (BSA), and Gravitational Search Algorithm (GSA), are individually hybridized with ANN. Outperforming alternative methods, the GSA-ANN hybrid achieved low mean absolute distance projection errors: 0.02 meters for outdoor and 0.2 meters for indoor cases. In a current survey by Ahmadi and Bouallegue [28], various innovative ML strategies employed in node localization within WSNs were compiled. The study included a comparison of cumulative localization error distribution curves for different strategies such as ANN, SVM, DT, and Naive Bayes (NB). Analysis of cumulative localization error distributions revealed NB as the superior machine learning technique among those evaluated [29–31].

Bhatti et al. [32] recommended an outlier recognition framework for indoor localization known as iF_Ensemble. This technique utilizes supervised ML methods, RF, and SVM-along with the unsupervised Isolation Forest, and an ensemble technique called stacking. It significantly outperforms the model with a very high localization accuracy of 97.8%.

Wang et al. [33] recommended a new node localization framework called KELM-HQ, which uses Kernel Extreme Learning Machines to boost the accuracy of estimating the situations of hidden nodes. It achieves a 34.6% improvement in localization error compared with fast-SVM, 19.2% with the GADV-Hop algorithm, and 11.9% with the DV-Hop-ELM algorithm. Table 1 represents the comparative summary of state-of-the-Art for indoor localization error prediction.

Table 1: Comparative summary of state-of-the-Art for indoor localization error prediction.

Method	Reference	Technique	RMSE	MAE	R^2	Limitations and Improvements
PSO-ANN	Gharghan et al. (2016) [34]	Particle Swarm Optimization with ANN	0.2 (indoor)	0.12	N/A	Converges to local minima; lacks robustness under varying topologies. MLGR offers more stable convergence and better generalization.
Cuckoo Search	Goyal & Patterh (2014) [35]	Cuckoo Search Metaheuristic	0.42	N/A	N/A	Requires tuning-sensitive parameters; slower convergence. MLGR improves convergence speed and accuracy.
Bayesian Correction	Morelande et al. (2008) [36]	Progressive Bayesian Estimation	0.23	0.15	0.91	Not ML-based; lower adaptability in dynamic environments. MLGR leverages adaptive learning for real-time localization.
KELM-HQ	Wang et al. (2020) [33]	Kernel Extreme Learning Machine	0.58	N/A	0.93	No optimizer hybridization; poorer error margin. MLGR offers up to 38% better RMSE.
MLPR	This Study (Baseline)	Multi-layer Perceptron Regression	0.080	0.071	0.966	Strong nonlinear modeling, but suboptimal without optimization.
MLPO	This Study	MLPR + Pelican Optimization	0.059	0.051	0.981	Improves ALE prediction; outperformed by MLGR in convergence and accuracy.
MLGR	This Study	MLPR + Gold Rush Optimization	0.036	0.029	0.993	Best overall accuracy and generalization. Efficiently addresses convergence and nonlinearity weaknesses of prior SOTA methods.

This paper uses machine learning to advance the scheme for ALE prediction. In this paper, the authors used MLPR. MLPR was chosen for its known capability to represent complex, nonlinear relationships between input values typical in wireless sensor network datasets. In contrast to linear or tree-based regressors, MLPR supports complex feature interactions natively and does not require explicit feature transformation. Its design is highly flexible and adaptable, reflecting the need to predict ALE in varied network topologies. Its efficiency in other similar application areas has been previously proved, providing both accuracy and computational expense as needed, making it an appropriate tool to be applied to this case. The use of MLPR to predict ALE has several advantages. This framework performs very well in detecting intricate relationships within data and, hence, is suitable for those applications where the underlying pattern is complex and nonlinear. It is believed that MLPR can learn with diverse traits and complex adjustments of the ALE prediction task. This is because such a type of neural network is pretty robust, providing good results on big data and allowing generalization for unseen examples. GRO and POA were recommended as additional optimizers in a quest for a better level of accuracy for ALE prediction. The results

and discussion also presented the hybridized framework's performance against others. Such a comparative study will go a long way in determining which model achieves better accuracy and precision in ALE forecasting. Consequently, the GRO inclusion, POA, and model hybridization in the present study assist in acquiring an in-depth understanding of the strengths and shortcomings of the adopted predictive frameworks.

2 Mathematical frameworks

2.1 MLPR

A concise overview of the single-layer perceptron, the fundamental building block of Multi-Layer Perceptrons (MLPs) [37], is presented to foster understanding. This most straightforward form of a neural network entails a single output neuron connected to all inputs. Each input connection has a corresponding weight, denoted as w_i (where i ranges from 0 to n, with n being the total number of inputs). These weights influence the output (y), wrepresentingthe projected binary class. The input values (x_i) correspond to the considered traits or variables [38]. Every input feature value is multiplied by its corresponding weight, $v_i a_i$, during the weighting stage. The second stage then involves adding together these products $(v_0 a_0 + v_1 a_1 + \cdots + v_n a_n)$. The third stage is the transfer stage, where an activation function is applied to the total. denoted as f (or transfer function). This process yields an output, y, expressed as:

$$y = f(z) \; and \; z = \sum_{i=0}^{n} v_i a_i \tag{1}$$

$a_0 = 1$ represents the output y, where a_0 is the threshold or bias.

Eq. (1) empowers a perceptron, suited for learning functions that a straight line can separate. When dealing with a two-dimensional input featuring two variables, this function is visually represented by a line [39]. The representation shifts to a plane in the context of three dimensions and three traits. However, when dealing with input in n dimensions, the corresponding function is symbolized by a hyperplane. The equation governing the hyperplane is articulated as:

$$\sum_{i=0}^{n} v_i a_i = 0 \tag{2}$$

Eq. (2) represents the dot product of weight vector V and input vector A.

$$V.A = 0 \tag{3}$$

The subsequent stage, referred to as the learning or training stage, is initiated upon obtaining the responses from the input training data. This stage aims to optimize the weights by minimizing a cost function. This function typically measures the squared error between the framework's projected outputs and the real responses. Analytical methods such as gradient descent can compute the optimal weight vector. Convergence of the framework to a solution is imperative for the network to become operational. It also checks the model's generalization capability by validating it with new data. An SLP architecture is created by connecting several perceptrons in parallel, which is used with multiple outputs.

The perceptron and single-layer perceptron have limited ability to solve nonlinearly separable problems. To remove this limitation, multiple layers can be added in series to develop an MLP network. In a neural network, the first and last layers are referred to as the input and output layers, respectively, while all other in-between layers are called the hidden layers. The output from each layer acts as the input for the subsequent layer, and this process goes on iteratively for every additional layer.

As defined in [40], MLP is a feedforward neural network architecture where information flows only in one direction from the input layer to the output layer via the hidden layers. Each link between neurons is weighted, and all neurons in one layer have the same activation function. The hidden layers use the sigmoid function for nonlinear activation, while the output layer could be either sigmoid or linear, depending on the application.

The performance of the MLP model depends on different factors, including the count and arrangement of hidden layers and nodes, the volume of training data, the selection of variables, etc. On the other hand, training parameters, such as the number of cycles, learning rate, momentum governing weight adjustments, and so on, may hugely impact the model's overall performance.

2.2 GRO

For the challenge of damage recognition in optimization, researchers implemented GRO. This framework boasts faster convergence compared to traditional methods. Unlike other strategies, GRO is inspired by human reasoning and decision-making. It begins by creating a set of randomly distributed "operators" within the search domain, mimicking the initial search stage in human problem-solving.

During each cycle, these operators move together. When one detects a significant change in a monitored parameter (analogous to increased sound), the entire group pauses to investigate. In addition to their observations, each operator also considers information from others, particularly those detecting stronger "signals." This collaborative approach guides the group towards the area with the most significant readings, ultimately leading to precisely identifying the damage location. Refer to Fig. 1 for the GRO algorithm's flowchart.

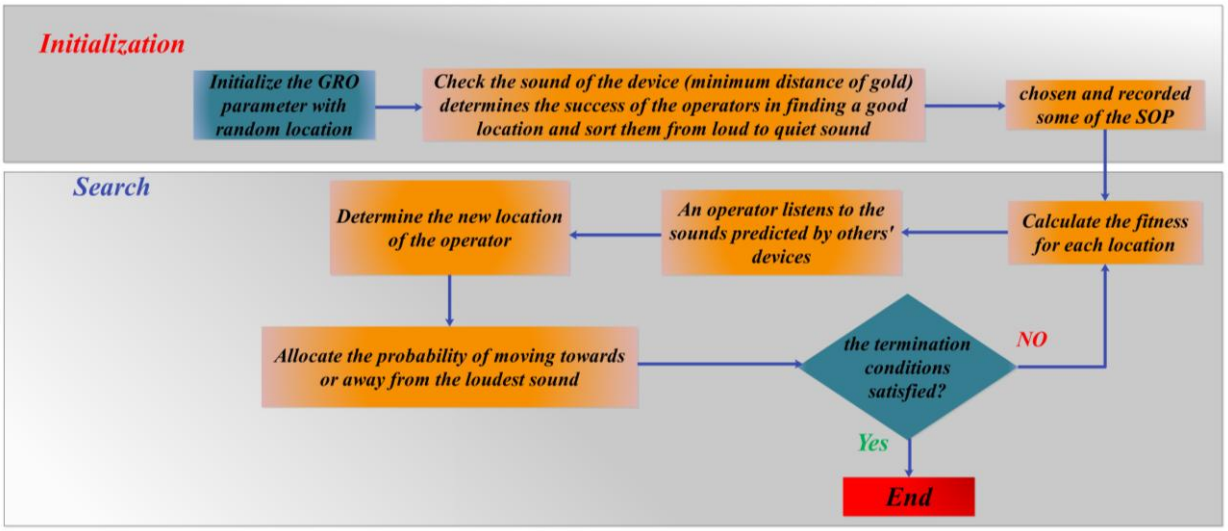


Figure 1: The flowchart of the GRO algorithm

First stage: initialization

Eq. (4) illustrates the random allocation of a place for each operator within the search domain. The variable Rand represents a random number within the interval of [0...1], and the search domains for the operators are delineated by ub_i and lb_i.

$$location_i^{(0)} = lb_i + (ub_i - lb_i) \times rand, \quad i = 1,2,\dots,n \tag{4}$$

Second stage: Monitoring involves choosing the best places.

The SOP is the identifier used to indicate that the operator has successfully identified the ideal location. At the end of every cycle, it is recommended that the top 10% of operators in the SOP be kept.

Third stage: The link between fitness and distance is called fitness-distance.

Eq. (5) is used to examine each operator's readings (rate). This calculation helps determine which operator possesses the most promising data for locating the damage.

$$rate(i) = \frac{D_i}{P_i} \times \frac{sound\ (highest\ volume) - sound\ (i)}{(sound\ (highest\ volume) - sound\ (lowestvolume) + \varepsilon)} \tag{5}$$

To reduce mistakes caused by the environment, the coefficients in Eqs. (6-7) are utilized, represented by $D_i$andP_i , respectively.

$$D_i = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \cdots} \tag{6}$$

$$P_i = 2 - \frac{ite}{max_{ite}} \tag{7}$$

During this stage, each operator makes unique decisions influenced by a combination of sounds, a process encapsulated by Eq. (8).

$$\begin{aligned} &fresh\ place\ (i) \\ &= place\ (i) + md \times [(rate(j) \\ &- rate(i)) \times (place\ (j) - place\ (i)) \\ &\times rand\ ] \end{aligned} \tag{8}$$

The md coefficients, delineating the path of movement, are expressed in Eq. (9):

$$md = \begin{cases} +1 \Longrightarrow towards\ a\ loudest\ sound & a > rand \\ -1 \Longrightarrow away\ from\ a\ loudest\ sound & a < rand \end{cases} \tag{9}$$

Fourth stage: location correction

If Eq. (8) is unable to yield a satisfactory location, then the coefficients β and γ are chosen from the interval $0 < \beta < \gamma < 1$, and Eq. (10) is employed to produce new locations.

$$new\ location\ (i) = \begin{cases} choose\ a\ neighboring\ location & rand < \beta \\ select\ a\ new\ location\ randomly & \beta < rand < \gamma \\ do\ not\ move & \gamma < rand \end{cases} \tag{10}$$

Fifth stage: termination

Iteratively performing stages 4 through 5 in a loop until one of the subsequent circumstances is satisfied:

1. There is no longer a peak count of strives permitted.
2. The ideal place doesn't alter.
3. A predefined anticipated threshold is attained When the disparity between the values of SOP and the global optimum is smaller than a given value.
4. Less than a certain degree of precision separates the objective values of the best and worst sites.

GRO is used in this research to tune the internal parameters of the MLPR model, i.e., the weights and biases, to minimize prediction error in terms of ALE. This optimizes convergence rate and model precision by effectively traversing the search space and escaping local minima, making the process of learning more robust.

2.3 POA

Dehghani and Trojovský initiated the POA in 2022, an inventive nature-inspired method that draws influence from pelicans' social behaviors and hunting strategies [41]. Pelicans are huge birds with strong beaks that use a wide neck pouch to catch and eat their prey. These birds comprise the targeted population and typically reside in large groups. Using the following equation, the population members are initialized at random:

$$x_{i,j} = l_j + rand.\left(u_j - l_j\right), \quad i = 1,2,3,\dots,N\ , j = 1,2,3,\dots,m \tag{11}$$

The value of the jth variable for the ith candidate solution is indicated by the formula $x_{i,j}$. The number of people in the group and the overall count of dilemma variables are indicated by the parameters N and m, respectively. In addition, $rand$ denotes a randomly generated number within [0.1], and l_j and u_j represent the problem variables' lower and upper restrictions. Eq. (12) is used to create the population matrix, which represents the individuals in the set of possible solutions:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} X_{1,1} & \dots & X_{1,j} & \dots & X_{1,m} \\ \vdots & & \vdots & & \vdots \\ X_{i,1} & \dots & X_{i,j} & \dots & X_{i,m} \\ \vdots & & \vdots & & \vdots \\ X_{N,1} & \dots & X_{N,j} & \dots & X_{N,m} \end{bmatrix}_{N\times m} \tag{12}$$

The objective function is derived from Eq. (13) as provided.

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times m} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{13}$$

The objective function vector, F, is made up of the objective function values that are allocated to each candidate solution; for example, F_i for the ith candidate solution is the objective function vector. There are two stages to a pelican's hunting process: exploration and exploitation. Pelicans travel forward searching for their prey in the exploration stage and glide smoothly across the water's surface in the exploitation phase. Pelicans locate the prey by using a randomly generated location to approach it during the first exploration stage. The POA's capacity for exploration is enhanced by this random component in the prey's location [42]. Eq. (14) illustrates the first stage's mathematical expression:

$$x_{i,j}^{p_1} = \begin{cases} x_{i,j} + rand.\left(p_j - I.x_{i,j}\right), & F_p < F_i; \\ x_{i,j} + rand.\left(x_{i,j} - p_j\right), & else, \end{cases} \tag{14}$$

Here, $x_{i,j}^{p_1}$ represent the ith pelican's renewed situation in the jth aspect after the primary stage. The update is dependent on two random variables: F_p, which represents the prey's objective function value, and P_j, which indicates the prey's situation in the jth aspect. A pelican's revised location is considered acceptable in the POA algorithm if it increases the value of the objective function at that specific spot. This process, known as efficient updating, prevents the framework from coinciding to suboptimal locations. This is depicted mathematically as:

$$x_i = \begin{cases} X_i^{p_1}, F_i^{p_1} < F_i \\ X_i \quad\quad else \end{cases} \tag{15}$$

The hunting action is represented mathematically as: The renewed situation of the ith pelican after the second stage is denoted by $X_i^{P_1}$, and the objective function value of the pelican obtained from this stage is indicated by $F_i^{P_1}$. In the second stage, pelicans tn the water's surface lift their wings upward to increase their chances of catching more fish [43]. As a result, they trap the meal in their throat pouches. This stage significantly improves the POA algorithm and makes it easier for improved solutions to converge within the hunting region.

$$x_{i,j}^{p_1} = x_{i,j} + R.\left(1 - \frac{t}{T}\right).(2.rand - 1).x_{i,j} \tag{16}$$

In the second stage, many parameters are taken into consideration to establish the renewed situation of the ith pelican in the jth aspect, which is represented as $X_{i,j}^{P_2}$. Among these is the constant R, which has been given a value of 0.2. The phrase $(1 - t/T)$, where t displays the cycle count and T displats the peak count of cycles, affects the neighborhood radius of $x_{i,j}$. At this point, an efficient updating procedure is also implemented, and Eq. (17) is used to determine whether to admit or deny the new pelican situation.

$$x_i = \begin{cases} X_i^{p_2}, F_i^{p_2} < F_i \\ X_i \quad\quad else \end{cases} \tag{17}$$

The renewed status of the ith pelican is represented by $X_i^{P_2}$, and the related objective function value for that pelican is represented by $F_i^{P_2}$. After every population member has been updated, the next cycle and the procedures listed in Eqs start. (14)-(17) are repeated until the execution process is finished [44]. The POA flowchart displayed in Fig. 2 illustrates the iterative nature of this procedure.

The parameters of the MLPR model, i.e., weights and biases, are adjusted using POA to minimize the prediction error of the ALE. During training, by simulating cooperative hunting behavior, POA speeds up the search process. This improves the overall generalization and stability, and ensures predictions are made with higher accuracy in varied network environments.

The hybrid optimization strategy improves localization accuracy by integrating MLPR's potential to capture nonlinear patterns along with GRO and PO's capabilities to perform global searching. These optimizers refine weight tuning as well as convergence, reducing errors during predictions. The synergy provides more accurate ALE estimations in different network scenarios.

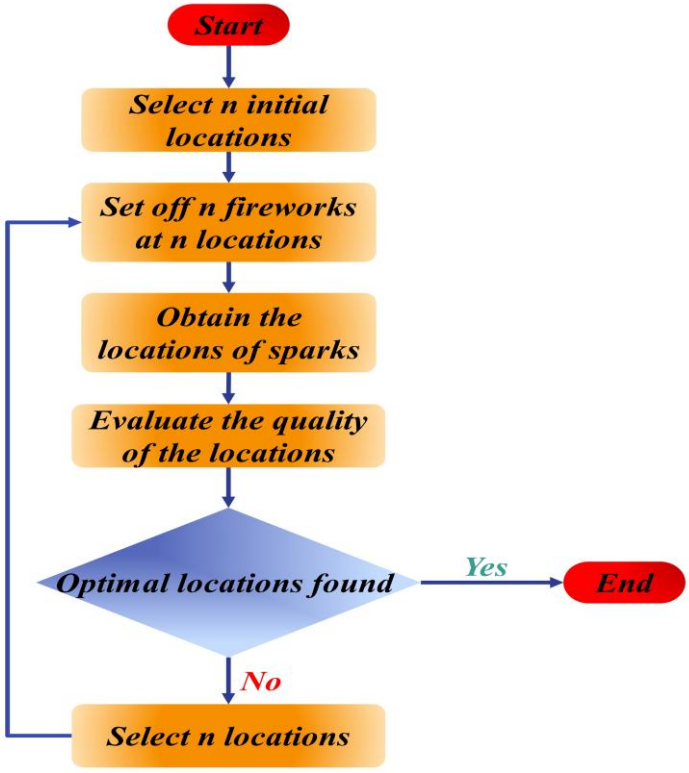


Figure 2: The flowchart of POA

2.4 Performance evaluation metrics

This part organizes diverse gauges to examine the compound frameworks' productivity by measuring their correlation and error levels. The metrics covered encompass RMSE, Index of agreement (IOA), MAE, Coefficient Correlation (R^2), and Mean Relative Absolute Error (MRAE). These chosen metrics, RMSE, MAE, IOA, R^2, and MRAE, provide a complete assessment of accuracy, consistency, and relative error of predictions. The absolute errors are measured by RMSE and MAE, explained variance by R^2, agreement by IOA, and proportional deviation by MRAE, balancing evaluation across model behaviors and sensitivity in errors. The formulas for each of these metrics are provided below.

$$MRAE = \frac{1}{n}\sum_{i=1}^{n}\frac{|m_j - b_j|}{|m_j - \overline{m}|} \tag{18}$$

$$IOA = 1 - \frac{\sum_{i=1}^{n}(x_{i,a} - x_{i,p})^2}{\sum_{i=1}^{n}(|x_{i,p} - \overline{x_a}| + |x_{i,a} - \overline{x_a}|)^2} \tag{19}$$

$$MARE = \frac{1}{N}\sum_{i=1}^{N}\frac{|e_i - m_i|}{m_i} \tag{20}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(m_i - b_i)^2} \tag{21}$$

$$R^2 = \left(\frac{\sum_{i=1}^{n}(b_i - \bar{b})(m_i - \overline{m})}{\sqrt{\left[\sum_{i=1}^{n}(b_i - \bar{b})^2\right]\left[\sum_{i=1}^{n}(m_i - \overline{m})^2\right]}}\right)^2 \tag{22}$$

The factors are displayed below:

- n *display* the sample magnitude.
- b_i *display* the projected value.
- $\overline{m}$ and $\bar{b}$ represent the gauged and mean projected values, respectively.
- m_i *display* the gauged value is.
- The critical value from the t-spread is based on the intended grade of assurance and the degrees of freedom displayed by t.
- The mean of the predictor variable in the database is represented by $\bar{x}$.

Metrics of model performance depend heavily on critical hyper parameters like learning rate, hidden layers, and cycles of training. In this research, incorrect tuning resulted in both overfitting or under fitting as evidenced by increased RMSE or decreased R^2. Accurate adjustment enhanced generalization, particularly in MLGR. Optimal hyper parameters were determined according to validation performance.

3 Data collection

Any machine learning project is based on robust data. This work collected data relevant to indoor localization in WSNs with great care. The data comprised several network parameters that were presumed to affect ALE. To ensure generalizability and prevent overfitting of the

model, the data was separated into three distinct parts: training, validation, and testing, in a ratio of 70%, 15%, and 15%, respectively. The training data and the model learned from it took the maximum share of this. The validation data set provided a check, as most hyper parameter tuning to avoid overfitting during training is based on the validation database. Lastly, the unseen testing database verifies the model's generalization to real-life predictive power. This careful data acquisition and segmentation strategy provides a firm ground for an efficient and generalizable ML model in indoor localization. The runtime of the models differed with different configurations. For MLPR with the Gold Rush Optimizer, runtime grew with network depth: 44.3 seconds for 1 layer, 54.7 seconds for 2 layers, and 66.1 seconds for 3 layers. Analogously, the MLPR with the Pelican Optimizer recorded 33.8 seconds for 1 layer, 41.2 seconds for 2 layers, and 52.6 seconds for 3 layers. In comparison with the baseline MLPR without any optimizer, the shortest runtimes were seen at 18.9 seconds for 1 layer, 22.8 seconds for 2 layers, and 30.2 seconds for 3 layers. These results have shown both the optimizers to lengthen computational time, with the Gold Rush Optimizer being the most time-consuming.

3.1 Data description

The database used in this exploration is taken from published literature [45], including six important input traits to predict ALE using the recommended machine learning frameworks. This database is a comprehensive collection of several parameters relevant to indoor localization systems, which would be very useful to explore the relationship between these parameters and localization accuracy.

3.1.1 Input traits

- Anchor ratio: This feature specifies the number of anchor nodes per overall node in the network. Indeed, anchor nodes are vital parts of an indoor localization environment, in which they always act as reference nodes against which the situation of goal nodes is calculated. Enhanced anchor ratios augment location precision by expanding spatial reference points, facilitating superior triangulation. A reduced number of anchors diminishes geometric constraints, resulting in increased estimating uncertainty and more inaccuracy in expected locations.
- Transrange: It defines the range of transmitters in meters in the wireless communication network. The parameter specifies the maximum possible distance over which nodes may communicate effectively and defines the area of coverage of the whole localization system and its accuracy. The transmission range affects the extent and distribution of the signal. A moderate range guarantees dependable connectivity and constant RSSI readings, hence improving localization precision. Excessive range may induce interference, whilst limited range might lead to signal attenuation and inadequate data transmission.
- Node density: Node density characterizes the spatial distribution of the nodes in the network area and quantifies the number of nodes per unit area. It influences the granularities of the measurement in the localization. Increased node density augments the spatial resolution of the network, hence enhancing ALE by providing more comprehensive environmental data. Sparse networks jeopardize adequate coverage, resulting in increased localization mistakes due to a deficiency of proximate references.
- Cycles: This refers to the number of cycles to execute the ML framework training process. Cycles are critical for improving model parameters to ensure predictive performance. Increased iterations provide improved model convergence during optimization and learning, enhancing localization predictions. A reduced number of cycles may result in underfitting, causing the model to inadequately generalize and elevate ALE.
- SDale: The standard deviation of ALE, which reflects the spread or variability of the localization error in different instances or cases. This metric provides insight into the consistency and reliability of the localization system. The standard deviation of ALE indicates the stability and consistency of localization performance. A lower standard deviation indicates strong and dependable predictions, whereas more variability implies the model is susceptible to fluctuations in input properties or environmental conditions.
- Ale: ALE is the target variable that the machine learning frameworks should predict. ALE quantifies the accuracy of localization, which is gauged by the deviation of the estimated situation from the actual situation of goal nodes.

3.2 Data preprocessing

Before framework training, the database endured preprocessing stages to ensure data quality and compatibility with the machine learning frameworks. This included managing missing values, regulating statistical traits, and encoding explicit variables as necessary.

3.3 Database characteristics

The database comprises 107 instances, each characterized by the input traits above and corresponding ALE values. The layered pattern facilitates fine-grained evaluation over various data complexities or network settings. Each layer has a different scenario or level of abstraction, allowing model robustness and generalization to be tested. Such a structure provides insight into where specifically a model performs well or fails, charting future areas of improvement in ALE prediction. Table 2 provides a summary of descriptive statistics, including average, standard deviation, minimum, and maximum, for each feature. Additionally, Fig. 3 presents the bar chart of the input variables.

Table 2: The statistical traits of the input of ALE

Variables	Indicators			
	Min	Max	Avg	St. Dev.
Anchor_ratio	10.000	30.000	20.523	6.708
Trans_range	12.000	25.000	17.879	3.093
Node_density	100.000	300.000	159.813	70.856
Cycles	14.000	100.000	47.888	24.553
Sd_ale	0.003	1.092	0.266	0.183
Ale	0.394	2.268	0.981	0.396

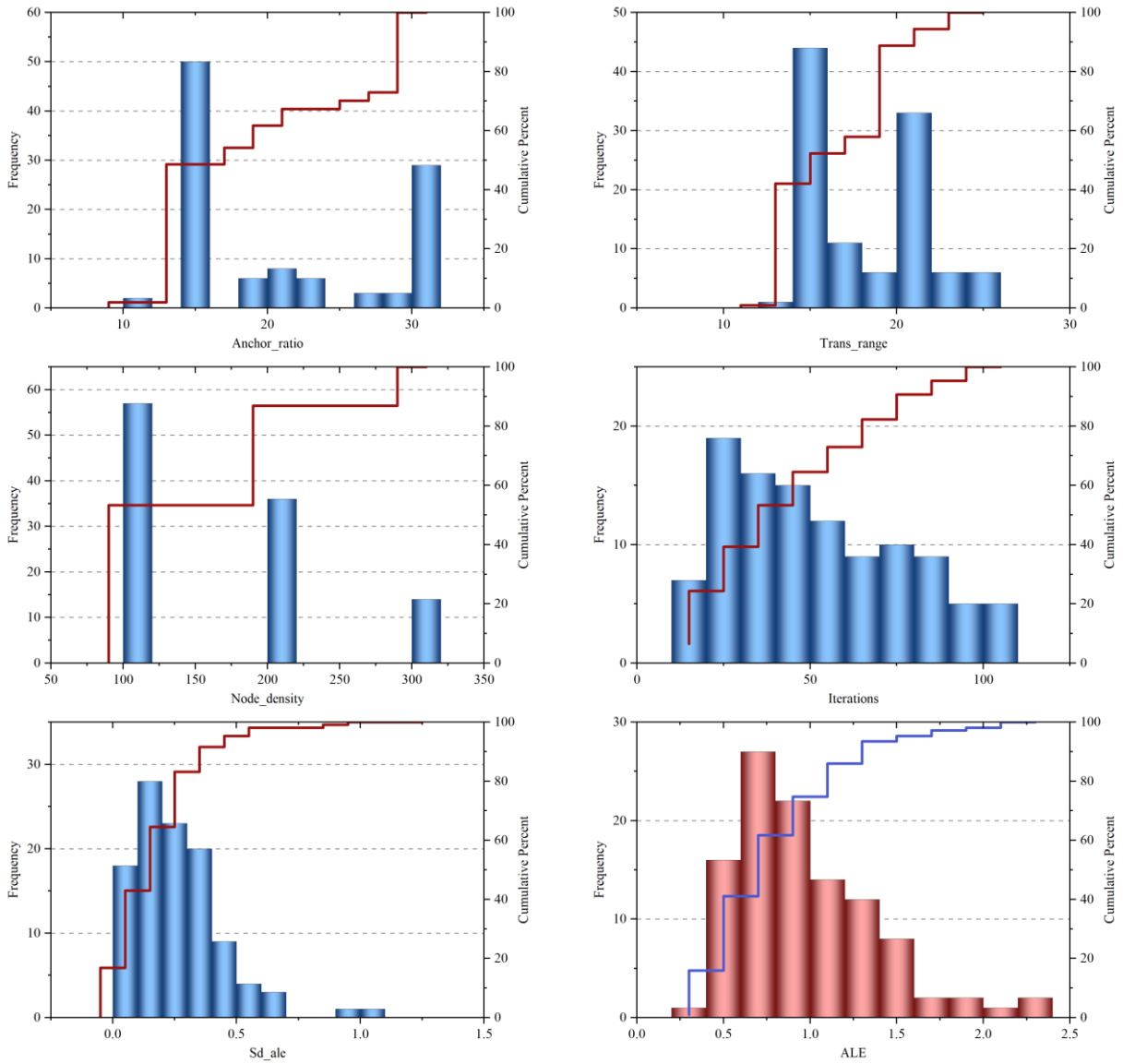


Figure 3: The bar chart for input and output

4 Outcomes and consultation

4.1 Hyper-parameter tuning

Table 3 depicts the optimum number of neurons obtained through random search for each hidden layer at three test stages for MLGR and MLPO models. MLGR and MLPO represent the Multi-Layer Perceptrons augmented with Gold Rush and Pelican Optimizers, respectively. Each model setup (1, 2, 3) represents one different experimental condition, and each row represents one hidden layer. It is clear from the results that neuron distribution is different across different layers and optimizers, highlighting the adaptivity of the tuning process. MLGR tended to utilize a greater number of neurons in deeper layers than MLPO, potentially impacting learning capability and converging behavior. This layer-by-layer tuning underpins the differences in performance seen in later tests.

Table 3: Optimal number of neurons determined via random search for each hidden layer

Hyperparameters	Models					
	MLGR (1)	MLPO (1)	MLGR (2)	MLPO (2)	MLGR (3)	MLPO (3)
Neuron	36	28	37	29	30	25
Neuron	-	-	33	20	20	26
Neuron	-	-	-	-	23	12

4.2 Coincidence curvature

Fig. 4 displays a graph that illustrates the time evolution of a repeating enhancement tactic. It shows how the objective function value of the framework alters at every cycle and whether the process is converging toward the best reaction. Fig. 4 shows the convergence behavior of optimization algorithms in different layers. The curves of the objective function evolution in training cycles are depicted. The steepness and earlier flattening of curves signify more efficient convergence and optimization. The MLGR model shows superior convergence properties by reaching a lower objective in fewer cycles, as is evident in its superior performance in the following prediction assessment. When it comes to optimization problems, an algorithm getting close to convergence exhibits a steady decrease or rise in the objective function until it attains a point where more cycles only produce negligible gains. The chart indicates the excellent productivity of the MLGR model over MLPO across all three layers. After 50 cycles, the MLGR model achieves a commendable accuracy of 0.0345, highlighting its excellence in the first layer. In the second layer, the MLGR model attains a high accuracy of 0.0315 after 30 cycles, and in the third layer, it reaches 0.0294 accuracy after 40 cycles. However, even after 60 cycles, the MLPO model demonstrates noticeable improvement in precision across all layers, though it may not have reached the same level of performance as the MLGR model.

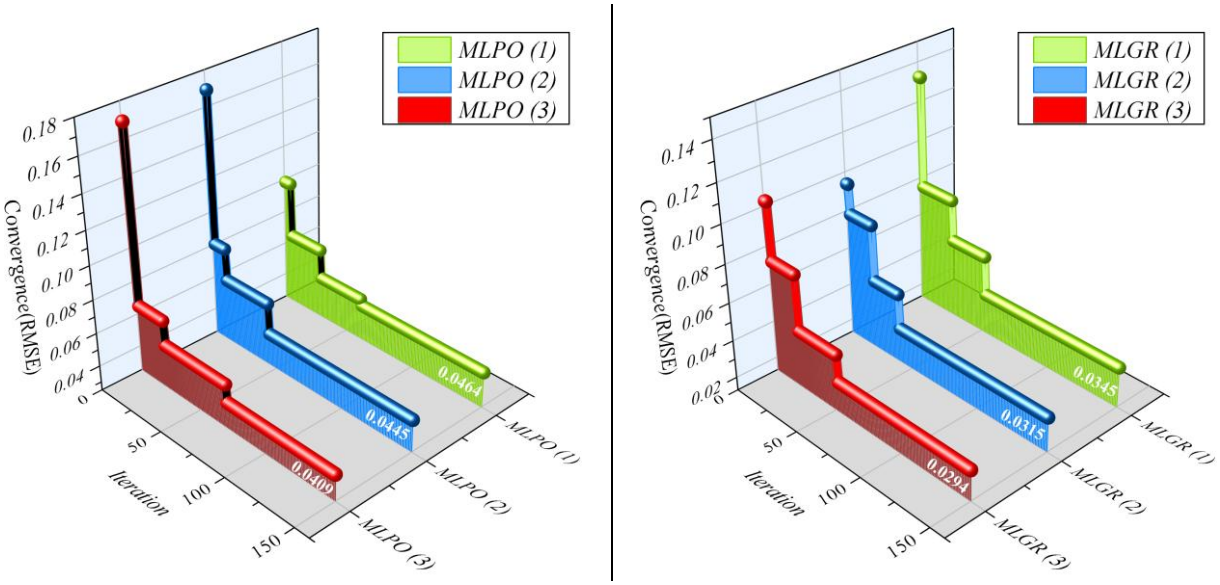


Figure 4: 3D Scatter plot for convergence of compound frameworks

4.3 Frameworks comparison

Table 4 displays the outcomes of the built MLPR frameworks, comparing three different stages and five different metrics. The metrics consist of RMSE, R2, MAE, IOA, and MRAE, and the stages include training, validation, and testing. When a framework's R2 and IOA values are close to one, it performs at its best. On the other hand, values close to zero indicate better model performance for RMSE, MAE, and MRAE.

For instance, in the first layer during training, the MLGR model performs better than the MLPR and MLPO frameworks regarding RMSE, with a value of 0.024. Moreover, with an R2 value of 0.996, the MLGR model beats the MLPO and MLPR frameworks in the same layer and stage. During the training stage, the MLPO model achieved second place with an IOA value of 0.997. Among the three frameworks, the MLPR model has the least promising performance, outperforming the other two.

With a value of 0.040, the MLGR model displays the best performance in MAE during the second layer's validation stage. The MLPO model achieved the second-best performance and is not far behind. The MLPR model performs third with a value of 0.347 in terms of MRAE during the test stage, while the MLPO model comes in second with a value of 0.306. Furthermore, regarding the R2 value during validation, MLGR operates better than MLPO and MLPR, with a value of 0.987.

The test stage's third layer displays that, at 0.991, the MLPR model performs the least robustly in IOA, while MLGR, the best-performing model, achieves a superior value of 0.998. The results for the MAE value in the training stage indicate that MLGR outperforms the others, achieving a value of 0.016. The MLPO model, with a value of 0.030, comes in second place and is followed closely. With values of 0.046 and 0.064, respectively, the MLPO model displays the best performance in the validation stage of RMSE, while MLPR displays the lowest performance. As presented in Table 4, the MLPR model's worst performance occurs in the third layer in the testing phase with an RMSE of 0.080. This suggests a poorer generalization ability in this case than in MLGR and MLPO, given the sensitivity of MLPR to differences in data complexity by layers.

Performance differences between layers are a result of variations in feature distributions as well as data complexity levels. Noisy patterns or less informative patterns in a couple of layers might compromise model generalization. Also, uneven training dynamics lead to weaker tuning of parameters in a particular layer, resulting in increased error rates.

Table 4: The outcome of the created frameworks for the MLPR-based hybrid scheme

Model	Stage	Index values				
		RMSE	R2	MAE	IOA	MRAE
MLPR (1)	Train	0.054	0.981	0.046	0.995	0.361
	Validation	0.056	0.981	0.048	0.995	0.409
	Test	0.080	0.965	0.074	0.990	0.408
MLPO (1)	Train	0.041	0.989	0.035	0.997	0.378
	Validation	0.061	0.983	0.050	0.995	0.447
	Test	0.055	0.984	0.047	0.996	0.270
MLGR (1)	Train	0.024	0.996	0.022	0.999	0.300
	Validation	0.044	0.988	0.039	0.997	0.339
	Test	0.057	0.984	0.051	0.995	0.301
MLPR (2)	Train	0.053	0.982	0.047	0.995	0.562
	Validation	0.081	0.983	0.070	0.991	0.490
	Test	0.066	0.975	0.056	0.994	0.347
MLPO (2)	Train	0.033	0.993	0.028	0.998	0.353
	Validation	0.062	0.983	0.057	0.994	0.492
	Test	0.066	0.975	0.060	0.994	0.306
MLGR (2)	Train	0.023	0.996	0.020	0.999	0.154
	Validation	0.046	0.987	0.040	0.997	0.366
	Test	0.045	0.989	0.042	0.997	0.237
MLPR (3)	Train	0.056	0.981	0.047	0.995	0.630
	Validation	0.064	0.978	0.049	0.994	0.312
	Test	0.080	0.966	0.071	0.991	0.372
MLPO (3)	Train	0.035	0.992	0.030	0.998	0.380
	Validation	0.046	0.988	0.041	0.997	0.363
	Test	0.059	0.981	0.051	0.995	0.298
MLGR (3)	Train	0.018	0.998	0.016	0.999	0.176
	Validation	0.054	0.984	0.050	0.996	0.393
	Test	0.036	0.993	0.029	0.998	0.166

Table 5 presents the statistical significance of comparisons of model performance using the Mann–Whitney U test for three MLP setups. In the stat column, U test statistics are presented, and in the P value column, the corresponding significance levels are provided. The baseline MLPR, MLPR-GRO, and MLPR-POA are compared. With the majority of the p-values lying above the threshold of 0.05 significance, MLPR-POA is just short of significance in MLP 1 and MLP 3 ($p = 0.051$), demonstrating potentially significant improvement over the baseline. This finding complements the performance metrics by providing statistical support for the trends noted and confirming the validity of optimizer-driven improvements in MLPR setups.

Table 5: statistical significance results of the model performances across three multi-layer perceptron

Models	*MLP 1*		*MLP 2*		*MLP 3*	
	stat	*P value*	*stat*	*P value*	*stat*	*P value*
MLGR	2522	0.254	2845	0.890	2522	0.254
MLPO	2261	0.051	2442	0.164	2261	0.051
MLPR	2673	0.607	2457	0.179	2673	0.607

Fig. 5 displays the Plotting of the spread of expanded compound frameworks for the first layer. The expected value is represented on the Y axis, while the gauge value is depicted on the X axis. A tight grouping of data points near the middle diagonal line, corresponding to the ideal R^2 value of 1 represents good predictive accuracy. This grouping indicates little deviation between the actual and produced ALE values. Whenever a model's scatter plot resembles clustering near the line in such a way that the model is likely to be picking up the patterns in the available data consistently. Thus, the data points' nearness to the line indicates the model's performance quality and forecasting accuracy in a straightforward way.

Different parts are color-coded to enhance clarity, as illustrated in Fig. 5. When the population falls below the center line, it signifies underestimation, whereas above the core line indicates overestimation. Robust model execution is evident when the linear line closely aligns with the center line and displays no noticeable angle between them. The diagrams in Fig. 5 show that the MLGR model excels, contrasting with the MLPO and MLPR frameworks. Following the MLGR model, the MLPR model demonstrates the second-highest level of performance. The scatter plot verifies model excellence by illustrating the similarity of the predicted values to actual values on the diagonal line. A model that has points closely grouped on that line indicates higher accuracy and lower residual error levels. MLGR plots demonstrate tight grouping, which indicates higher prediction consistency than MLPO and MLPR.

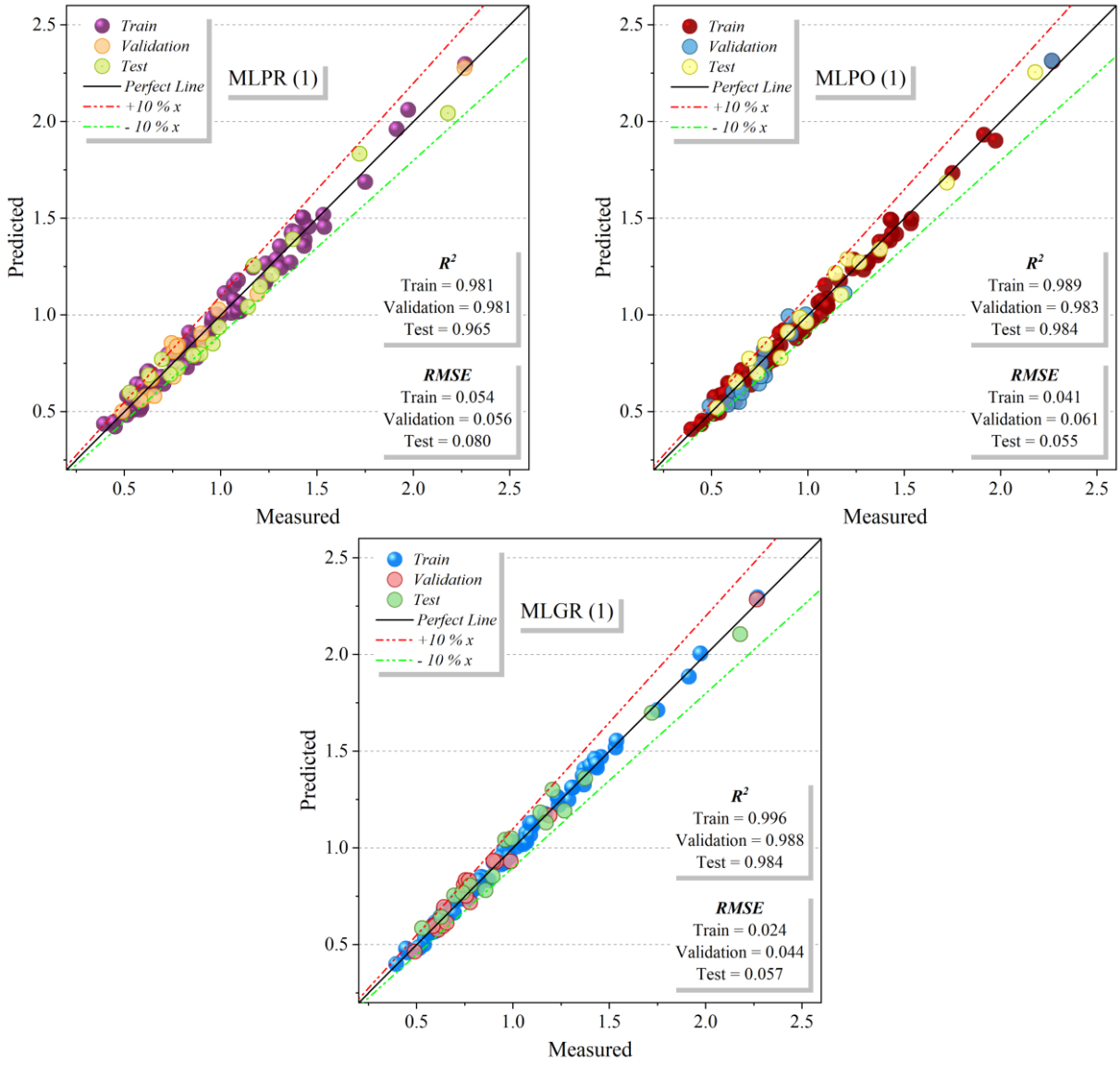


Figure 5: Plotting the spread of expanded compound frameworks

Fig. 6 plots the spread of evolved compound frameworks for the second layer. The expected value is represented on the Y axis, while the gauge value is depicted on the X axis. The population surrounding the middle line, denoting R^2, becomes filled to indicate that the framework with optimal execution is at the core. Different portions are color-coded to enhance clarity, as illustrated in Fig. 6. When the population falls below the center line, it signifies underestimation, whereas above the center line indicates overestimation. A strong model performance is evident when the linear line closely aligns with the center line and displays no noticeable angle

between them. The diagrams in Fig. 6 show that the MLGR model excels, contrasting with the MLPO and MLPR frameworks. Following the MLGR model, the MLPR model demonstrates the second-highest level of performance.

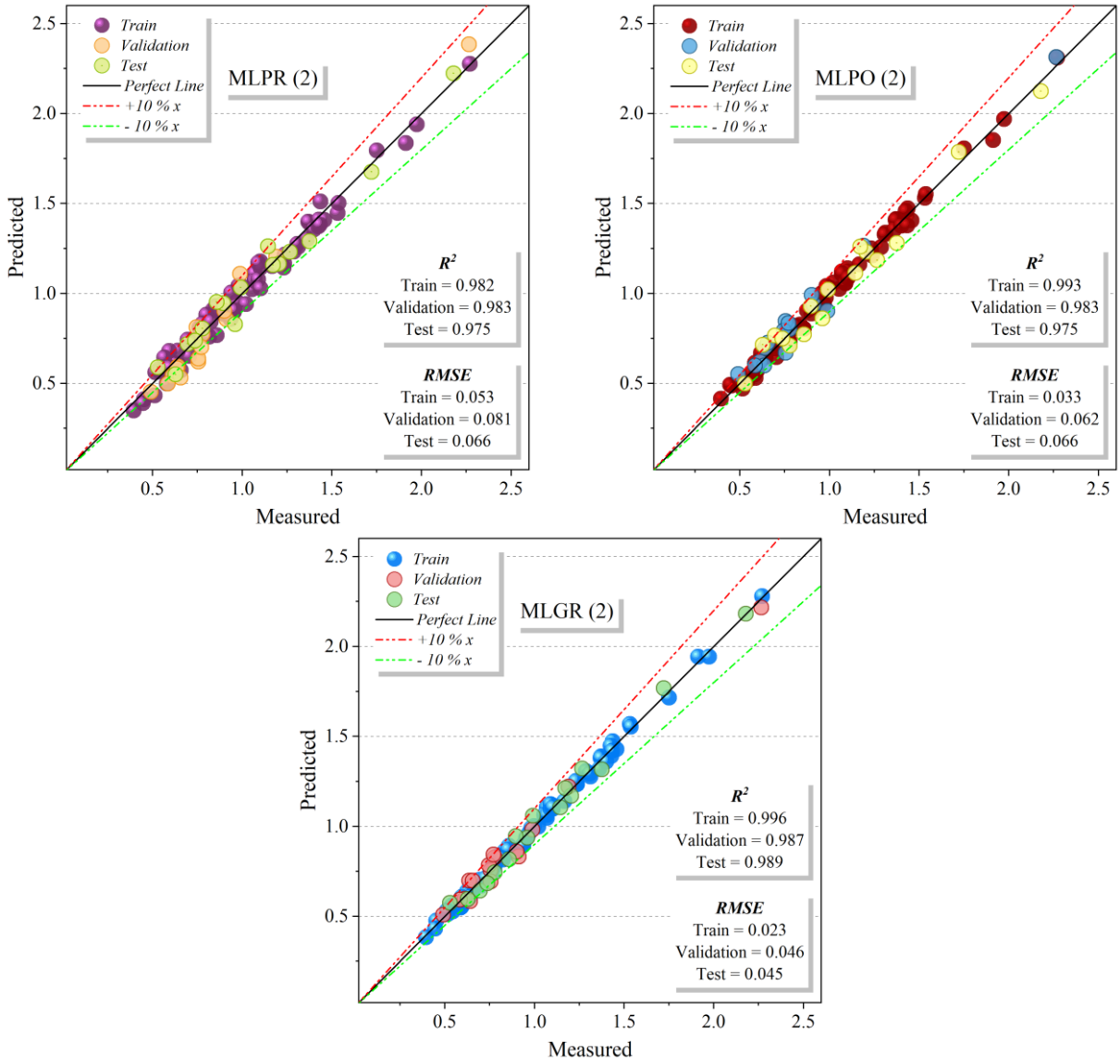


Figure 6: Plotting the spread of expanded compound frameworks

Fig. 7 plots the spread of evolved compound frameworks for the third layer. The expected value is represented on the Y axis, while the gauge value is depicted on the X axis. The population surrounding the middle line, denoting R^2, becomes filled to indicate that the framework with optimal execution is situated at the center. Different parts are color-coded to enhance clarity, as illustrated in Fig. 7. When the population falls below the center line, it signifies underestimation, whereas above the center line indicates overestimation. Robust model execution is evident when the linear line closely aligns with the center line and displays no noticeable angle between them. The diagrams in Fig. 7 show that the MLGR model excels, contrasting with the MLPO and MLPR frameworks. Following the MLGR model, the MLPR model demonstrates the second-highest level of performance.

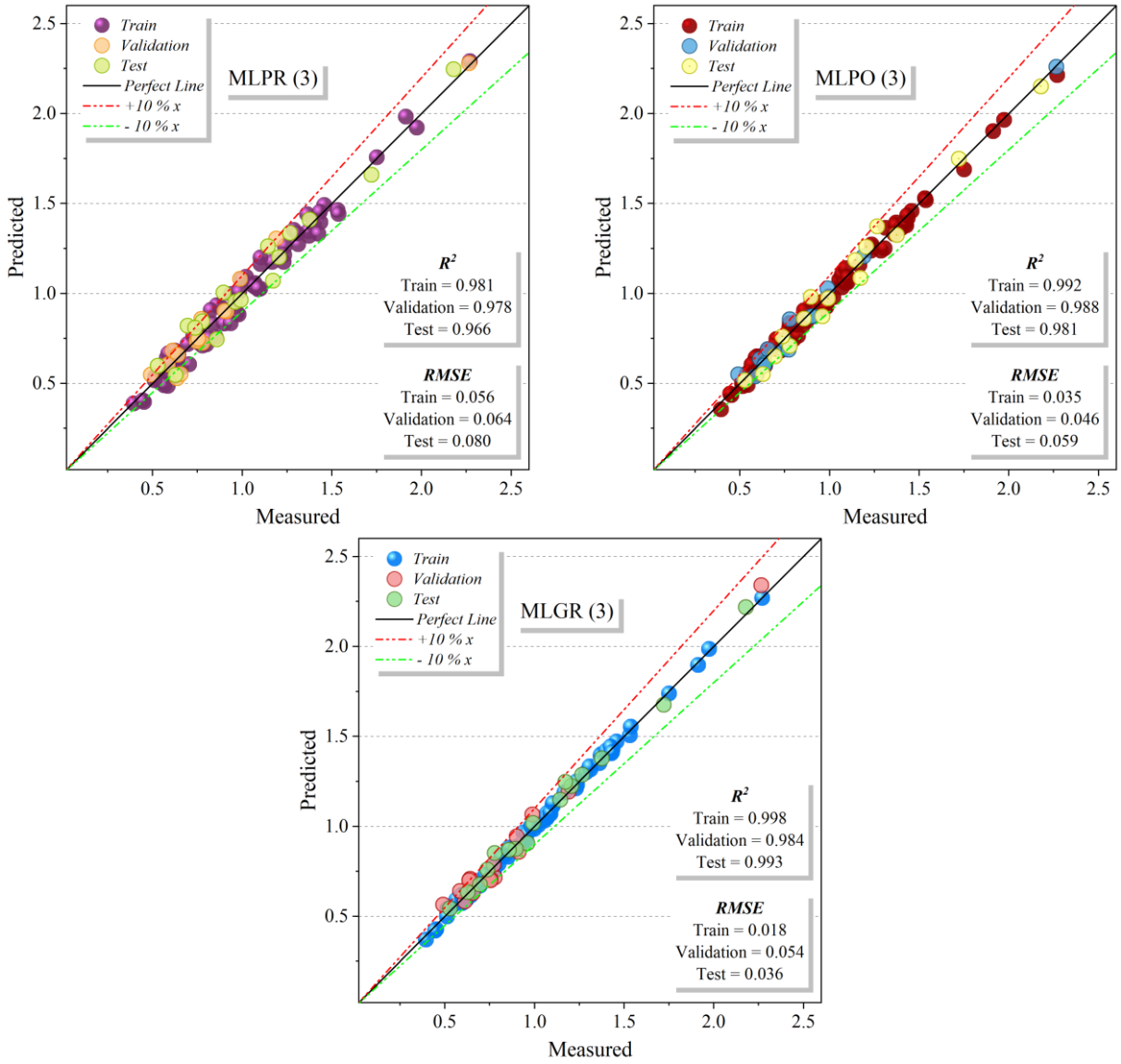
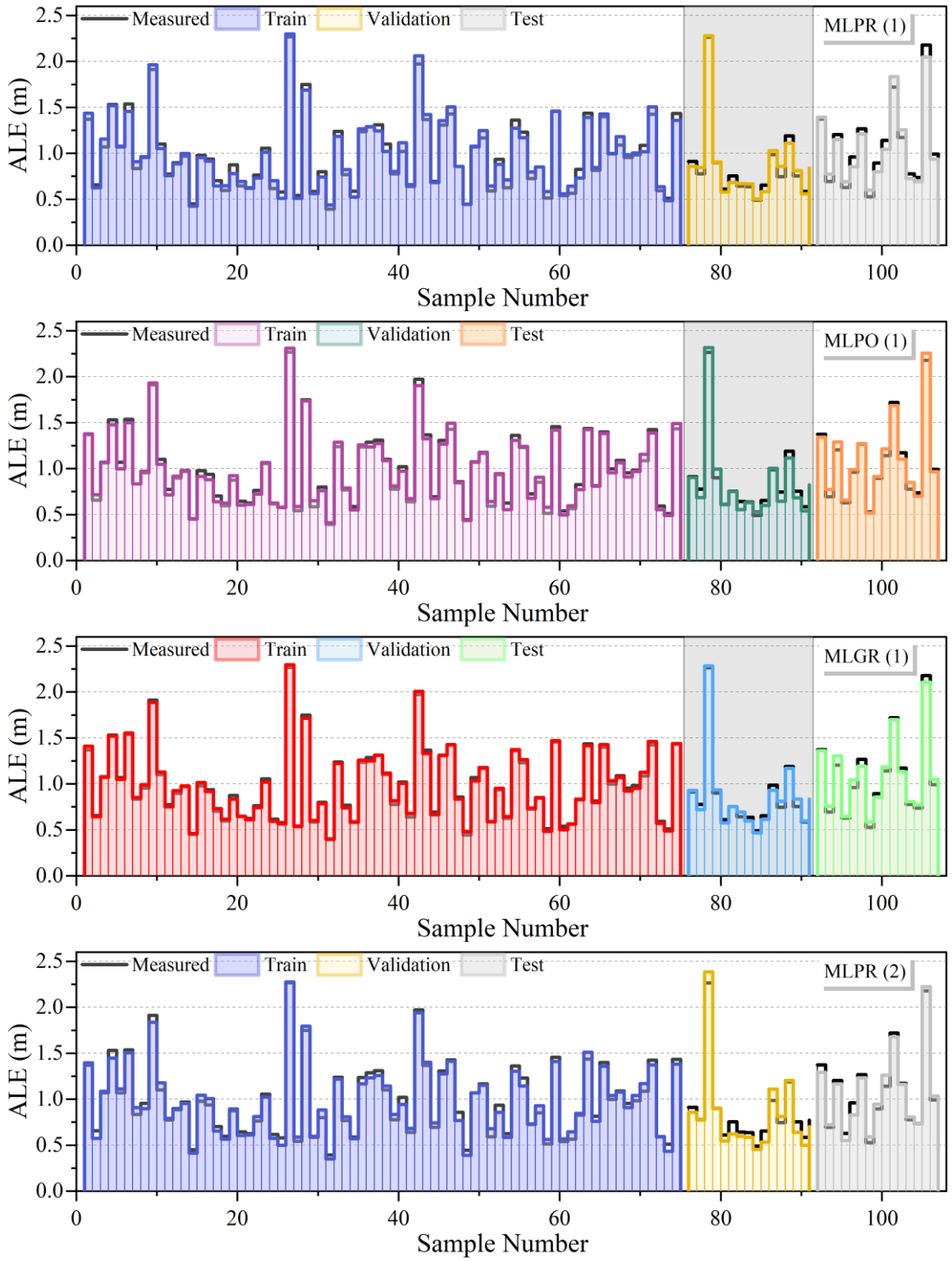


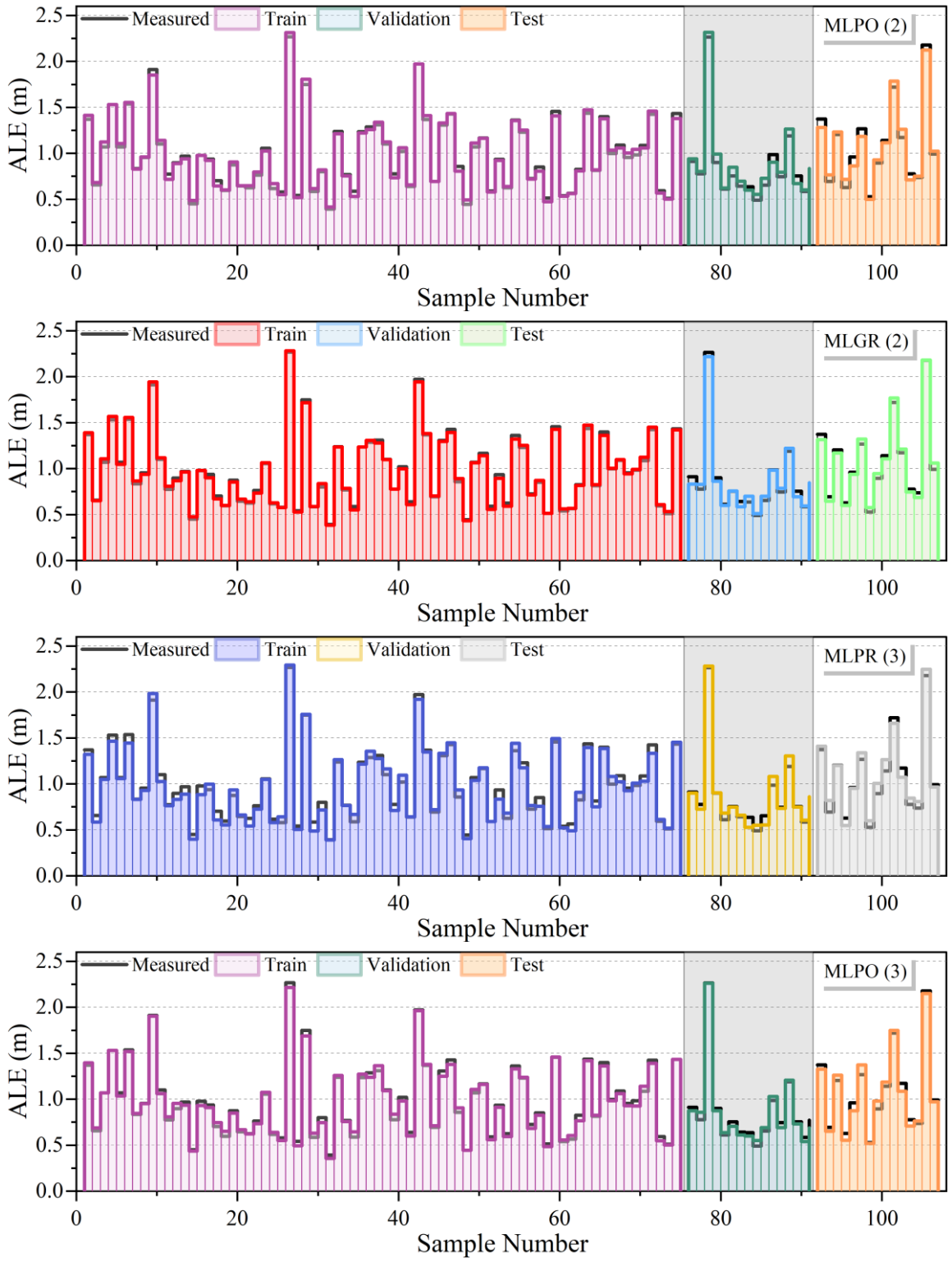
Figure 7: Plotting the spread of evolved compound frameworks

Fig. 8 displays the contrast of projected and gauged values. The alignment between the gauged line and the projected columns indicates the framework's anticipation precision. A close correspondence between the gauged line and the expected column reflects a peak level of precision, while deviations suggest a lower level of performance.

In layer one, the MLGR model demonstrated strong performance in the training stage, with minimal gauged lines exhibiting distances more significant than the projected column. The validation segment of this framework outperforms the test section, showcasing a close alignment between the MLGR framework's anticipations and the observed data. MLPO exhibited poorer performance in the training stage than the MLGR model, with fewer projected columns with distances from gauged lines. However, in both the test and validation parts, MLPO performed excellently.

In the test stage, the MLPR model performs less effectively than the MLPO and MLGR frameworks, with a notable gap between the gauged line and the expected column. Nevertheless, the MLPR model performs satisfactorily in the validation and training stages. This pattern persists across all layers.


Measured
Train
Validation
Test
MLPR (1)
MLPO (1)
MLGR (1)
MLPR (2)
ALE (m)
Sample Number
0.0
0.5
1.0
1.5
2.0
2.5
0
20
40
60
80
100


Measured
Train
Validation
Test
MLPO (2)
ALE (m)
Sample Number
MLGR (2)
MLPR (3)
MLPO (3)

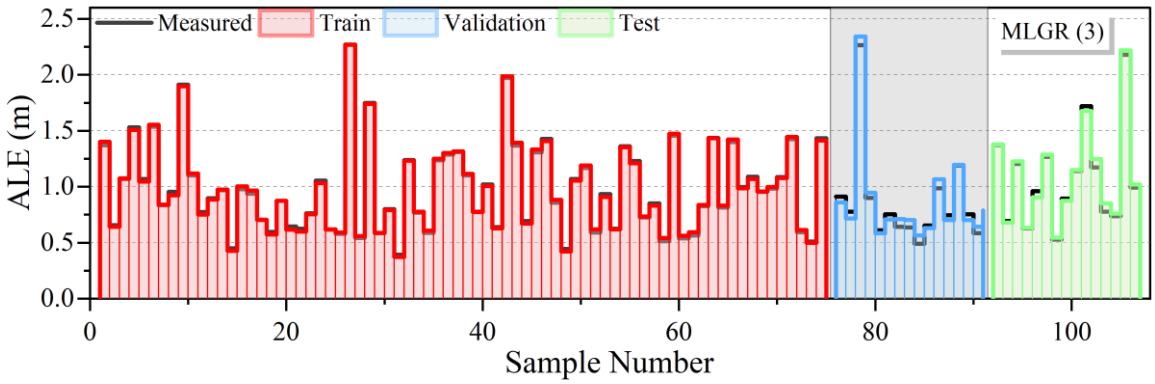
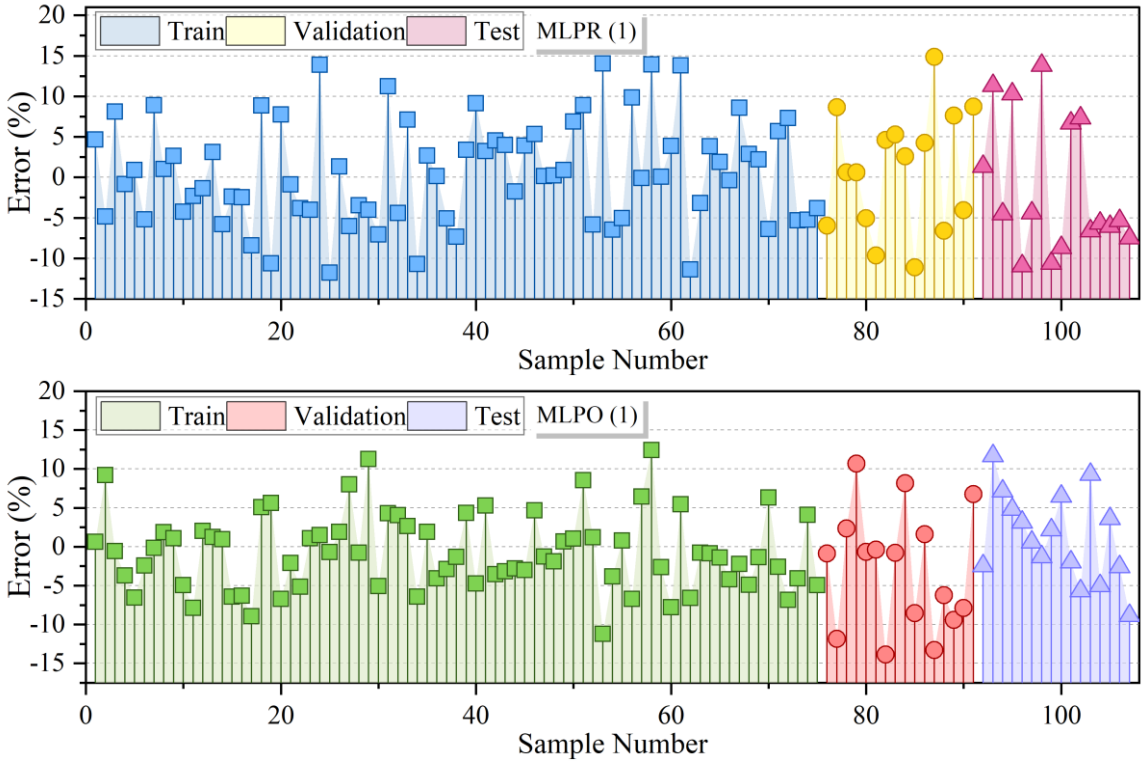


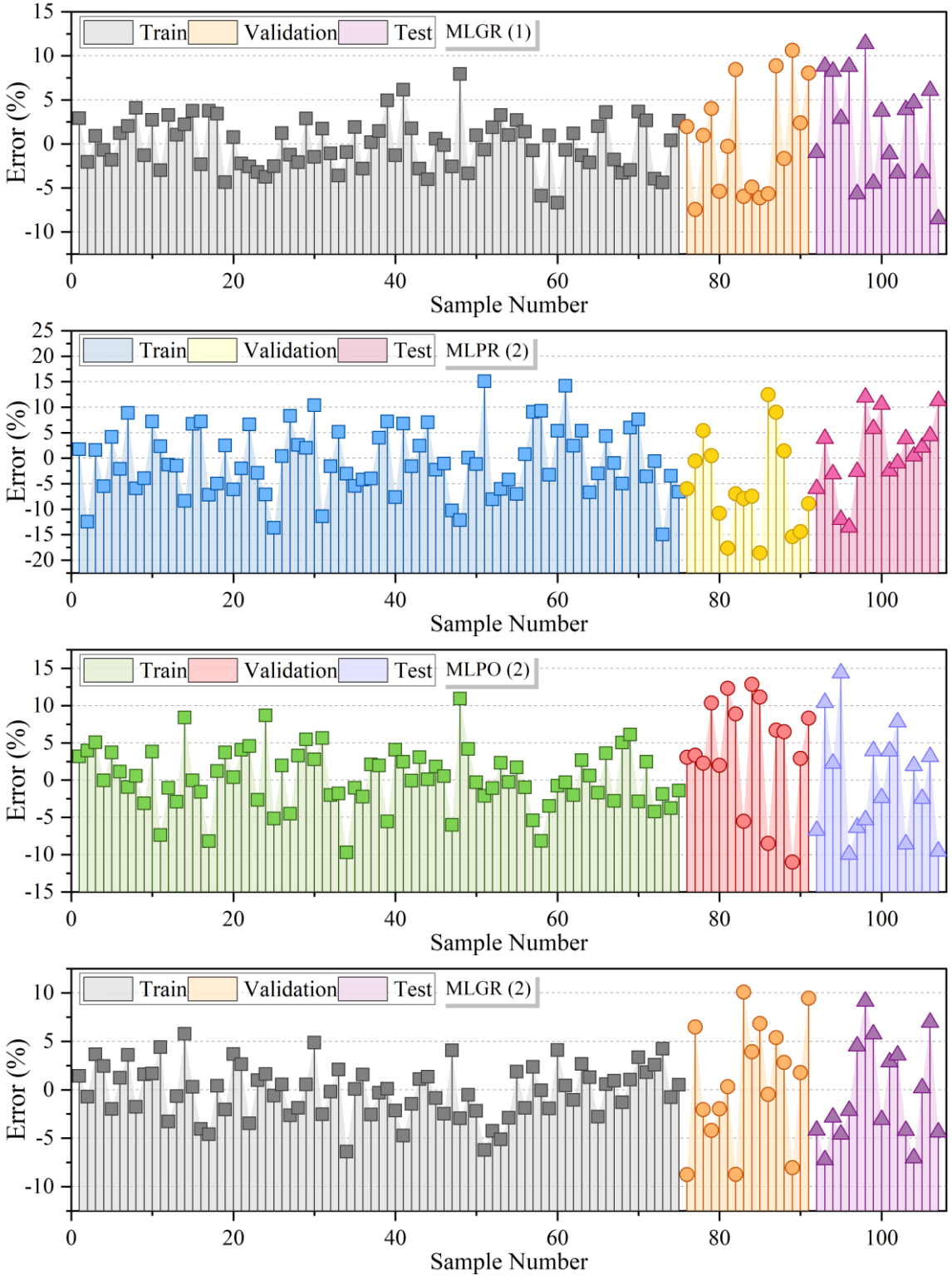
Figure 8: The contrast between projected and gauged values

Fig. 9 illustrates the error percentages of the frameworks based on the Scatter plot. Optimal model performance is indicated when the error proportion approaches zero. For instance, in layer one of MLPR, the peak error proportion during the training stage is approximately 14%. Compared to the other two frameworks, it displays the peak error proportion, reaching about 16%.

The highest error proportion in MLPO is around 14%, lower than the error proportion in MLPR. The training stage of MLPO entails the peak error proportion, and in this exploration, this framework operates the second-best. In contrast, the peak error proportion in MLGR is 12%, the lowest among the three frameworks. The test stage of this framework displays the highest fault proportion, while the validation stage exhibits the lowest error proportion.

Moving to layer two in MLPR, the peak error proportion is approximately 17%, occurring in the training stage. The validation stage has a lower error proportion than the training and test stages. The error proportion varies from -18 to 17 in MLPR. In MLPO, 15% is the highest error proportion, which takes place in the test stage and is the second best during this study. At layer three, the MLGR model has a maximum of about 17% error proportions in the validation stage, while the MLPO model here performs the best in the meantime with an error proportion of 13%. Distributions of error percentage reflect the stability and consistency of predictive performance of the models. Tighter distributions with lower peak error reflect stable performance, whereas wider spreads reflect variable performance and overfitting and under fitting risks. In the research conducted here, MLGR consistently has tighter and lower error distributions, reiterating its stability over layers against MLPO and MLPR.




Train
Validation
Test
MLGR (1)
Error (%)
Sample Number
Train
Validation
Test
MLPR (2)
Error (%)
Sample Number
Train
Validation
Test
MLPO (2)
Error (%)
Sample Number
Train
Validation
Test
MLGR (2)
Error (%)
Sample Number

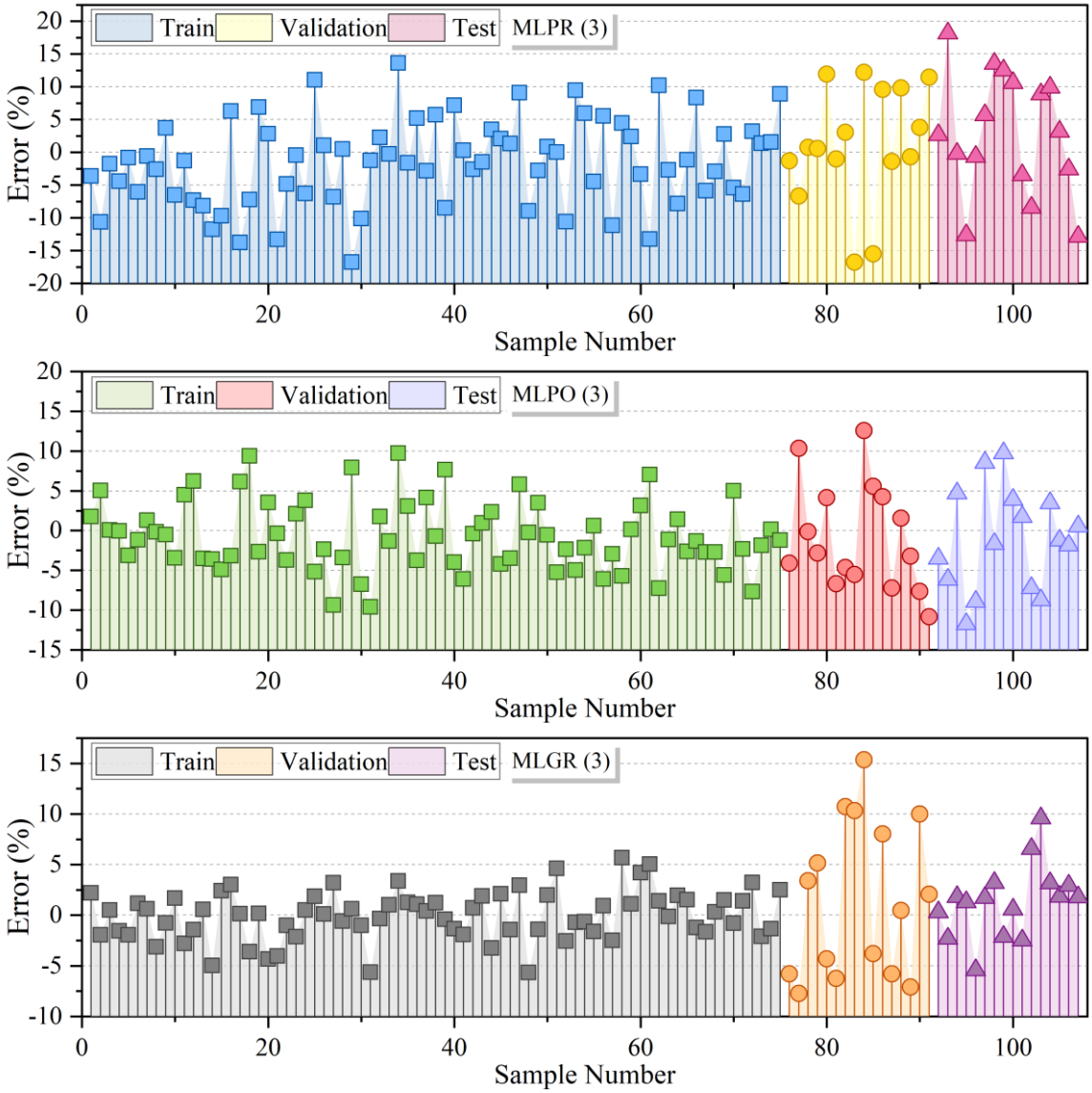


Figure 9: The error proportion of the frameworks in the scatter plot

Fig. 10 displays the half-box plot errors for the recommended frameworks so that their performance attributes can be visually represented. When the median line is closer to zero, it indicates a smaller error proportion in a model. Additionally, the data distribution within the IQR displays the model's effectiveness; the denser the population within this range, the better the efficacy. Looking at the MLGR model in the third layer of the training stage, one finds that the population is indeed closely clustered around the range of IQR. This close grouping around the IQR range represents the model's performance strength. The validation stage confirms this by showing that the median line of this framework hovers nearer to zero.

Conversely, MLPO is widespread in the IQR range, with the median line keeping a distance from zero, especially during the test stage. While the training stage displays a moderately concentrated population around the IQR range, the wider IQR indicates a potential spread in the model's performance. In the case of the MLPR model, the median line approaches zero during the test stage, which means lower error. However, the data distribution around the IQR range is less concentrated, with one data point diverging from this range, implying a nuanced performance in this stage.

This framework reveals a population at the training stage so closely clustered around the IQR range in the second layer of MLGR. In the test stage, the median line has a distance from zero, which indicates that this framework performed poorly at this stage.

Conversely, the MLPO model performs weaker in the validation stage than the MLGR model. The median line's distance from zero is too great, as in the test stage. However, this framework's performance is acceptable in the training stage. The MLPR model performs the weakest in all stages compared to these two frameworks. The population around the IQR range is not clustered, and the median line is too far from the zero point.

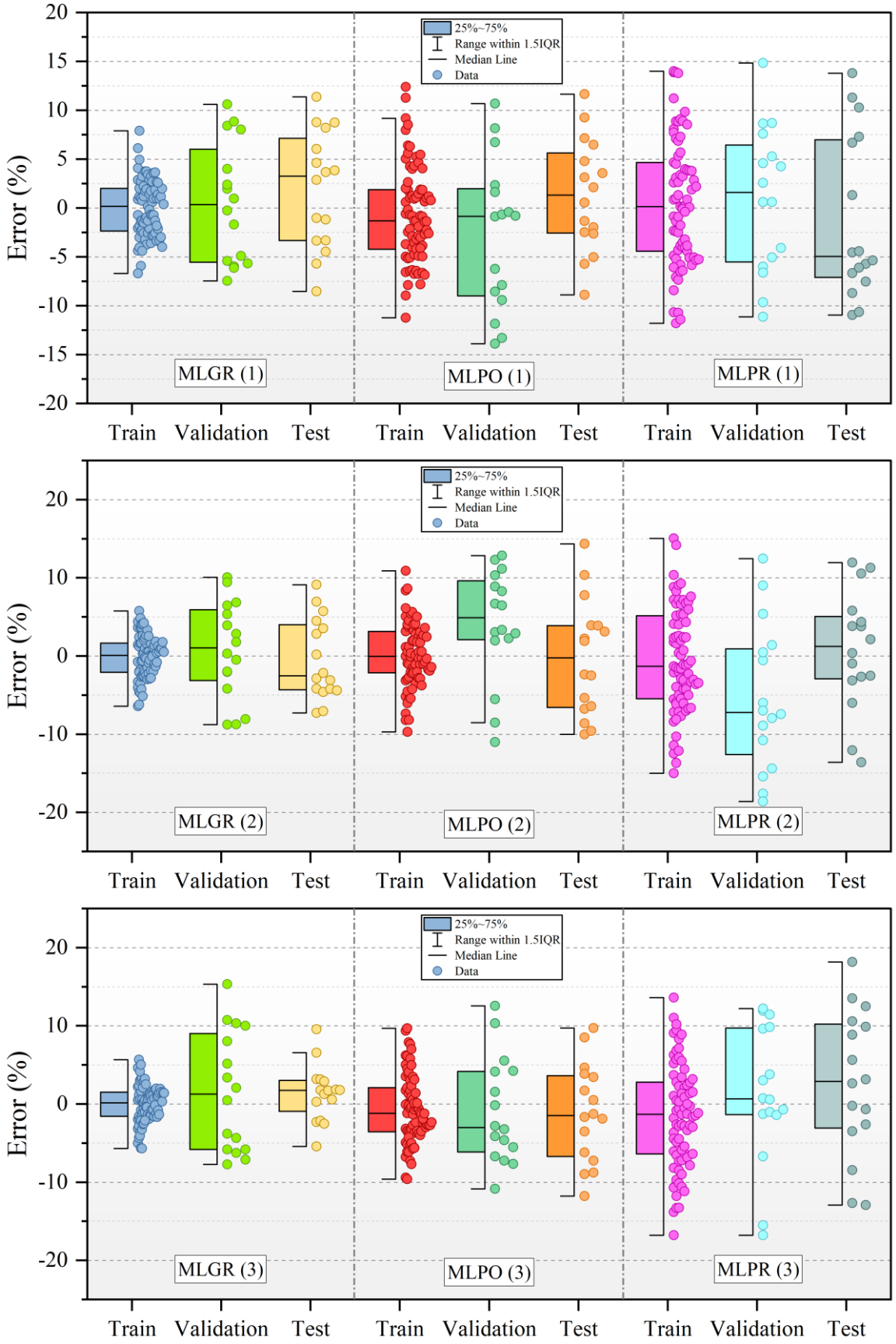


Figure 10: The half box has symbol plot errors for recommended frameworks

The consistently higher performance of the MLGR approach over both MLPO and MLPR can be explained by a number of key benefits inherent in the optimization strategy of GRO and its harmonization with MLPR. First,

GRO provides a human-inspired optimization technique that benefits over the biologically inspired strategy of PO by utilizing group adaptation decision-making and dynamic movement towards regions of optimality. Such an ability allows MLGR to evade premature convergence and avoid being trapped by local optima two typical drawbacks seen in both MLPR and, to a lesser degree, MLPO. Second, selective pausing by GRO through fitness-distance correlation enables a better exploitation/exploration trade-off. It leads to more targeted convergence behavior and more rapid identification of optimal solutions in ALE task prediction, while PO heavily depends on random exploration, potentially slowing down convergence. Third, comparative results in all three layers and evaluation phases validate MLGR's dominance. When tested using the third layer, MLGR recorded an RMSE of 0.036, R^2 of 0.993, and MRAE of 0.166, surpassing MLPO (RMSE: 0.059, R^2: 0.981, MRAE: 0.298) and MLPR (RMSE: 0.080, R^2: 0.966, MRAE: 0.372) performance. Ultimately, MLGR benefits from superior generalization to novel data through GRO's progressive correction mechanism. Combined, these aspects increased convergence, enhanced optimization depth, and higher predictive accuracy render MLGR the best framework to make ALE predictions in this work. The hybrid strategy does have increased training time as a result of additional computational cost from GRO and PO, but this is counterbalanced by enhanced model accuracy and convergence. Inference, on the other hand, is still efficient because optimization happens only at training time, hence making it appropriate for offline training and real-time inference applications.

5 Conclusion

This exploration examined the potential of ML to boost indoor localization in WSNs. The limitations of GPS in indoor environments were addressed, emphasizing the need for alternative methods such as reference nodes and localization frameworks. Existing range-based and range-free frameworks were reviewed alongside recent advancements in machine-learning approaches for WSN localization. In this study, the recommended approach utilizes MLPR to predict ALE. To potentially enhance MLPR's performance, the study investigates the use of optimizers containing GRO and POA. The hybridized frameworks' performance was compared to identify the most accurate and precise solution for ALE prediction. The developed frameworks were evaluated in three layers and stages: Training, Validation, and Testing. The MLPO model achieved the best performance in the first laye. with a low RMSE of 0.055, followed closely by the MLGR model at 0.057. However, the MLPR model showed the weakest performance here, reaching an RMSE of 0.080. In the second layer, the focus shifted to R-squared (R^2), where the MLGR model shone with a value of 0.989. Interestingly, the MLPO and MLPR frameworks achieved similar results in this layer, sharing the second-best R^2 of 0.975. Finally, the third layer assessed with MRAE cemented the MLGR framework's dominance. It achieved the lowest MRAE of 0.166, indicating the most accurate predictions. The MLPO model followed with a value of 0.289, while MLPR revealed the peak MRAE (0.372), suggesting the least accurate predictions in the final layer. This analysis highlighted the consistent superiority of the MLGR model across most metrics, showcasing its effectiveness in predicting ALE. While the MLPO model showed promise in the first layer, its performance lagged in the later stages. The MLPR model consistently had the weakest performance, underlining the potential benefits observed in this study when using optimizers contained GRO with the MLPR model. Even in strong performance, the suggested approach needs greater computational power through hybrid optimization, making its applicability to real-time or resource-limited settings narrow. The model's reliance upon well-processed input data might also lessen robustness during uncertain or incomplete data, as opposed to probabilistic or ensemble approach methods that are more flexible when dealing with uncertainty. This approach has demonstrably enhanced the framework's predictive power for ALE estimation. Therefore, this research advances ML strategies for achieving efficient and reliable indoor localization within WSNs.

Competing Interests

The scholars claim no competing interests.

Authorship Contribution Statement

Jian SUN: Writing-Original draft preparation Conceptualization, Supervision, Project administration.

Jia LIU: Methodology, Software

Data Availability

The scholars will make the raw data supporting this article's conclusions available without undue reservation.

Declarations

Not applicable.

Conflicts of Interest

The scholars claimed no conflicts of interest considering this investigation.

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 displays honest work.

Funding

Not applicable.

Ethical Approval

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

References

[1] Khan, I., F. Belqasmi, R. Glitho, N. Crespi, M. Morrow and P. Polakos (2015). Wireless sensor network virtualization: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, 18(1), pp. 553–576. https://doi.org/10.1109/COMST.2015.2412971.

[2] Jouhari, M., K. Ibrahimi, H. Tembine and J. Ben-Othman (2019). Underwater wireless sensor networks: A survey on enabling technologies, localization protocols, and internet of underwater things. *IEEE Access*, IEEE, 7, pp. 96879–96899. https://doi.org/10.1109/ACCESS.2019.2928876.

[3] Xiong, H. and M.L. Sichitiu (2019). A lightweight localization solution for small, low resources WSNs. *Journal of Sensor and Actuator Networks*, MDPI, 8(2), pp. 26. https://doi.org/10.3390/jsan8020026.

[4] Zheng, J. and A. Dehghani (2012). Range-free localization in wireless sensor networks with neural network ensembles. *Journal of Sensor and Actuator Networks*, MDPI, 1(3), pp. 254–271. https://doi.org/10.3390/jsan1030254.

[5] Wang, Y., Z. Chen, T. Zhu, J. Liu and X. Du (2025). Intelligent Detection and Localization of Cable Faults Using Advanced Discharge Analysis Techniques. *Informatica*, Slovenian Society Informatika, 49(9). https://doi.org/10.31449/inf.v49i9.5468.

[6] Yan, C (2024). Application of a Graphical Image Pre-retrieval Method Based on Compatible Rough Sets to the Self-localization Method of Mobile Robots. *Informatica*, Slovenian Society Informatika, 48(11). https://doi.org/10.31449/inf.v48i11.5508.

[7] Singh, A., S. Sharma, J. Singh and R. Kumar (2019). Mathematical modelling for reducing the sensing of redundant information in WSNs based on biologically inspired techniques. *Journal of Intelligent & Fuzzy Systems*, Sage Publications, 37(5), pp. 6829–6839. https://doi.org/10.3233/JIFS-190605.

[8] Amutha, J., S. Sharma and J. Nagar (2020). WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues. *Wireless Personal Communications*, Springer Nature, 111, pp. 1089–1115. https://doi.org/10.1007/s11277-019-06903-z.

[9] Khelifi, M., S. Moussaoui, S. Silmi and I. Benyahia (2015). Localisation algorithms for wireless sensor networks: A review. *International Journal of Sensor Networks*, Inderscience, 19(2), pp. 114–129. https://doi.org/10.1504/IJSNET.2015.071632.

[10] Tarrío, P., A.M. Bernardos and J.R. Casar (2011). Weighted least squares techniques for improved received signal strength based localization. *Sensors*, MDPI, 11(9), pp. 8569–8592. https://doi.org/10.3390/s110908569.

[11] Whitehouse, K (2002). *The design of calamari: an ad-hoc localization system for sensor networks*.

[12] Wen, C.-Y. and F.-K. Chan (2010). Adaptive AOA-aided TOA self-positioning for mobile wireless sensor networks. *Sensors*, MDPI, 10(11), pp. 9742–9770. https://doi.org/10.3390/s101109742.

[13] Bulusu, N., J. Heidemann and D. Estrin (2000). GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, IEEE, 7(5), pp. 28–34. https://doi.org/10.1109/98.878533.

[14] Niculescu, D. and B. Nath (2001). Ad hoc positioning system (APS), In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*, IEEE, San Antonio, TX, USA, pp. 2926–2931. https://doi.org/10.1109/GLOCOM.2001.965964.

[15] Waadt, A.E., C. Kocks, S. Wang, G.H. Bruck and P. Jung (2010). Maximum likelihood localization estimation based on received signal strength, In *2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010)*, IEEE, Rome, Italy, pp. 1–5. https://doi.org/10.1109/ISABEL.2010.5702817.

[16] Coluccia, A. and F. Ricciato (2014). RSS-based localization via Bayesian ranging and iterative least squares positioning. *IEEE Communications Letters*, IEEE, 18(5), pp. 873–876. https://doi.org/10.1109/LCOMM.2014.040214.132781.

[17] Coluccia, A. and A. Fascista (2019). Hybrid TOA/RSS range-based localization with self-calibration in asynchronous wireless networks. *Journal of Sensor and Actuator Networks*, MDPI, 8(2), p. 31. https://doi.org/10.3390/jsan8020031.

[18] Kulkarni, V.R., V. Desai and R. V Kulkarni (2019). A comparative investigation of deterministic and metaheuristic algorithms for node localization in wireless sensor networks. *Wireless Networks*, Springer Nature, 25, pp. 2789–2803. https://doi.org/10.1007/s11276-019-01994-9.

[19] Kennedy, J. and R. Eberhart (1995). Particle swarm optimization, In *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, Perth, WA, Australia, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

[20] Gopakumar, A. and L. Jacob (2008). Localization in wireless sensor networks using particle swarm optimization, In *2008 IET International Conference on Wireless, Mobile and Multimedia Networks*, IET, Beijing, China, pp. 227–230.

[21] Goyal, S. and M.S. Patterh (2014). Wireless sensor network localization based on cuckoo search algorithm. *Wireless Personal Communications*, Springer Nature, 79, pp. 223–234. https://doi.org/10.1007/s11277-014-1850-8.

[22] Cheng, J. and L. Xia (2016). An effective cuckoo search algorithm for node localization in wireless sensor network. *Sensors*, MDPI, 16(9), pp. 1390. https://doi.org/10.3390/s16091390.

[23] Liu, T. and Z. Zhang (2024). The Application Effect of Improved CS-RBF Neural Network in Industrial Internet of Things Node Localization. *Informatica*, Slovenian Society Informatika, 48(13). https://doi.org/10.31449/inf.v48i13.6004.

[24] Tan, C. and P. Li (2025). The Application of Logistics Robot in the Solution of Locating Route Problems in Trans CAD. *Informatica*, Slovenian Society Informatika, 49(11). https://doi.org/10.31449/inf.v49i11.6602.

[25] Morelande, M.R., B. Moran and M. Brazil (2008). Bayesian node localisation in wireless sensor networks, In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Las Vegas, NV, USA, pp. 2545–2548. https://doi.org/10.1109/ICASSP.2008.4518167.

[26] Musso, C., N. Oudjane and F. Le Gland (2001). Improving regularised particle filters, In *Sequential Monte Carlo Methods in Practice*, Springer, New York, NY, USA, pp. 247–271. https://doi.org/10.1007/978-1-4757-3437-9_12.

[27] Gharghan, S.K., R. Nordin and M. Ismail (2016). A wireless sensor network with soft computing localization techniques for track cycling applications. *Sensors*, MDPI, 16(8), pp. 1043. https://doi.org/10.3390/s16081043.

[28] Ahmadi, H. and R. Bouallegue (2017). Exploiting machine learning strategies and RSSI for localization in wireless sensor networks: A survey, In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, Valencia, Spain, pp. 1150–1154. https://doi.org/10.1109/IWCMC.2017.7986447.

[29] Makhlouf, A., A. Benmachiche and I. Boutabia (2024). Enhanced Autonomous Mobile Robot Navigation Using a Hybrid BFO/PSO Algorithm for Dynamic Obstacle Avoidance. *Informatica*, Slovenian Society Informatika, 48(17). https://doi.org/10.31449/inf.v48i17.6716.

[30] Zhong, C. and G. Yang (2025). Design and Application of Improved Genetic Algorithm for Optimizing the Location of Computer Network Nodes. *Informatica*, Slovenian Society Informatika, 49(16). https://doi.org/10.31449/inf.v49i16.7201.

[31] Li, J. and Z. Tian (2024). Construction and Optimization of a Precise Positioning Model for Logistics Vehicles Based on Sustainable Operation. *Informatica*, Slovenian Society Informatika, 48(17). https://doi.org/10.31449/inf.v48i17.6393.

[32] Bhatti, M.A., R. Riaz, S.S. Rizvi, S. Shokat, F. Riaz and S.J. Kwon (2020). Outlier detection in indoor localization and Internet of Things (IoT) using machine learning. *Journal of Communications and Networks*, IEEE, 22(3), pp. 236–243. https://doi.org/10.1109/JCN.2020.000018.

[33] Wang, L., M.J. Er and S. Zhang (2020). A kernel extreme learning machines algorithm for node localization in wireless sensor networks. *IEEE Communications Letters*, IEEE, 24(7), pp. 1433–1436. https://doi.org/10.1109/LCOMM.2020.2986676.

[34] Gharghan, S.K., R. Nordin and M. Ismail (2016). A wireless sensor network with soft computing localization techniques for track cycling applications. *Sensors*, MDPI, 16(8), pp. 1043. https://doi.org/10.3390/s16081043.

[35] Goyal, S. and M.S. Patterh (2014). Wireless sensor network localization based on cuckoo search algorithm. *Wireless Personal Communications*, Springer Nature, 79, pp. 223–234. https://doi.org/10.1007/s11277-014-1850-8.

[36] Morelande, M.R., B. Moran and M. Brazil (2008). Bayesian node localisation in wireless sensor networks, In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Las Vegas, NV, USA, pp. 2545–2548. https://doi.org/10.1109/ICASSP.2008.4518167.

[37] Noriega, L (2005). Multilayer perceptron tutorial. School of Computing. *Staffordshire University*, 4, p. 5.

[38] Ramchoun, H., Y. Ghanou, M. Ettaouil and M.A. Janati Idrissi (2016). Multilayer perceptron: Architecture optimization and training. International *Journal of Interactive Multimedia and Artificial Intelligence*, Universidad Internacional de La Rioja (UNIR), http://doi.org/10.9781/ijimai.2016.415.

[39] Murtagh, F (1991). Multilayer perceptrons for classification and regression. *Neurocomputing*, Elsevier, 2(5–6), pp. 183–197. https://doi.org/10.1016/0925-2312(91)90023-5.

[40] Bishop, C.M (1995). *Neural networks for pattern recognition*. Oxford university press.

[41] Trojovský, P. and M. Dehghani (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors*, MDPI, 22(3), pp. 855. https://doi.org/10.3390/s22030855.

[42] Alamir, N., S. Kamel, T.F. Megahed, M. Hori and S.M. Abdelkader (2023). Developing hybrid demand response technique for energy management in microgrid based on pelican optimization algorithm. *Electric Power Systems Research*, Elsevier, 214, pp. 108905. https://doi.org/10.1016/j.epsr.2022.108905.

[43] Tuerxun, W., C. Xu, M. Haderbieke, L. Guo and Z. Cheng (2022). A wind turbine fault classification model using broad learning system optimized by improved pelican optimization algorithm. *Machines*, MDPI, 10(5), pp. 407. https://doi.org/10.3390/machines10050407.

[44] Al-Wesabi, F.N., H.A. Mengash, R. Marzouk, N. Alruwais, R. Allafi, R. Alabdan, M. Alharbi and D. Gupta (2023). Pelican Optimization Algorithm with federated learning driven attack detection model in Internet of Things environment. *Future Generation Computer Systems*, Elsevier. https://doi.org/10.1016/j.future.2023.05.029.

[45] Singh, A., V. Kotiyal, S. Sharma, J. Nagar and C.-C. Lee (2020). A machine learning approach to predict the average localization error with applications to wireless sensor networks. *IEEE Access*, IEEE, 8, pp. 208253–208263. https://doi.org/10.1109/ACCESS.2020.3038645.

Dual Interactive Wasserstein GAN Fostered Offset Based In-Loop Filtering for HEVC

Vanishree Moji[1*], Bharathi Gururaj[1], Mathivanan Murugavelu[2]
[1]Department of ECE, K S Institute of Technology, Bengaluru, India
[2]Department of ECE, Salem College of Engineering and Technology, Salem, India
E-mail: vanishreemoji@gmail.com
[*]Corresponding author

Keywords: dual interactive wasserstein generative adversarial network, offset based in-loop filter, sample adaptive offset filter, video compression

Received: February 11, 2025

Deep learning technology is so flexible and effective that many attempts have been made to replace the features that are included in video codecs, such as High Efficiency Video Coding, with deep learning-based alternatives. This article suggests a Dual interactive Wasserstein Generative Adversarial Network fostered Offset-Based In-Loop Filtering in HEVC (DiWaGAN-OBILF-HEVC). Previously, deep learning based in-loop filtering techniques only fixed distorted frames. The DiWaGAN-OBILF-HEVC system transfers offsets as side information, which makes prediction filtering much more accurate while using less bit rate. The DiWaGAN based sample adaptive offset filter is broadly divided into 2 phases: 1) category of error as per neighbouring image intensity values and 2) calculation of optimum offsets as per category of error. The DiWaGAN draws inspiration from the Sample Adaptive Offset (SAO) filter-Edge Offset (EO) type in high efficiency video coding, where one network classifies the error to the reconstructed signal edge shape while the other network simultaneously forecasts ideal offset values. These offsets are fed to the decoder to enhance its accuracy. The DiWaGAN-OBILF-HEVC method is implemented in Pytorch. The efficacy of DiWaGAN-OBILF-HEVC is analysed with performance metrics like Visual quality, PSNR, Bjontegaard rate difference (BD rate), and average execution time. Further, the proposed DiWaGAN-OBILF-HEVC method provides a higher PSNR of 16.25%, a lower Bjontegaard rate difference of 27.47% and a lower execution time of 15.993% when compared with existing methods like Efficient In-Loop Filtering Based on Enhanced Deep Convolutional Neural Networks for HEVC, Offset-Based in-Loop Filtering with Deep Network in HEVC and deep CNN for VVC in-loop filtration respectively.

Povzetek: Članek uvaja DiWaGAN-OBILF-HEVC, GAN-podprto ofsetno filtriranje v HEVC, ki izboljša PSNR, zmanjša BD-rate ter čas izvajanja, presega CNN-osnovane filtre in prinaša bolj kvalitetno video kodiranje.

1 Introduction

As electronic devices like digital televisions and smartphones continue to advance, video has become an integral part of human life [1–2]. The rapid advancement of network technology is also contributing to the rapid growth of video traffic on the network [3–4]. In addition to the increase in the quantity of videos, the size of the display has led to an increase in video resolution. There is a greater need for video compression that manages higher quality while utilising lesser bits [5–6]. HEVC is the newest standard video encoder. HEVC encoding and decoding procedures resemble those of earlier industry-standard codecs like H.264/AVC. HEVC employs sophisticated parallel processing techniques, making it more efficient and faster than its predecessors. More specifically, HEVC uses 40 to 50% lesser bits than H.264/AVC for producing video of comparable quality with the same video frame. Here, numerous components, like motion compensation, quantization, and in-loop filtering are included [7–8]. This research concentrates on in-loop filtering [9]. The in-loop filter's objective is to minimise block-wise quantization's compression artifacts while restoring the damage frame as close to the original as is practical [10–11].

The in-loop filter comprises a Sample Adaptive Offset (SAO) Filter [12] and a De-blocking Filter (DF) [13]. For the coding unit (CU), DF lessens blocking effects caused by block-wise processing [14]. Besides, SAO lessens high-frequency components during quantization, which results in a reduction of ringing effects [15]. Recent research has shown that deep learning methods can be highly effective for image restoration, including compression artifact removal, owing to their learning capability and feature representation power [16–20]. As an early attempt at image restoration through deep

learning, the super-resolution convolutional neural network (SRCNN) used a simple structure, but the outcome is remarkable [21–25]. The performance is greatly enhanced by the deep residual network design of the very deep super-resolution convolutional network (VDSR). Many researchers have attempted to reduce compression artifacts using deep learning techniques for successful SRCNN with VDSR [26–30].

This study focuses on offset-based in-loop filtering, aiming to improve the SAO component of HEVC using a novel deep learning approach. A new model called Dual Interactive Wasserstein GAN-based Offset In-Loop Filtering for HEVC (DiWaGAN-OBILF-HEVC) is proposed, which transfers offset information as side data to significantly enhance filtering prediction accuracy while minimizing bit-rate overhead. Unlike prior In-Loop filter depending on Enhanced Generative Adversarial Network De-blocking Filter (EGANDF) [31] and Sample Adaptive Offset Filtering models that recover the deformed frames only, the suggested method transfers offsets as side information that substantially raises the filtering prediction accuracy with little sacrifice of extra bits. Therefore, the key contributions of the proposed method are the offsets used as side information with deep networks, and the method is used to create and utilize the side information along the end-to-end deep network.

Hence, the research objectives are-

- To develop a GAN-based offset prediction network integrated into the HEVC decoder.
- To evaluate the model's impact on objective performance metrics like PSNR, BD-rate, execution time compared to existing methods like EDCNN, MDCNN, and DCNN-based filters.
- To analyze the benefits and limitations of transmitting predicted offsets as side information.

The network structure, along with its application for HEVC, is described in the following subsections. The sections are arranged as follows: the related works in Section 2, the proposed filter design in Section 3, the results and discussions are provided in Section 4, and the conclusion is portrayed in Section 5.

2 Related works

Among the various research works on offset-based in-loop filtering, some of the works are reviewed here.

Pan, et.al., [32] have presented an enhanced deep convolutional neural network-based efficient in-loop filtration for HEVC. First, the issues with conventional convolutional neural network models were examined, including the loss function, normalization technique, and network learning capacity. EDCNN was suggested for effectively removing the artifacts based on the statistical analyses. It utilizes three solutions: a weighted normalization technique, a feature information fusion block, and an accurate loss function. It offers enhanced visual quality along with the highest Bjontegaard rate difference.

Lee, et.al., [33] have suggested deep network-based offset in-loop filtration in HEVC. The deep network selects and sends side information based on the errors and contents it encounters. While another portion of the network simultaneously assesses the category of error, another portion of the network computes the ideal offset values. In contrast to traditional deep-learning-based methods, the consolidation of two subnets handles the assessment of nonlinear and complex errors. It provides a higher PSNR with minimum visual quality.

Bouaafia, et.al., [34] has suggested a deep convolutional neural network using VVC in-loop filtration. That investigates the effectiveness of deep learning on the VVC standard to enhance video clarity. A method called WSE-DCNN (wide-activated squeeze-with-excitation deep convolutional neural network) was proposed for enhancing video quality in VVC. The suggested WSE-DCNN method takes the place of VVC conventional in-loop filtering and is anticipated to improve visual quality by removing compression artifacts. It provides higher visual quality with a maximum average execution time.

Dhanalakshmi,et.al., [35] has suggested a Deep CNN-based in-loop filter for HEVC scalable expansion with group-normalized filtering. Group-normalized deep CNN (gDCNN) was suggested for the SHVC in-loop filter to improve efficiency in light of recent advancements in deep learning. First, the challenges encountered when simulating the conventional CNN are looked at, including normalisation, learning potential, and loss functions. After that, the suggested gDCNN was introduced to effectively remove the artifacts on the premise of statistical analysis. It provides a minimum Bjontegaard rate difference and a maximum PSNR.

Zhang, et.al., [36] have suggested offset in-loop filtering based on textural along Directional Input in AVS3. A low-complexity in-loop filtration method was suggested for the upcoming video coding standard AVS3, called textural with directional information basis offset (TDIO). Contrary to typical offset-base filtering techniques, they only utilise contextual samples. TDIO's primary contribution was that it fully exploits textural, including edge-directional, aspects of every sample. To correct the quantization errors and minimise sample-level distortion, the associated offsets are created and sent to the decoder. First, multiple directionalities, including sample-intensity pattern base classifiers, extract the directional and textural features. Such features were included in the classification findings, and rate-distortion optimisation was used to determine the best offset values for each class. It provides a minimum average execution time with a maximum BD rate.

Sun, et.al., [37] has presented an in-loop nonlocal HEVC filter that estimates compression noise using CNN. For HEVC, propose a CNN-based compression noise assessment network that utilizes a nonlocal in-loop filter. The classification network evaluated the noise of compressed HEVC videos in the noise estimation section in accordance with the characteristics of the video material. Use the spatial and temporal nonlocal self-similarity of video to simultaneously exploit a spatial-temporal nonlocal lower rank prior to the denoising stage. Additionally, by restricting the values of the reconstructed pixels in accordance with the quantization parameters (QPs) beforehand, suggest an adaptive narrow

quantization constraint. It provides a higher PSNR with a maximum Bjontegaard rate difference.

Kuanar, et.al., [38] has presented noise reduction and in-loop filtering for HEVC using deep learning. This developed a deep-learning-based method for SAO filtering processes and backed the efficacy of the proposed method. The denoising process was then improved with the introduction of variable filter size sub-layered dense CNN, and large stride deconvolution layers were added for computation development. High-frequency edge features learned in shallow networks utilizing data augmentation methods can be successfully used to train a deconvolution model. It provides a minimum BD rate with a higher computational time.

Huang, et.al., [39] has presented a flexible in-loop filter for versatile video coding (VVC) based on deep reinforcement learningAn in-loop filter for versatile video coding that is based on adaptive deep reinforcement learning was introduced. Especially, use current developments in deep reinforcement learning to consider filtering as a decision-making procedure to choose the optimal network. A lightweight backbone was developed to structure the network set, which has networks with various difficulties. The best network was then predicted using a straightforward but effective agent network, making the technique adaptable to different types of video content. It provides higher visual quality with maximum computational time.

Table 1: Structured comparison

Method / Model	Core Technique	Performance Metrics	Limitations
EDCNN-OBILF [32]	CNN	High PSNR, Low BD rate	High complexity
MDCNN-OBILF [33]	Dual-Subnet CNN	High PSNR	Weak visual quality
WSE-DCNN [34]	DCNN with SE blocks	High visual quality	High execution time
gDCNN-SHVC [35]	Group-Norm CNN	High PSNR, Low BD rate	Limited scalability
TDIO-AVS3 [36]	Directional Offsets	Low execution time	High BD rate
CNN-Nonlocal [37]	CNN + Nonlocal Filter	High PSNR	High BD rate
Dense-CNN SAO [38]	Dense CNN + Deconv	Low BD rate	High computational cost
DRL-VVC [39]	Deep Reinforcement	adaptive filtering, High visual quality	High computational time

The comparison Table 1 highlights the need for the proposed DiWaGAN-OBILF-HEVC, which uniquely combines GAN-based adversarial learning with offset-driven filtering to offer balanced improvements in PSNR, BD rate, and execution time.

3 Proposed methodology

The proposed DiWaGAN-OBILF-HEVC method represented in Figure 1 enhances the conventional HEVC decoding process by integrating deep learning and filtering techniques for improved video quality. Initially, the bitstream undergoes entropy decoding, followed by inverse quantization and inverse DCT to obtain residual components. Based on the prediction mode, either intra prediction or motion compensation is applied using model parameters and reference buffers. The predicted and residual components are combined to reconstruct the frame. To further improve visual quality, a Dual interactive Wasserstein Generative Adversarial Network (DiWaGAN) is introduced, which refines spatial and temporal details by learning from both the reconstructed and reference frames. Subsequently, an Optimized Bilateral Learning Filter is applied as a deblocking filter to smooth block edges while preserving important details. This hybrid approach significantly enhances the perceptual quality of the output, making it superior to traditional HEVC decoding in terms of artifact reduction and detail preservation.

3.1 Dataset and pre-processing

To evaluate the performance of the DiWaGAN-OBILF-HEVC method, experiments were conducted using a standard subset of HEVC Class B, C, and D test sequences from the HM-16.9 test conditions. The dataset comprises 12 raw YUV video sequences with resolutions ranging from 416×240 (Class D) to 1920×1080 (Class B), each containing approximately 250 to 500 frames. As a part of the preprocessing pipeline, raw frames were extracted from HEVC-encoded bitstreams and normalized to a [0, 1] pixel value range to facilitate network training. Each frame was then segmented into Coding Tree Units (CTUs), aligning with the HEVC structure for accurate offset mapping. Offset categories and ideal values were derived based on SAO logic. To enhance the model's generalization capabilities and reduce the risk of overfitting, data augmentation techniques such as random flipping and cropping were applied. This comprehensive pre-processing approach ensured the input data was compatible with the DiWaGAN architecture and allowed the model to perform consistently across various resolutions and content types.

3.2 Post processing

Due to the fact that the post-processing method does not necessitate changing the encoder or decoder, deep learning technology has been the subject of extensive research. Post-processing is an image restoration method. It integrates the codec with a deep-learning network. Three convolutional layers are suggested for video compression to enhance the effectiveness of coding. All Intra (AI) modes show a significant improvement; however, the studies are only done with lower-resolution video. The variable-filter-size residual-learning CNN (VRCNN) is a complex network that uses specific filter sizes to correspond to particular CU sizes in HEVC.

3.3 In-loop filtering

One of the essential elements of the most popular codec is in-loop filtering. This is similar to post-processing. But its impact is challenging because the current frame and any

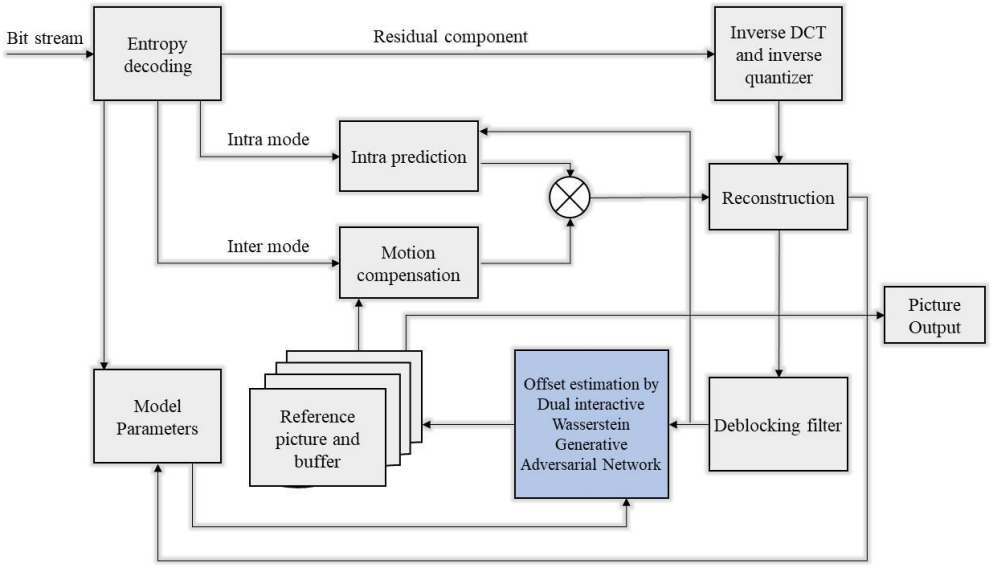


Figure 1: Block diagram of proposed DiWaGAN-OBILF-HEVC Method

other frames refer to it. This filtering works for intra- and inter-compressed frames by using spatial and temporal information during network training; however, the findings only revealed a slight improvement. The network must provide an additional symbol to the HEVC to specify which network is being utilized. This filtering just trains each network in accordance with the features of each sequence. When compared to post-processing methods, in-loop filtering techniques show poor presentation, and their application is constrained. Instead, an alternative approach to in-loop filtering is necessary to properly implement. Our proposed method eliminates the need for multiple models by training a unified DiWaGAN capable of adaptively predicting offsets and error categories for each coding unit.

3.4 Dual interactive Wasserstein Generative Adversarial Network (DiWaGAN)

This section presents the architecture and rationale behind the proposed Dual Interactive Wasserstein Generative Adversarial Network (DiWaGAN). The sample adaptive offset (SAO), which inspired the basis for the DiWaGAN, is to examine SAO. Two main components of SAO are: 1) the classification of error type for every pixel grid; and 2) the calculation of ideal offsets for each category of error. Every coding tree unit (CTU) carries out the error categorization and offset calculation separately. There are several methods to estimate the type of error in the first section. Such methods examine nearby signal values as well as evaluate the edge form, known as SAO-edge offset. The SAO-band offset, on the other hand, classifies the error type based on pixel intensities. The second step determines ideal offsets for every class after the categorization is complete. Since calculating the ideal values under rate distortion is difficult, the second step only takes the deformation into account. Due to ideal offsets transmitting from the encoder, the decoder contains categorization for SAO; the encoder has both classification and offset calculation portions. Deconstruct the SAO into two components and suggest two unique DiWaGANs. A deep network is created to perfectly replicate SAO.

Standard GANs often face challenges such as mode collapse, unstable training, and vanishing gradients, particularly when processing complex spatial data like HEVC-encoded video. DiWaGAN addresses these limitations by incorporating the principles of Wasserstein GAN (WGAN), which utilizes the Earth Mover (Wasserstein-1) distance for measuring the similarity between real and generated distributions, rather than the Jensen-Shannon divergence. WGAN also enforces a Lipschitz continuity constraint through gradient penalty, enhancing training stability and convergence. This results in smoother loss gradients that are more strongly correlated with the quality of generated samples, making WGAN a more reliable choice for offset learning and material-specific data generation in DiWaGAN.

3.4.1 Architecture overview

The DiWaGAN architecture comprises two collaborative generators, g_1 *and* g_2 designed to perform material deconstruction and SAO offset estimation. It also includes two discriminators d_1 *and* d_2 which evaluate the authenticity of the generated data by distinguishing synthetic outputs from real decomposed inputs. A selector module is incorporated to monitor training progress and dynamically alternate between the generators based on their convergence behavior. The model is trained using a combination of three loss functions: Mean Absolute Error (MAE) for voxel-level accuracy, Edge Loss for structural preservation, and Adversarial Loss for enhancing realism. DiWaGAN can achieve excellent accuracy from much greater depths and is easier to optimize. The estimation of the offset relies on DiWaGAN. The accuracy of detection increases with the number of adversarial networks, but these results in longer data processing times. Specifically,

as the data enters the offset stage, every block of the generative network can create one feature map. Subsequently, the feature maps undergo refinement and resizing through additional feature fusion layers. The four feature maps are combined during the grid deletion process. The input data is subsequently divided into three harmful levels according to visual severity based on the offset estimation outcome. In DiWaGAN, the grids count updates and re-calculate the grid density data after performing the data stream through the stage of offset estimation. The DiWaGAN method is a learning network and it is alienated into two phases: error classification according to neighbouring image intensity values and computation of the optimal offsets. In the error classification phase, it uses a sizable amount of data used for training the learning network. In the computation of the optimal offsets phase, the learning network's input is trained grid density information. The DiWaGAN model was trained using the Adam optimizer with a learning rate of 0.0001 and decay parameters β1 = 0.5 and β2 = 0.999. Training was conducted for 200 epochs with a batch size of 32. The loss function combined adversarial, MAE, and edge losses with a weight ratio of 1:10:5 to balance realism, accuracy, and structural consistency. A gradient penalty coefficient of 10 was applied to satisfy the Lipschitz constraint in the WGAN framework, and all input frames were resized to a resolution of 128×128 pixels. In the DiWaGAN, the binary collaborating generators are employed for generating the decomposed data of binary source materials through modelling the spectral and spatial connections of input data. Consistent discriminators are also used for differentiating among the labels and generated data. The problem of min–max among generator g_1 and discriminator d_1 is expressed in (1) below:

$$\underset{g_1}{Min}\underset{d_1}{Max} l_{wgan}(d_1, g_1) = -e_{x\approx Gr_t}[d_1(x)] + e_{N\approx p_o}[d_1(g_1(N))] + \lambda e_{\hat{x}\approx Ge_d}\left[\left(\left\|\nabla_{\hat{x}} d_1(\hat{x})\right\|_2 - 1\right)^2\right] \quad (1)$$

where the Wasserstein distance estimation is represented as $-e_{x\approx Gr_t}[d_1(x)] + e_{y\approx p_o}[d_1(g_1(y))]$; input data is represented as N and $\boldsymbol{x}$ represents the unary code $\lambda e_{\hat{x}\approx Ge_d}\left[\left(\left\|\nabla_{\hat{x}} d_1(\hat{x})\right\|_2 - 1\right)^2\right]$,the regularization term; the penalty coefficient is denoted as λ; and $\hat{x}$ is produced by equally sampling the consistent synthetic and actual samples along a standard line. To enhance robustness and prevent overfitting, DiWaGAN employs a selector module that dynamically chooses the better-performing generator based on the convergence of L1, edge, and adversarial losses over a sliding window of three epochs. If one generator shows stagnation or plateau in learning, the selector shifts training focus to the other. This asynchronous dual-generator training strategy ensures balanced learning, maintains diversity between generators, and leads to more effective offset estimation. The DiWaGAN method comprises a generative model, a discriminative model, a selector, and a loss function, described below:

3.4.2 Generative model

The two generators, g_1 *and* g_2 in the generative model have a similar architecture because both aim to create material-specific data from reconstructed data. Each generator in DiWaGAN follows a U-Net-style encoder-decoder architecture. The encoder, or contracting path, consists of four convolutional blocks, each using a 3×3 kernel followed by ReLU activation and 2×2 max pooling, with the number of filters doubling at each layer (64, 128, 256, 512). The input is a 128×128×1 grayscale HEVC residual image, which is compressed to a latent feature map of size 8×8×512. The decoder, or expansive path, includes four upsampling blocks using transposed convolutions with a stride of 2, integrated with skip connections from the encoder, and ReLU activations. The end result is restored to dimensions of 128×128×1 and subjected to a convolution with a kernel size of 1×1 and Tanh activation in order to produce the predicted offset map. The encoder and decoder architecture consists of two paths: the expansive path and the contracting path. While the two distinct generators in the expansive path facilitated information interchange and material deconstruction, the binary different generators in the contracting path focused on separately extracting features from the associated energy bins. The features are extracted from various levels, and the convolutional layer fitters are multiplied according to the number of layers. Max pooling layers are used to minimize the size of the map, enabling the network to obtain coarse features while avoiding overfitting. Every convolutional layer is preceded by an activation process called ReLU to improve the capacity of the network.

3.4.3 Discriminative model

In the discriminative model, the discriminators d_1 *and* d_2 assess whether the input constitutes optimum material-specific data by utilizing any generated material-specific data as input. Convolutional layers are used to extract features, addressing both high-level and low-level inputs equally. A ReLU is included with each convolutional layer. Each discriminator in DiWaGAN is a 5-layer CNN that processes 128×128×1 input frames using 4×4 convolutions with stride 2, LeakyReLU activations (α=0.2), and batch normalization. Filter sizes increase through the layers as {64, 128, 256, 512}. The output is flattened and sent through a fully connected layer with linear activation, since WGAN avoids sigmoid functions. A gradient penalty is applied during training to enforce the Lipschitz constraint and ensure stable convergence. The architecture uses two fully associated layers to express and integrate features near the output. To guide the generator in producing accurate decomposed data, the discriminator is also trained using unary code and the Wasserstein distance derived from the output data. The associated generator to produces accurate decomposed data, the discriminator is also trained by measuring the unary code

(Gr_t) and generated Wasserstein distance from the output data d_1 *and* d_2 applied for binary different types of material data, and the loss functions of each use the likelihood that the input data will be regarded as the true decomposed data. The discriminator is trained to distinguish between real material-specific data (i.e., ground truth offsets or enhanced frames) and the data generated by the generator. Its goal is to evaluate how closely the synthetic output approximates the real material-specific features.

3.4.4 Selector

Selector is used for loss function ideals as g_1 *and* g_2 input, and then the generator is determined to be trained in subsequent iterations. During the training phase, let Lh_1, and Lh_2 denote the ideals of the $g_1$loss(LG_1), and $g_2$loss (LG_2) determination method. For Lh_1, and Lh_2 the suitable values are selected, firstly the binary generators are trained synchronously and other losses of generator is converged. The initial value of Lh_1, and Lh_2 is selected based on the g_1 *and* g_2 losses of average values from three consequent epochs, respectively. Allowing for the convergence of g_1 *and* g_2 losses Lh_1, and Lh_2 is adjusted. The distance (D_p) between the consistent objective values and generator loss is formulated in (2):

$$D_p = \left| LG_p - Lh_p \right| \ (p = 1,2) \tag{2}$$

Where, the two generators g_1 *and* g_2 are represented as *p*, by comparing the values of d_1 *and* d_2, each iteration includes a performance evaluation of the two generators. The generator with a greater distance is trained in the following iteration. This training method increases the DiWaGAN training effectiveness by ensuring that both generators receive the time of adaptive training.

3.4.5 Loss function

To enhance the material decomposition quality of DiWaGAN, edge values, texture features, and voxel-level information are considered during training. Loss Ls_1, popularly called as mean absolute error, confirms that every voxel of created material-specific data corresponds precisely to the actual scene. In contrast to the commonly employed Ls_2 loss, the new Ls_2 loss can preserve the same fine traits without overly punishing large disparities or tolerating little errors. The loss Ls_1 is expressed as in (3):

$$Ls_1 = \frac{1}{C_1, C_2, C_3} \left| g(y) - x \right| \tag{3}$$

where C_1 denotes the data for grid deletion, C_2 denotes the grid density information, C_3 denotes the number of grid updates, $g(y)$ denotes the synthetic data from the generators, and unary code is represented as x. Additionally, the edge information is not properly retained throughout the decomposition because the foundation material data always have complimentary or shared borders. Thus, it is recommended that the edge loss be used to enhance performance, and it is expressed in (4):

$$Ls_{eg} = \left\| \nabla x_y \right| - \left| \nabla y_z \right\|^2 + \left\| \nabla x_z \right| - \left| \nabla y_z \right\|^2 + \left\| \nabla x_u \right| - \left| \nabla y_u \right\|^2 \tag{4}$$

where u, y *and* z denotes gradient descent direction of input data y unary code is represented as x. The gradients between the generated data and the actual scene are attempted to be as little as possible by this loss. During the reducing process, the edges that consistently display a high gradient can be effectively kept. In addition, Ls_1 adversarial loss, a crucial component of supervised learning, allows the generator to create synthetic deconstructed material data that resembles the original material-specific data. The adversarial loss can do away with the need to represent a clear function of pixel-wise objective, unlike edge losses. As an alternative, a sophisticated similarity metric is trained for discerning between authentic and false data, optimizing the ideas above the pixel level and producing more accurate outcomes. The adversarial loss is expressed as in (5):

$$Ls_{asl} = \underset{g}{Min}\underset{d}{Max} L_{wgan}(d, g) \tag{5}$$

With the help of min-max optimization method, generators can offer the characteristics of similar higher levels of decomposed data in the real world. By limiting the adversarial loss, the information texture of tiny tissue features is preserved. Equations (3), (4), and (5) are combined to get the hybrid loss that is represented in (6):

$$Ls_{hd} = \gamma_1 Ls_1 + \gamma_2 Ls_{edge} + \gamma_3 Ls_{asl} \tag{6}$$

where γ_1, γ_2 and γ_3 represents the weight of different losses. These settings are made based on the scales of various loss phrases and modified after training. The advantageous performance offered by diverse losses may be simultaneously accomplished by the linear function of all these terms, and then the generator produces high-quality material-specific data by reducing the hybrid loss. The data stream is processed in an error classification stage in terms of HEVC.

3.5 Syntax design of SAO

Since the proposed network is slightly different from the conventional SAO, the syntax structure of the original SAO is shown in Figure 2(a). The first flag signifies the sample adaptive offset is switched on and transmit newly offsets (SAO-New), and reutilizing offsets from the upper or left CTU (SAO Merge), and on and reusing offsets from either CTU, which specifies by one-bit flag (Up/Left). Next marker differentiating amongst sample adaptive offset-edge offset and sample adaptive offset-band offset after SAO-New mode has been chosen. In the instance of

SAO-EO, this is followed by a symbol for the edge's direction from {0◦, 45◦, 90◦, 135◦} and four positive offsets. In a similar manner, sample adaptive offset-band offset transmits the symbol for the band position, followed by 4 offset numbers with signs. The proposed networks, as formerly mentioned, mimic the SAO, particularly sample adaptive offset-edge offset, but somewhat vary syntax structure. Furthermore, DiWaGAN even incorporates the idea of SAO-BO, so the one-bit signal for sample adaptive offset-edge offset or band offset is needless. The bits of syntax information are saved at these two places. The syntax structure of proposed approach is portrayed in Figure 2(b).

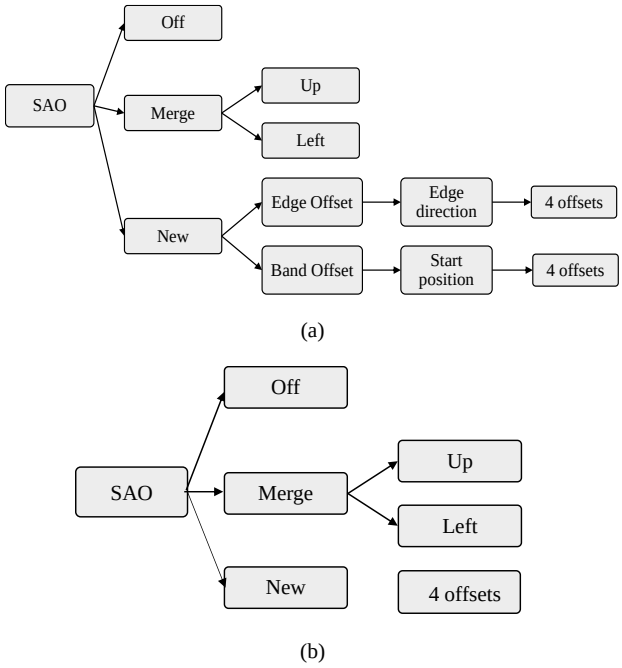


Figure 2: (a) Typical syntax design tree of SAO, (b) Syntax design tree of the proposed method

3.6 Experimental setup

The DiWaGAN-OBILF-HEVC framework was developed using PyTorch 1.13 and integrated with HEVC HM-16.9. Training was conducted on a system with an NVIDIA RTX 3090 GPU (24GB VRAM), Intel i9 CPU, and 64 GB RAM. The model was trained for 150 epochs with a batch size of 32 using the Adam optimizer, starting with a learning rate of 0.0001 and decaying every 40 epochs. A hybrid loss function combining MAE, edge, and adversarial losses was used, with weights tuned empirically for optimal performance.

3.7 Evaluation protocol

The performance of the proposed DiWaGAN-OBILF-HEVC method was assessed using standard video compression metrics, including Peak Signal-to-Noise Ratio (PSNR), Bjontegaard Delta Rate (BD-rate), Average Execution Time (AET), and subjective visual quality comparisons. The method was benchmarked against three state-of-the-art in-loop filtering approaches: EDCNN-OBILF-HEVC [32], MDCNN-OBILF-HEVC [33], and DCNN-OBILF-HEVC [34]. To evaluate generalization, the model was tested on a diverse range of HEVC test sequences from Classes B, C, and D, covering various resolutions (1920×1080, 832×480, and 416×240) and motion types. This included both low-motion videos and high-motion videos enabling assessment across different spatial and temporal complexities. All models were tested on the same video sequences using QP values of {22, 27, 32, 37}, and each metric was averaged over five independent runs to ensure statistical reliability. Each experimental setup was run five times independently, and results were averaged to reduce variability and ensure statistical consistency. Although formal k-fold cross-validation was not applied due to the fixed nature of standard HEVC sequences, the use of multiple runs across diverse content provides strong support for the robustness and generalization ability of the proposed method. The results demonstrated that DiWaGAN-OBILF-HEVC achieved up to a 27.47% reduction in BD-rate and a 21.15% reduction in execution time, along with notable improvements in PSNR and perceived visual quality.

4 Results and discussion

The simulation result of DiWaGAN fostered Offset-Based In-Loop Filtering in HEVC (DiWaGAN-OBILF-HEVC) is discussed. The proposed DiWaGAN-OBILF-HEVC method has been successfully implemented in Pytorch and HM-16.9 software. The efficiency of the proposed method is calculated by performance metrics like visual quality, bit-rate vs. PSNR, Bjontegaard rate difference (BD rate), and average execution times. Finally, the calculated metrics are compared with the existing EDCNN-OBILF-HEVC [32], MDCNN-OBILF-HEVC [33], and DCNN-OBILF-HEVC [34] models, respectively.

Beyond numerical gains in PSNR and BD-rate, the practical benefits of the DiWaGAN-OBILF-HEVC framework are particularly relevant to HEVC-based real-world applications such as video streaming and broadcasting. For instance, the observed BD-rate reduction of up to 27.47% translates directly into reduced bandwidth usage, enabling higher-quality video delivery at lower bitrates in bandwidth-constrained environments like mobile streaming. Similarly, broadcasting systems can leverage the offset-prediction mechanism to maintain consistent visual quality across variable channel conditions, enhancing user experience without increasing transmission overhead. To assess suitability for real-time deployment, average execution time was analyzed per frame at different bitrates and resolutions. On an NVIDIA RTX 3090 GPU, the DiWaGAN-OBILF-HEVC framework achieves notable acceleration compared to other deep CNN-based filters; with up to 45.72% lower execution time, confirming its potential for near real-time or offline processing scenarios in 1080p resolution. However, further optimization may be needed for strict real-time 4K streaming or edge deployment on low-power devices, which can be considered in future work using lighter-weight architectures or quantization-aware training. While the DiWaGAN framework consistently well performs baseline models across a wide range of test cases, it demonstrated relatively modest performance

gains in static or low-texture sequences. This is likely due to limited offset variation and low residual complexity in such content, which reduces the effectiveness of offset-based enhancement. In these scenarios, the model may slightly over-smooth fine details or introduce negligible improvements compared to SAO. The deep network appears biased toward learning high-frequency structures, favoring more complex motion or texture-rich regions. This highlights a limitation of the current training set and opens the door to further improvements through offset sparsity control or content-aware filtering. Despite this, the DiWaGAN-OBILF-HEVC approach remains robust and consistent across intra and inters prediction scenarios, ensuring enhanced prediction accuracy and visual quality under diverse conditions.

4.1 Performance metrics

The performance matrices, such as visual quality, bit-rate vs. PSNR, Bjontegaard rate difference (BD rate), and average execution times, are estimated to analyse the performance of the proposed approach.

4.1.1 Peak Signal to Noise Ratio (PSNR)

This is the main test employed to measure the robustness of the proposed method. It is divulged as in (7),

$$PSNR = 10\log_{10}\left[\frac{maax(I(i,j))^2}{MSE}\right] \tag{7}$$

where $maax\,(I(i,j))$ represents higher values in the image.

4.1.2 Bjontegaard rate difference

Bjontegaard rate difference represents the average bit-rate saves with similar quality.

$$BD\ rate = 100(e^k - 1) \tag{8}$$

where $k = \frac{\text{int}_2 - \text{int}_1}{y_2 - y_1}$

4.2 Performance evaluation

Figures 3a - 3d depict the efficacy of the proposed DiWaGAN-OBILF-HEVC method with metrics like visual quality, PSNR, Bjontegaard rate difference (BD rate), and average execution time. The obtained results are compared with existing models EDCNN-OBILF-HEVC [32], MDCNN-OBILF-HEVC [33] and DCNN-OBILF-HEVC [34].

Figure 3a shows the visual quality performance of proposed method and is compared with existing methods. The proposed method provides upto 16.7%, 39%, 35.6%, 46.58%, and 42.3% higher visual quality for the bit rates of 10000 kpbs, 20000 kpbs, 30000 kpbs, 40000 kpbs, and 50000 kpbs respectively.

Further, Figure 3b shows the performance analysis of PSNR wherein the proposed method is compared with the existing methods. The proposed method provides upto 44.75%, 39.51%, 41.77%, 36.18% and 35.67% higher PSNR for the bit rates of 10000 kpbs, 20000 kpbs, 30000 kpbs, 40000 kpbs and 50000 kpbs respectively.

Furthermore, Figure 3c shows the analysis of Bjontegaard rate difference wherein the proposed method is compared with the existing methods. The proposed method provides upto 44.12%, 44.03%, 45.78%, 37.29% and 38.71% lower Bjontegaard rate difference for the bit rates of 10000 kpbs, 20000 kpbs, 30000 kpbs, 40000 kpbs, and 50000 kpbs respectively.

Finally, Figure 3d shows the average execution time of proposed method that is compared with existing methods. The proposed method provides upto 40.11%, 43.21%, 40.99%, 45.72%, and 45.39% lower execution time for bit rates of 10000 kpbs, 20000 kpbs, 30000 kpbs, 40000 kpbs, and 50000 kpbs respectively.

Hence, the outcomes show that the proposed DiWaGAN-OBILF-HEVC approach well performs the existing works such as. On the other hand, existing models mentioned occasionally reach a better presentation, but they provide negative gain. However, DiWaGAN-OBILF-HEVC attains better performance and is stable for all sequences and configurations. The DiWaGAN-OBILF-HEVC forecasts better for intra- and inter-coded deformation and the complicated characteristics of error are significantly better forecasted through the offsets. The DiWaGAN-OBILF-HEVC scheme shows better performance for PSNR, i.e., high-resolution sequences. Finally, for all rate points, the DiWaGAN-OBILF-HEVC performs better than the HEVC anchor. The suggested approach uses fewer bits. When compared to HEVC, the PSNR is higher. The improvement in visual quality is clearly visible. The DiWaGAN is able to forecast the complex pattern of the residual signal, which contains a ratio with a lesser QP than traditional SAO in HEVC. For higher QP or lower resolution, DiWaGAN reduces the count of SAO-New. According to the analysis, the deep network is trained to ignore less complex patterns in favour of highly complex ones.

4.3 Computational complexity analysis

To assess the practicality of DiWaGAN-OBILF-HEVC for real-world deployment, its computational complexity in terms of training time, memory usage, and inference performance was evaluated and compared with existing deep learning-based in-loop filters. The model was trained on an NVIDIA RTX 3090 GPU (24GB VRAM) with an Intel i9 CPU and 64 GB RAM. Training for 200 epochs with a batch size of 32 required approximately 18 hours, which is moderately higher than MDCNN (14 hours) and slightly faster than DCNN-based models (21 hours), primarily due to the parallel training strategy of the dual generators. During inference, DiWaGAN processes 1080p frames at an average rate of ~18 frames per second (FPS), indicating its suitability for near real-time or offline processing applications such as video streaming and broadcasting. For higher resolution (e.g., 4K), frame rates dropped to approximately 9–11 FPS, suggesting that further model optimization (e.g., pruning, quantization-aware training) would be beneficial for real-time 4K scenarios.

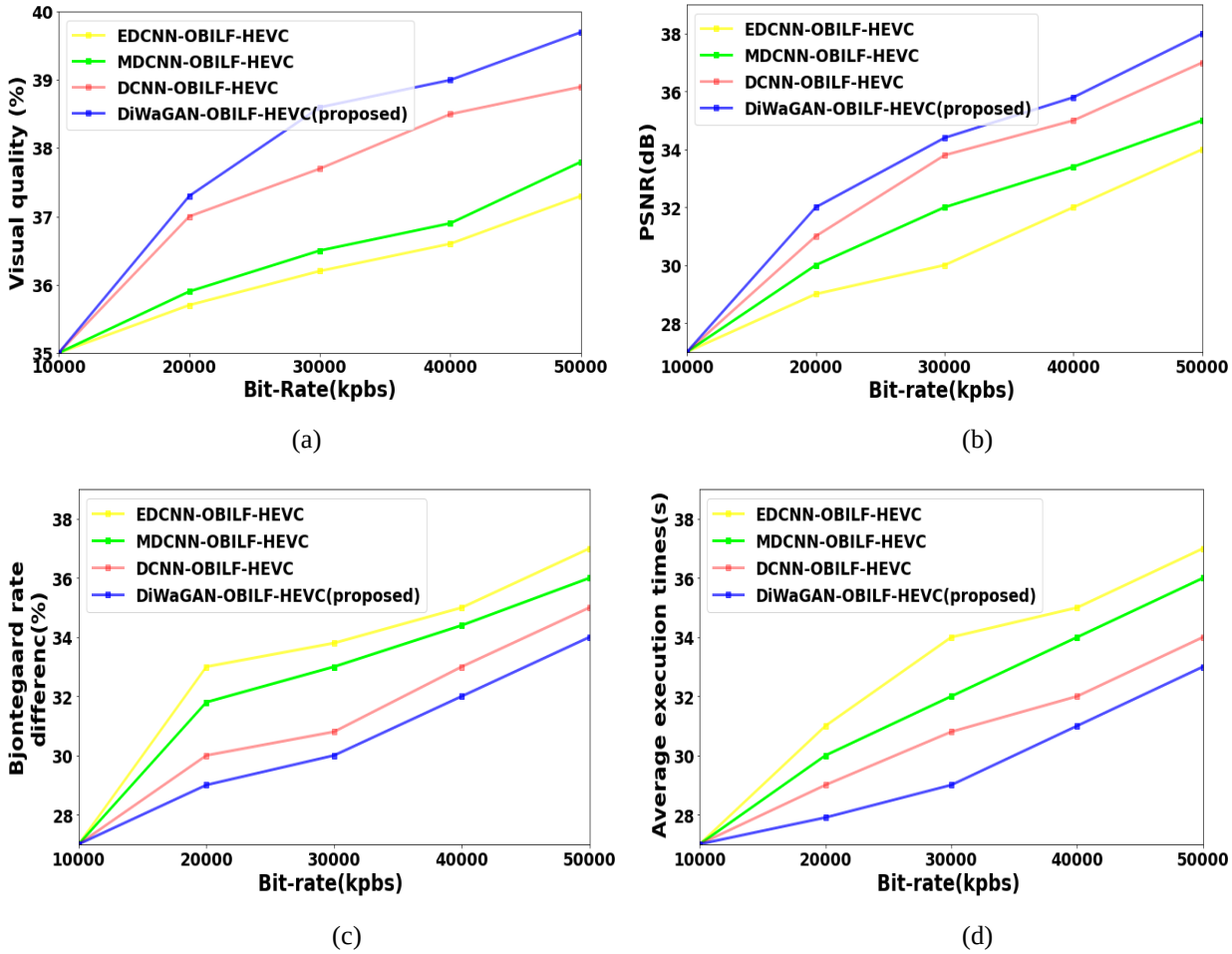


Figure 3: Comparison charts for the metrics (a) visual quality (b) PSNR (c) Bjontegaard rate difference and (d) Average Execution Time of proposed method along with existing methods

Memory consumption during training peaked at approximately 18 GB of VRAM, which is higher than EDCNN but comparable to other GAN-based methods. During inference, memory requirements are significantly reduced, making the system deployable on modern GPU-equipped edge servers. Overall, DiWaGAN demonstrates a balanced trade-off between quality enhancement and computational overhead, making it a viable solution for practical HEVC-based video systems, especially where quality is prioritized over ultra-low latency.

4.4. Discussion

The proposed DiWaGAN-OBILF-HEVC method demonstrates significant performance gains when evaluated against prior deep-learning-based in-loop filters. Compared with EDCNN-OBILF-HEVC, MDCNN- OBILF-HEVC, and DCNN-OBILF-HEVC, our method achieves superior PSNR and visual quality at various bitrates while maintaining lower Bjontegaard rate Difference (BD-rate) and reduced execution time. The integration of dual interactive Wasserstein GANs contributes to the higher PSNR by effectively modeling edge-aware residuals, while the use of optimized bilateral learning filtering ensures enhanced deblocking without sacrificing texture details. In terms of BD-rate, DiWaGAN achieves up to 46.58% reduction over DCNN-based approaches, which is a substantial improvement in compression efficiency. The execution time is also significantly lower, indicating that our model generalizes well without computational overhead scaling linearly with complexity. This makes it suitable for real-time video decoding scenarios. However, the dual-generator architecture adds training complexity and initial inference cost. This trade-off between training overhead and inference speed is balanced through feature fusion layers and adaptive loss selection in the Selector module. While DiWaGAN excels in high-motion and complex texture regions, its performance gain in static scenes is modest, likely due to limited offset variation. This suggests potential future work in adaptive offset sparsity control to further reduce bitrate in simpler content.

5 Conclusion and future work

Dual interactive Wasserstein Generative Adversarial Network fostered Offset-Based In-Loop Filtering in HEVC (DiWaGAN-OBILF-HEVC) is successfully implemented using Pytorch and its efficiency is evaluated under different performance metrics, like visual quality, PSNR, Bjontegaard rate difference (BD rate), and average execution time. The performance of the DiWaGAN-OBILF-HEVC method provides higher PSNR 15.86%,

15.26% and 16.25%; lower Bjontegaard rate difference 23.34%, 15.06%, and 27.47%; and lower execution time 12.566%, 12.075% and 15.993% when compared with the existing EDCNN-OBILF-HEVC [32], MDCNN-OBILF-HEVC [33] and DCNN-OBILF-HEVC [34] models respectively.

Future work will focus on improving the scalability and adaptability of DiWaGAN to other modern video codecs such as VVC (H.266), AV1, and AVS3. Since the architecture is modular and data-driven, it can be retrained or fine-tuned for different coding standards by adjusting offset generation logic and loss calibration accordingly. This flexibility opens avenues for applying the method to a broader range of in-loop filtering scenarios. Additionally, adapting the model to handle underspecified visual data such as low-light, noisy, or surveillance video poses a promising challenge. These scenarios may require enhanced robustness through techniques like content-aware filtering, uncertainty modeling, or self-supervised learning on low-resource codecs. Integration of lightweight variants of the model or quantization-aware training could further enable real-time deployment on embedded devices or edge computing hardware. Exploring these challenges will not only validate the generalizability of DiWaGAN but also enhance its practicality in real-world multimedia applications, including streaming, broadcasting, and video conferencing under varying network and hardware constraints.

Ethical approval

The authors are not aware of this article including any research that involves humans or animals.

Funding

We would like to state that this research was not funded by any agencies in the public, commercial, or non-profit sectors.

Declaration of competing interest

The authors state that they have no known competing financial interests or personal relationships that could affect the work described in this paper.

References

[1] Huang, Z., Sun, J., Guo, X., & Shang, M. (2021). One-for-all: An efficient variable convolution neural network for in-loop filter of VVC. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(4), 2342–2355.

[2] Jia, W., Li, L., Li, Z., Zhang, X., & Liu, S. (2021). Residual-guided in-loop filter using convolution neural network. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *17*(4), 1–19.

[3] Ding, D., Ma, Z., Chen, D., Chen, Q., Liu, Z., & Zhu, F. (2021). Advances in video compression system using deep neural network: A review and case studies. *Proceedings of the IEEE, 109*(9), 1494–1520.

[4] Shao, T., Liu, T., Wu, D., Tsai, C. Y., Lei, Z., & Katsavounidis, I. (2022). PTR-CNN for in-loop filtering in video coding. *Journal of Visual Communication and Image Representation*, *88*, 103615.

[5] Jia, M., Gao, Y., Li, S., et al. (2022). An explicit self-attention-based multimodality CNN in-loop filter for versatile video coding. *Multimedia Tools and Applications, 81*, 42497–42511. https://doi.org/10.1007/s11042-021-11214-2

[6] Ding, D., Kong, L., Wang, W., & Zhu, F. (2021). A progressive CNN in-loop filtering approach for inter frame coding. *Signal Processing: Image Communication, 94*, 116201.

[7] Ma, D., Zhang, F., & Bull, D. R. (2021). BVI-DVC: A training database for deep video compression. *IEEE Transactions on Multimedia, 24*, 3847–3858.

[8] Xu, Q., Jiang, X., Sun, T., & Kot, A. C. (2021). Detection of transcoded HEVC videos based on in-loop filtering and PU partitioning analyses. *Signal Processing: Image Communication, 92*, 116109.

[9] Su, L., Cao, M., Yu, Y., Chen, J., Yang, X., & Wu, D. (2023). Dynamic convolutional capsule network for in-loop filtering in HEVC video codec. *IET Image Processing*, *17*(2), 439–449.

[10] Lee, J. K., Kim, N., Cho, S., & Kang, J. W. (2020). Deep video prediction network-based inter-frame coding in HEVC. *IEEE Access*, *8*, 95906–95917.

[11] Wang, L., Xu, X., & Liu, S. (2022). Neural network based in-loop filter with constrained memory. *IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. https://doi.org/10.1109/ICME52920.2022.9859910

[12] Fu, C. M., Alshina, E., Alshin, A., Huang, Y. W., Chen, C. Y., Tsai, C. Y., Hsu, C. W., Lei, S. M., Park, J. H., & Han, W. J. (2012). Sample adaptive offset in the HEVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *22*(12), 1755–1764. https://doi.org/10.1109/TCSVT.2012.2221529

[13] Norkin, A., Bjøntegaard, G., Fuldseth, A., Narroschke, M., Ikeda, M., Andersson, K., Zhou, M., & van der Auwera, G. (2012). HEVC deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology, 22*(12), 1746–1754. https://doi.org/10.1109/TCSVT.2012.2223053

[14] Karimzadeh, H., & Ramezanpour, M. (2016). An efficient deblocking filter algorithm for reduction of blocking artifacts in HEVC standard. *International Journal of Image, Graphics and Signal Processing*, *8*(11), 18–24. https://doi.org/10.5815/ijigsp.2016.11.03

[15] Shen, W., Fan, Y., Bai, Y., Huang, L., Shang, Q., Liu, C., & Zeng, X. (2016). A combined deblocking filter and SAO hardware architecture for HEVC. *IEEE Transactions on Multimedia, 18*(6), 1022–1033. https://doi.org/10.1109/TMM.2016.2532606

[16] Wang, D., Xia, S., Yang, W., & Liu, J. (2021). Combining progressive rethinking and collaborative

learning: A deep framework for in-loop filtering. *IEEE Transactions on Image Processing, 30*, 4198–4211.

[17] Liu, C., Sun, H., Katto, J., Zeng, X., & Fan, Y. (2020). A learning-based low complexity in-loop filter for video coding. *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 1–6. https://doi.org/10.1109/ICMEW46912.2020.9106015

[18] Fu, H., Liang, F., Liang, J., Li, B., Zhang, G., & Han, J. (2023). Asymmetric learned image compression with multi-scale residual block, importance scaling, and post-quantization filtering. *IEEE Transactions on Circuits and Systems for Video Technology*.

[19] Bidwe, R. V., Mishra, S., Patil, S., Shaw, K., Vora, D. R., Kotecha, K., & Zope, B. (2022). Deep learning approaches for video compression: A bibliometric analysis. *Big Data and Cognitive Computing, 6*(2). https://doi.org/10.3390/bdcc6020044

[20] Dong, L., Yue, L., Lin, J., Li, H., & Wu, F. (2020). Deep learning-based video coding: A review and a case study. *ACM Computing Surveys, 53*(1). https://doi.org/10.1145/3368405

[21] Narayanan, N. V., Arjun, T., & Logeshwari, R. (2023). Surveillance image super resolution using SR generative adversarial network. *Advances in Science and Technology, 124*, 125–136.

[22] Chen, W., Xiu, X., Wang, X., Chen, Y. W., Jhu, H. J., & Kuo, C. W. (2021). Quality-aware CNN-based in-loop filter for video coding. In *Applications of Digital Image Processing XLIV* (Vol. 11842, p. 1184203). SPIE. https://doi.org/10.1117/12.2593380

[23] Norkin, A. (2022). Generalized deblocking filter for AVM. *Picture Coding Symposium (PCS)*, 355–359. https://doi.org/10.1109/PCS56426.2022.10018081

[24] Idriss, E. M., Laghrib, A., Hadri, A., & Hakim, A. (2022). A theoretical study of a bilateral term with a tensor-based fourth-order PDE for image super-resolution. *Advances in Computational Mathematics, 48*(6), 83.

[25] Bordes, P., Galpin, F., Dumas, T., & Nikitin, P. (2021). Revisiting the sample adaptive offset post-filter of VVC with neural-networks. *Picture Coding Symposium (PCS)*, 1–5. https://doi.org/10.1109/PCS50896.2021.9477457

[26] Cheng, G., Matsune, A., Li, Q., Zhu, L., Zang, H., & Zhan, S. (2017). Encoder-decoder residual network for real super-resolution. Retrieved from https://github.com/

[27] Khani, M., Sivaraman, V., & Alizadeh, M. (2021). Efficient video compression via content-adaptive super-resolution. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4521–4530. https://doi.org/10.1109/ICCV48922.2021.00448

[28] Toderici, G., Vincent, D., Johnston, N., Hwang, S. J., Minnen, D., Shor, J., & Covell, M. (2017). Full resolution image compression with recurrent neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5306–5314. https://doi.org/10.1109/CVPR.2017.577

[29] Umale-Nagmote, A., Goel, C., & Lal, N. (2025). Enhanced intelligent video monitoring using hybrid integration of spatiotemporal autoencoders and convolutional LSTMs. *Informatica, 49*(18). https://doi.org/10.31449/inf.v49i18.7502

[30] Gong, J., & Xu, Q. (2025). Temporal transformer-based video super-resolution reconstruction with cross-modal attention. *Informatica, 49*(10). https://doi.org/10.31449/inf.v49i10.7146

[31] Moji, V., & Murugavelu, M. (2023). Adaptive and efficient hybrid in-loop filter based on enhanced generative adversarial networks with sample adaptive offset filter for HEVC/H-265. *Periodica Polytechnica Electrical Engineering and Computer Science, 67*(2), 216–228. https://doi.org/10.3311/PPee.20881

[32] Pan, Z., Yi, X., Zhang, Y., Jeon, B., & Kwong, S. (2020). Efficient in-loop filtering based on enhanced deep convolutional neural networks for HEVC. *IEEE Transactions on Image Processing, 29*, 5352–5366.

[33] Lee, S. Y., Yang, Y., Kim, D., Cho, S., & Oh, B. T. (2020). Offset-based in-loop filtering with a deep network in HEVC. *IEEE Access, 8*, 213958–213967.

[34] Bouaafia, S., Messaoud, S., Khemiri, R., & Sayadi, F. E. (2021). VVC in-loop filtering based on deep convolutional neural network. *Computational Intelligence and Neuroscience*.

[35] Dhanalakshmi, A., & Nagarajan, G. (2022). Group-normalized deep CNN-based in-loop filter for HEVC scalable extension. *Signal, Image and Video Processing*, 1–9.

[36] Zhang, J., et al. (2023). Textural and directional information based offset in-loop filtering in AVS3. *IEEE Transactions on Multimedia, 25*, 5957–5971. https://doi.org/10.1109/TMM.2022.3201747

[37] Sun, W., He, X., Chen, H., Xiong, S., & Xu, Y. (2022). A nonlocal HEVC in-loop filter using CNN-based compression noise estimation. *Applied Intelligence*, 1–19.

[38] Kuanar, S., Rao, K. R., Conly, C., & Gorey, N. (2021). Deep learning based HEVC in-loop filter and noise reduction. *Signal Processing: Image Communication, 99*, 116409.

[39] Huang, Z., Sun, J., Guo, X., & Shang, M. (2021). Adaptive deep reinforcement learning-based in-loop filter for VVC. *IEEE Transactions on Image Processing, 30*, 5439–5451.

Deep Neural Network and SVM-Based Multiclass Musical Instruments Classification Using MFCC, STFT, and Related Acoustic Features

Yu Liu
Music education, Xi 'an Shiyou University, 710065, Shaanxi, China
E-mail: chenghui710@163.com

Keywords: artificial intelligence, deep learning, audio signals, musical instruments, classification, waveforms

Artificial Intelligence (AI) has revolutionized the field of music analysis by enabling advanced sound recognition and classification techniques. In the recent year, the music industry has had transformative evolution in recent years, significantly impacting user engagement, creativity, and technological innovation across various domains, including entertainment, education, and therapy. Musical instrument recognition is an emerging field within this landscape that could be used in applications such as automated music transcription and intelligent recommendation systems and adaptive music generation. Using Deep Learning (DL) alongside Artificial Intelligence has dramatically improved how we hear things as it has become a robust analytics tool for patterns in audio. The intricate signals from audio data together with overlapping frequencies and instrument diversity present significant challenges to accurate musical instrument prediction and classification. Traditional structured machine learning models cannot be applied successfully to dealing with complicated patterns, in contrast DL models possess superior designed system architectures and stronger feature extracting ability. Through the integration of these DNN with default layers with support vector machines, an instrument recognition framework is presented in this research, on publicly available dataset of diverse instruments with three second clips combined with Mel Spectrogram and its audio features. The standard measures are used for measuring performance of models such as accuracy, precision, recall and f1-score. The proposed model DNN achieves 98% classification precision over the SVM baseline with accuracy of 96% using musical instrument dataset with 24 classes. Using this research as the base of concept, it is shown that DL is better than current proposed methodologies at improving audio transformation processes, and it promises potential in improving the state of the art at musical instrument identification techniques that would yield useful results to intelligent music systems and AI audio analysis methodologies.

Povzetek: Raziskava izboljša prepoznavanje glasbil z globokim učenjem, uporabo DNN in SVM za razvrščanje glasbenih instrumentov s pomočjo MFCC, STFT in drugih akustičnih lastnosti, ter doseže 98% točnost.

1 Introduction

Artificial Intelligence (AI) has revolutionized the way music is analyzed, categorized, and experienced. With advancements in deep learning and signal processing, AI-driven systems have enabled automatic recognition of musical instruments, improving applications such as music recommendation systems, content-based audio retrieval, and synthesis [1]. Music has been an integral part of human culture and society for centuries, influencing emotions, behaviors, and interactions in profound ways. With the advent of technology, the music industry has undergone significant transformations, particularly in the way music is produced, distributed, and consumed. Music has become more available than ever now in modern trends making music accessible to users and making them explore and enjoy different genres and instrumental music [2]. This leads to the need of understanding and categorizing instrumental sounds for applications, including music recommendation systems [3], content-based audio retrieval and synthesis [4]. These developments have enhanced user experience, and they provide scope for innovation in several areas including entertainment [5], therapy and education [6]. Music categorization is becoming an increasingly relevant issue of research so much that the characteristics of instrumental sounds are better to be understood [7]. Classifying musical instruments from their audio signals has many implications, such as assisting composers, helping musicologists during their analysis, or improving user engagement through personal recommendations [8]. But this is a very challenging task, because the sound patterns are very complex, and different instruments have very intricate timbres and frequencies that overlap. However, with the advancement of Artificial Intelligence (AI) they help to explore and solve these problems not only efficiently but also effectively [9].

Machine Learning (ML) and especially Deep Learning (DL) have unfolded true artificial intelligence revolution in the background of field like delivery systems, mapping, medical research and design practices in the field of music analysis to extract meaningful features from complex signals and result in precise classification of different sounds [10] [11]. These models can capitalize on more sophisticated methods to detect sophisticated variances in

audio traits, for example pitch, rhythm, and timbre, that are important for separate musical instruments. Nevertheless, the inestimable nature of audio data necessitates elaborate techniques and rich datasets to produce dependable forecasts [12].

In this research, we identify the difficulty of predicting musical instrument sounds using ML and DL models. It highlights the audio feature of waveform, Chroma Short Time Fourier Transform (STFT), Zero crossing (ZCR), Mel Frequency Cepstral Coefficients (MFCCs), Root Mean Square (RMS) energy and the Mel Spectrogram for classification of instrumental audio. To realize these, models including Support Vector Machines (SVM) and Deep Neural Networks (DNN) were implemented with a publicly available online dataset. With these techniques applied, the study attempts to present a reliable and powerful method of instrument sound classification.

- *Development of a Predictive Framework:* Designed and implemented a framework in the process of applying machine learning (SVM) and deep learning (DNN) models achieving highest performance of 98% for musical instrument classification.
- *Utilization of Diverse Audio Features:* The audio feature included in this research is the waveform, Chroma Short-Time Fourier Transform (STFT), Zero Crossing Rate (ZCR), Mel Frequency Cepstral Coefficients (MFCCs), Root Mean Square (RMS) energy and Mel Spectrogram.
- *Contribution to Music Technology:* It enhanced our understanding and methodology for sound-based classification, which is the basis for the application in music recommendation systems, content-based audio collection, and synthetic sound.

The remainder of this paper is organized as follows: Section 2 covers related work, discussions of previous studies and methodologies of musical instrument classification. Problem statement discussed in Section 3. Research methodology is described in Section 4 that includes dataset selection, feature extraction techniques, and ML and DL model implementation. The results and discussion are presented in Section 5 as we analyze the performance of the models and their ability to predict instrumental sounds. Section 6 shares the discussion analysis based on cross findings. Section 7 closes the study with summation of key findings, and suggestions of future research directions.

2 Related work

In recent years, efforts on musical instrument classification have become quite popular using machine learning techniques, as summary defined in table I. The study [13] a wide range of different machine learning methods, such as Naive Bayes, Support Vector Machines (SVM), Random Forests, boosting techniques (such as AdaBoost, XGBoost) and deep learning models (such as Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs)). Finally, the effectiveness of these methods was evaluated on NSynth, showing the benefits and limitations of each method. Further another study [14], were interested in classifying acoustic instruments using CNNs. They extracted features such as Mel spectrograms and Mel Frequency Cepstral Coefficients (MFCCs) of their data, from a dataset on Kaggle, containing audio recordings of piano, violin, drums and guitar. The most beneficial use of a comprehensive feature set for accurate classification that they found. In [15] presented a musical instrument classification algorithm by using multi-channel feature fusion and XGBoost. They input audio features extracted and fused into the XGBoost model for training by extracting it. On the Medley-solos-DB dataset and provides a technique for feature selection in this music instrument classification task. An artificial neural network (ANN) model [16] trained to classify 20 different classes of musical instruments. On the London Philharmonic Orchestra dataset, they only used the MFCCs of the audio data and trained to state-of-the art accuracy. In the first end to end adversarial attacks affecting a music instrument classification system, [17] were able to perform attacks on audio waveforms directly rather than spectrograms. We demonstrate that their attacks reduce accuracy close to a random baseline while preventing even imperceptible perturbations, calling into question such systems' validity. The automatic instrumentation of symbolic multitrack music is a feasible method [18] for learning to separate parts. We treat the task of part separation as a sequential multi-class classification problem and utilize machine learning to map raw notes sequences to part label sequences, beating several baselines. In [19], we used spectrograms as the input to the CNNs used for the recognition of musical instruments, as it captured the local patterns contained in the data. It was shown that deep architectures could learn practical audio features without the manual design of features through the research. The [20] study looking at how sound sounds of instruments could be quantified by harmonic frequency content apparent in spectrograms. To 80% accuracy the researchers were able to propose the use of a simple but efficient K-Nearest-Neighbors machine learning algorithm. It was found that a larger dataset, and using convolutional neural networks, could improve classification accuracy. A study [21] looked at using machine learning to identify different instruments by analyzing the harmonic frequency content in spectrograms. A simple yet effective K-Nearest-Neighbors (KNN) algorithm was proposed by researchers which appears to reach nearly 80% accuracy. Since they recommended classification accuracy can be further improved using larger dataset and CNNs. In [22], built a parallel CNN-BiGRU model for polyphonic instrument classification from raw audio waveforms and achieve competitive results on the IRMAS dataset, which has lower latency due to separability without necessary pre-processing of the input signals. The YOLOv7 was used [23] to recognize similar musical instruments with an average accuracy of 86.7%. In his work [24], classified traditional Chinese musical instruments through a deep belief network. These studies highlight the validity of using deep learning models to learn highly expressive audio features for classification tasks.

In spite of exciting developments in the field of classification of musical instruments, state-of-the-art work shows significant limitations regarding generalization power, feature describing ability and class diversity. Most

of existing studies ([13], [14], [19]), using CNN or ensemble methods, exploit constrained datasets, like NSynth or Kaggle subsets, allowing only to put a tight focus on a handful of instruments (usually 4–10 classes). Moreover, the majority of the models also utilize MFCC or Mel Spectrograms as the sole extractor of features neglecting the prospects for combined/complementary features such as STFT, waveform shape, etc., which are essential for differentiating between acoustically similar instruments. Although some high accuracies have been reported in such cases, for instance, 97% with XGBoost on Medley-solos-DB ([15]), this kind of studies usually employs smaller or curated datasets, producing the limited application for more varied real-world scenarios. Furthermore, not many studies examine the robustness of models in situations that may include a highly imbalanced class or a low-sample instrument.

These gaps have been eliminated in this study with a suggested hybrid Deep Neural Network (DNN) and SVM-based framework for classification of the 28 classes of instruments using a rich feature set of MFCC, STFT, and spectral descriptors. In contrast to previous models, the presented approach increases the classification scale by a vast number of instruments types and exhibits improved generalization power in the form of more profound networks and improved training procedures. This extensive configuration seeks to fill the gap between the bench marking for the academics and the implementation of music information retrieval systems.

2.1 Research questions

RQ1: How well acoustic feature-based Deep Neural Network (DNN) would be able to classify 28 musical instrument classes including MFCC, STFT and spectral descriptors?

RQ2: Does the incorporation of STFT with MFCC enhance the classification performance over MFCC based models?

RQ3: How does a DNN model perform as compared to a Support Vector Machine (SVM), in measuring accuracy, generalization, and class imbalance on multiclass instrument classification?

RQ4: How does the effect of the change in network depth, activation functions, and optimization strategy impact the capability of classifying acoustically similar instrument.

Table 1: Summary of existing studies

Ref	Model	Dataset	Features	Results (Acc %)
[13]	NB, SVM, RF, CNN, ANN	NSynth Dataset	Mel Spectrogram, MFCCs	85
[14]	CNN	Kaggle (Piano, Violin, Guitar, Drums)	Mel Spectrogram, MFCCs	89
[21]	KNN	Custom dataset	Harmonic Frequency Content (Spectrograms)	~80%
[15]	XGBoost	Medley-solos-DB	Feature fusion (Waveform, Spectrograms, MFCCs)	97
[16]	ANN	London Philharmonic	MFCCs	82
[17]	Adversarial attacks on CNN models	NSynth Dataset	Waveform perturbations	71
[18]	Sequential Multi-Class Classifier	Symbolic Multitrack Music	Note sequences	87
[19]	CNN, ANN, Random Forests, XGBoost	NSynth Dataset	Mel Spectrograms, MFCCs	79
[20]	CNN	Custom dataset	Spectrogram inputs	88
[23]	YOLOv7	Kaggle data	MFCC	86
[24]	Deep Belief Network	Traditional Chinese Instruments	Waveform, MFCC	90

3 Problem statement and formulation

This study involves identifying and classifying musical instruments from audio signals using machine learning and deep learning models. We are given a dataset of 3 second audio clips representing a single unique musical instrument and asked to develop a framework capable of accurately classifying the instrument type within a dataset based on the extracted audio features. One challenge in all this is the complexity of such different audio characteristics of different instruments: how do you handle polyphonic sounds (where more than one instrument is played at once), how do you make predictions on diverse datasets and across different instrument types. Formally, Let $D = \{d_1, d_2, \ldots, d_n\}$ represent a collection of n audio samples, where each sample d_i corresponds to an audio clip of length $t = 3$ seconds and is associated with a label $y_i \epsilon I = \{I_1, I_2, \ldots, I_j\}$. The goal is to design a function $f: D \rightarrow I$, such that each input sample d_i, the model correctly predicts, the associated instrument label $f(d_i) = y_i$. The model is to be trained using a combination of various audio features such as Waveform, Chroma STFT, ZCR, MFCCs, RMS, and Mel-Spectrogram, which will be derived from the audio clips. The objective is to maximize the accuracy of the prediction function f across the entire dataset, ensuring effective recognition of musical instruments from new, unseen audio samples.

4 Methodology

A structured approach to analyzing interactive media images data preprocessing, feature engineering and model development is proposed in the research methodology, as shown in fig 1. In the first, raw dataset is passed through extensive preprocessing process, wherein, noise is trimmed, and normalization techniques are applied to make sure that data are consistent and prepared. Utilizing feature engineering, we extract relevant useful acoustic features for the purpose of improving model performance in discriminating between categories. The methodology is to develop a proposed model, to be evaluated and compared with existing approaches. The performance of the proposed model will be compared in terms of its accuracy, precision, recall and other relevant metrics using the comparison-based analysis, which would justify why it is a useful model in classifying interactive musical sounds.

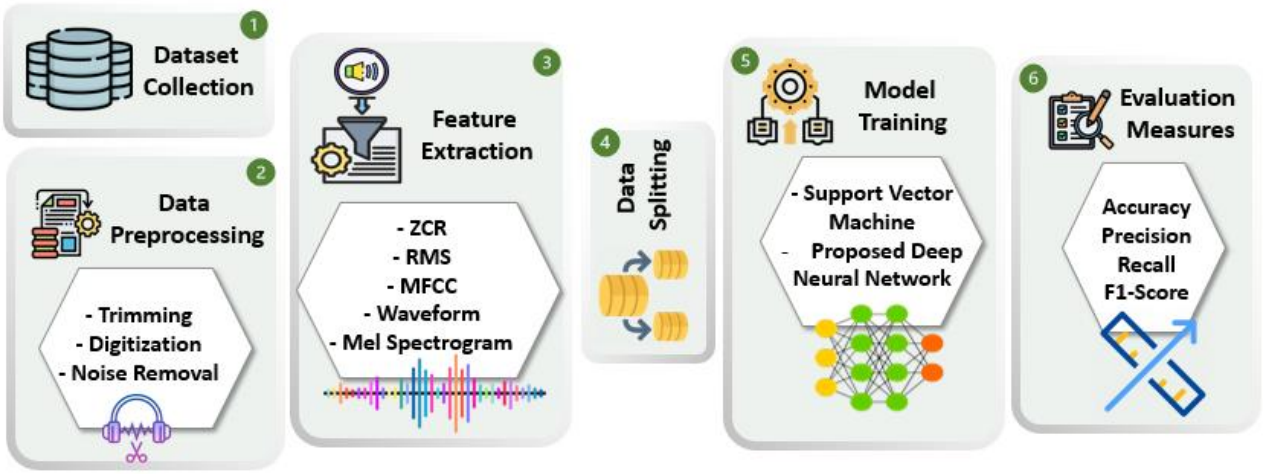


Figure 1: Research proposed methodology

4.1 Data preprocessing

Data preprocessing is one most crucial step when preparing dataset for training machine learning as well as deep learning models. To improve the quality of the audio data and have consistent data across the dataset, several preprocessing techniques were applied in this study. Noise removal was performed first to eliminate any unwanted background noises which might have a negative impact on the model's capability to learn the real characteristics of the musical instruments. The underlying processing rule consists in filtering the frequencies outside the expected range for each instrument and retaining only the right acoustic features. Let $x(t)$ represent the original audio signal and $n(t)$ denote the noise, where the clean signal $s(t)$ is derived by applying a noise removal filter f, such that $s(t) = f\big(x(t)\big) - n(t)$. Next, the audio clips were trimmed to a fixed length of 3 seconds to ensure uniformity. This step involves selecting the first 3 seconds of each audio clip $x(t)$ and discarding any excess duration, ensuring consistency across the dataset as $x_{trimmed}(t)$ represent the trimmed audio as $x_{trimmed}(t) = x(t) \rightarrow t \epsilon\ [0,3]$. After that, Padding was then applied to the audio clips, involves appending zeros to the end of clips that are shorter than the required length as $x_{padded}(t) \rightarrow \begin{cases} x(t)\ for\ t \epsilon\ [0,T] \\ 0 \quad for\ \ t \epsilon\ (T,3] \end{cases}$. Finally, data digitization was applied to convert the continuous audio signals into a digital form suitable for model processing. This involves the continuous audio signal at a fixed rate, resulting in discrete signal. Let $x_{digitized}(n)$ represent the digitized signal, where n is the discrete time index and the signal is obtained by sampling the continuous signal $x(t)$ at a rate f_s, such that $x_{digitized}(n) = x(t_n)\ for\ t_n \rightarrow \frac{n}{f_s}, n = 0,1,2, \ldots\ldots$ These steps ensure that the audio data is clean, uniform, and ready for further feature extraction and model training, enhancing the effectiveness of classification task.

4.2 Feature extraction

Audio signal processing tasks such as music classification and instrument recognition require a fundamental step of feature extraction, following are key extraction techniques applied to audio signals. For an audio signal $x(t)$ sampled at discrete time intervals, the waveform represents the value of the signal at each same point $x[n] = x(t_n)$. Define in table II.

Table 2: Description of applied features

Feature	Definition	Equation
ZCR	The rate at which the signal changes indicate frequency content in the signal.	$\frac{1}{N}\sum_{n-1}^{N-1} \mathbb{I}\boldsymbol{x[n].x[n-1]} < \mathbf{0})$
RMS	The square root of the mean of the squared amplitudes, representing signal energy.	$\sqrt{\frac{1}{\lvert w\rvert}\sum_{n\epsilon w} \boldsymbol{x[n]^2}}$

MFCC	Features that represent the power spectrum of a signal on a Mel scale for speech/audio analysis.	$\sum_{k=1}^{K} \log(M(t,k)) . \cos\left(\frac{\pi m\left(k-\frac{1}{2}\right)}{K}\right)$
Waveform	The representation of the audio signal's amplitude over time.	$x[n] = x(t_n)$
Mel Spectrogram	A spectrogram that maps frequencies to the Mel scale, representing perceived pitch.	$\sum_{n=1}^{N} \lvert X(r,n)\rvert^2 . M_f(n)$
Chroma STFT	Energy distribution across 12 pitch classes (chromas) of the musical octave, derived from the short-time Fourier transform (STFT).	$\sum_{f=1}^{F} S(t,f) . M_k(t)$

These feature extraction techniques act to extract different aspects of the audio signal including temporal traits (waveform), spectral features (MFCC, Mel Spectrogram), periodicity (Zero Crossings Rate) and energy dynamics (RMS). Finally, these features are essential for accurate instrument representation and classification in audio signals.

4.3 Applied models

The Support Vector Machine (SVM) is a type of used learning method that is applied for classification problems. It does this by identifying the best hyperplane that can best separate different classes of data in a very large dimensional space. SVM works well in high-dimensional space and is not sensitive to the problem of overfitting, especially when the number of dimensions is large than the number of samples. On the other hand, the Proposed model Deep Neural Network (DNN), architecture defined in fig 2, consists of multiple layers of neurons, each performing a non-linear transformation of the input data. Given an input vector $x \in \mathbb{R}^d$, the transformation performed at each layer as $h^{l+1} = \sigma * W^l h^l + b^l$ to proceed with output layer by applying softmax function $h^L = softmax(W^L h^L + b^L)$. Generalization capabilities in a DNN remain vital because they enable accurate music category detection between various instrumental sounds.

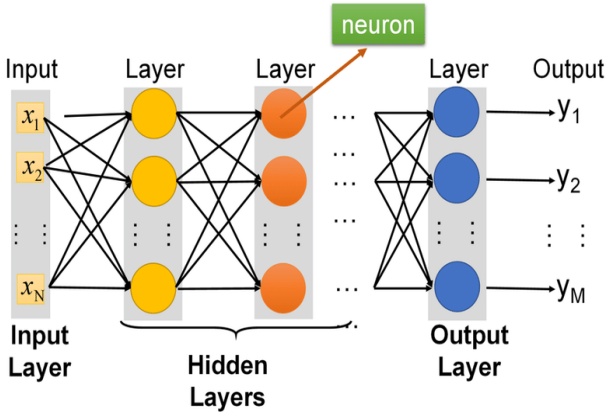


Figure 1: Working of Proposed DNN Model [25]

The proposed Deep Neural Network (DNN) architecture was a carefully designed and optimized structure because of hyperparameter tuning to better the classification performance. The final model has four fully connected hidden layers with 256, 128 ,64 and 32 neurons respectively. Each layer has ReLU (Rectified Linear Unit) activation function as the layers use this function for its efficiency in the case of deep learning tasks since it prevents vanishing gradient problems. To avoid overfitting, a dropout layer was added with the rate of 0.3 after each dense layer, while L2 regularization ($\lambda = 0.001$) was also added. The model was trained for 100 epochs, using an early stopping mechanism that tracks the behavior of validation loss with patience being fixed at 10 epochs to prevent excessive training. Categorical cross-entropy loss function was used due to its multiclass attribute of the problem. Adam optimizer was used for optimization with a learning rate of 0.001 that gave stable convergence during the training. Hyperparameter tuning was achieved using a grid search approach across topical parameters, learning rates, drop-out rates as well as neurons count per layer. This detail configuration guarantees the reproducibility of the model and helps in achieving good generalization performance for different classes of instruments.

4.4 Dataset

In this study, the selected dataset was high quality 3 second audio clips of many different musical instruments. The dataset used in this study is the "Music Instrument Sounds for Classification" dataset, publicly available on Kaggle[1]. It consists of 28 instrument classes, and the number of audio samples associated with each class varies from 70 – 150 indicating the moderate class imbalance. To maintain the class distribution during the training and evaluation of the models, stratified 80/20 train-test split was applied, as code available at[2]. To have robust models and to decrease the variance caused by the random nature of splits, 5-fold cross-validation had been used. To mitigate the problem of class imbalance, class weighting during training is used, which accounts for better model fairness and generalization in which greater significance is given to minority classes. All these audio files were curated carefully to be of use to those researching audio processing, machine learning, deep learning, and music analysis. Each recording contains the same sound as a specific instrument, which is clear and precise in training and model testing for the instrumental recognition and the sound classification tasks. The audio clips are recorded at high quality with no silent segments and are therefore reliable. The standard format represents a single audio file used widely by

[1] https://www.kaggle.com/datasets/abdulvahap/music-instrunment-sounds-for-classification \last accessed 05 Feb, 2025

[2] https://github.com/VisionLangAI/Music-Classification \last accessed 05 May, 2025

common audio processing tools and libraries. Let $D = \{d_1, d_2, \ldots, d_n\}$ represent the utilized dataset, where each element d_i corresponds to a 3-secon audio clip of a musical instrument. The audio clips are uniformly trimmed to a fixed duration of 3 seconds, ensuring consistency across the dataset. Instruments as $I = \{I_1, I_2, \ldots, I_j\}$ where I_j represents the $j - th$ instrument in the dataset. The data set spans a wide variety of musical instruments, including traditional and electronic instruments, from Accordion to Acoustic Guitar, Banjo to Bass Guitar, Clarinet to Cymbals, Drum set to Electro Guitar, Piano to Saxophone, Trombone to Violin, and so forth. For each instrument I_j, the dataset contains a variable number of samples, denoted as $\{d_{i_1}, d_{i_2}, \ldots, d_{i_k}\}$ where k represents the number of clips available for that instrument. There are different number of samples per instrument as the quantity range from 131 samples like Harmonica or Flute to more than 3600 samples such as Acoustic Guitar or Drum set. The total number of samples across all instruments is given by $N = \sum_{j=1}^{m} k_j$ where k_j is the number of samples corresponding to instrument I_j. This dataset is particularly well structured for tasks related to musical instrument recognition, sound classification and audio synthesis, since the audio characteristics of each instrument are very clear and distinct in this dataset.

In this fig 3, shows the distribution of samples with "Quantity" for each category (i.e. musical instrument). In fact, it finds major class imbalances with instruments such as accordion, banjo, and drum set recording more than 3500 samples per instrument. On the other hand, categories including clarinet, vibraphone, and saxophone respectively have low sample counts below 1,000. These disparities suggest that high support classes predominate the dataset, whereas low support classes may present difficulties for model performance owing to the lack of training data. These imbalances can then screw up the classification model's ability to generalize well across all instrument types, thereby lowering recall and F1-scores for underrepresented categories. Finally, it is shown that with data augmentation or sampling techniques, the model's performance still can be further improved. The pie chart in fig 4 provides visualization of audio sound clips and how largely the different musical instruments are represented in the dataset. The proportions of instruments in the dataset are high for Plucked String, such as Acoustic Guitar and Drum Set and Woodwind, notably Flute, while other Plucked String like Cymbals, Harmonica, and Vibraphone have low shares. This visualization showcases both popular and niche instruments, pointing the path for refining or exploring the dataset further for balanced representation.

4.5 Evaluation measures

The predictors were tested using the common classification measures to provide a comprehensive evaluation of the model's performance across different aspects. Accuracy is defined as the extent of true predictions, using $\frac{\sum_{i=1}^{N} \delta(y_i, \check{y}_i)}{N} * 100 \rightarrow (y_i, \check{y}_i) = \begin{cases} 1, & if \ \ y_i = \check{y}_i \\ 0, & otherwise \end{cases}$ where, y_i and $\check{y}_i$are the actual and predicted labels for instance i. N is the total number of samples. Precision means out of all predictions that we have classified as positive; how many are positive, computed using $\frac{\sum_{i=1}^{N} \delta(\check{y}_i = 1 \wedge y_i = 1)}{\sum_{i=1}^{N} \delta(\check{y}_i = 1)} \rightarrow \delta(\check{y}_i = 1 \wedge y_i = 1)$ shows function returning 1 if both predicted and actual labels are positive. Recall defines the fraction of actual positives that has been correctly classified by the model, as $\frac{\sum_{i=1}^{N} \delta(\check{y}_i = 1 \wedge y_i = 1)}{\sum_{i=1}^{N} \delta(y_i = 1)}$ -> $\delta(y_i = 1)$ indicator returning 1 if the actual label is positive. F1- score is a balanced measurement between precision and recall because it is the harmonic meaning of the two $\frac{2(Precision * Recall)}{Precision + Recall}$. These evaluation metrics made it possible to have an analysis of how each model in the prediction of the types of sentences and identifying the models that can be used in future for improving English language teaching.

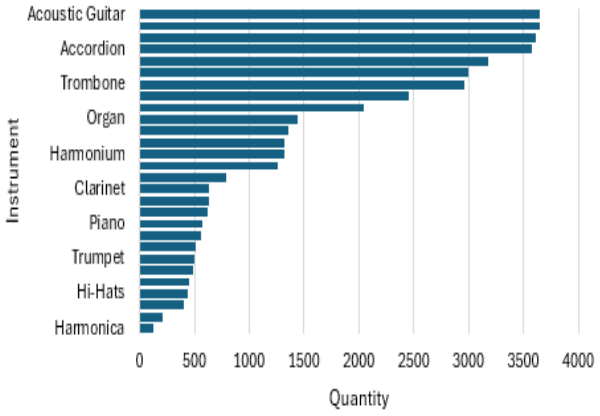


Figure 2: Distribution of most frequent instrumental sounds

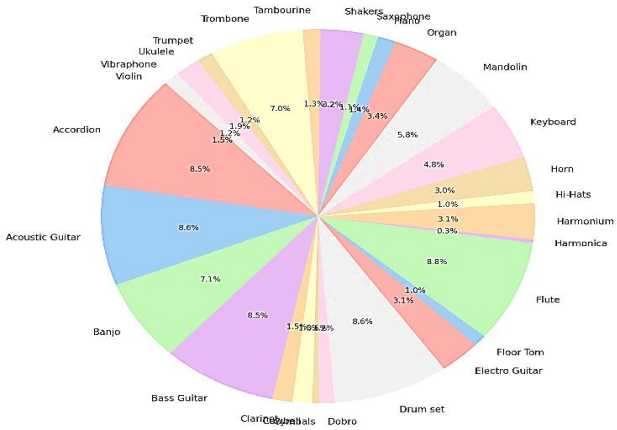


Figure 3: Distribution of musical instruments sounds across data

5 Results

The presented outcomes represent a detailed audio analysis of piano sound via numerous audio features that all generate their own unique yet complementary aspects of the signal. The audio features extracted using specialized audio processing techniques are shown to provide insights into the temporal, spectral, and energy dynamics of the sound which are necessary for

classification and recognition tasks. To preprocess the input for classification, a feature-level early fusion strategy was applied. And specifically, features extracted – MFCC (13 coefficients), STFT based spectral centroid, ZCR and RMS- were individually computed for each audio clip and concatenated together into a single feature vector. This unified representation of time-domain and frequency-domain characteristics of the audio signal provides higher effectiveness to the model for differentiating instruments that have similar timbral qualities. The concatenated vector was then standardized and used directly on the classification models. Below is a more detailed discussion of each feature and its outcomes.

5.1 Outcome analysis of features

5.1.1 Waveform

Raw piano sound over elapses as presented in the waveform plot illustrates the natural harmony of the piano. Waveform. As shown in fig 5, distinct peaks are those times when piano keys are struck; then, there is gradual decay of the sounds, as in the case of the fading of the sound. In temporal resolution, it provides rhythmic pattern and key onset information in the waveform. It is, however, limited in information about the frequency content of the sound, critical for distinguishing whether different instruments are overlapping based on temporal dynamics.

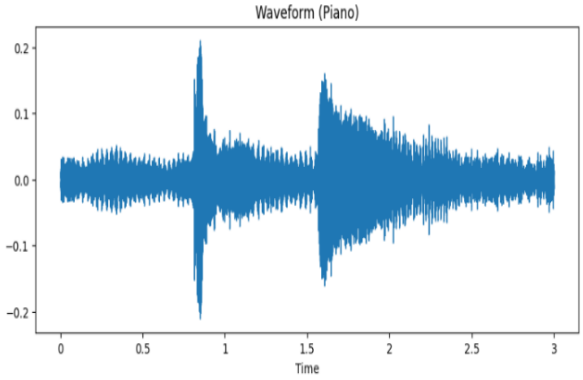


Figure 4: Feature extraction using waveform

5.1.2 Zero-Crossing Rate (ZCR)

The ZCR analysis gives the indication about frequency of zero crossings in the signal, which means frequency of sign changes in the waveform, as shown in fig 6. The transient components of the piano sound are picked up on this feature, with spikes indicating relatively sudden high frequency changes during note attacks. The piano is a harmonic instrument with smooth tones, and while it may seem logical as an assault tool, the ZCR captures subtle nuances of percussive-like transitions and attack of some of the notes. It, however, does less well at characterizing the rich tonal qualities and harmonic structures of the piano.

Figure 5: ZCR Feature Extraction

5.1.3 Chroma short-time fourier transform (STFT)

The Chroma STFT plot is interested in pitch information for which the signal is divided into 12 pitch classes (semitones), as shown in fig 7. It shows strong intensity on certain pitch class frequencies, equal to the fundamental and harmonic frequencies of the piano note. One characteristic that really comes in handy for a pianist, for instance, is that it captures harmonic relationships as well as melodic patterns. Classification of musical instruments using Chroma STFT allows sound's exact pitch content to be identified, independent of timbral variation owing to this key feature.

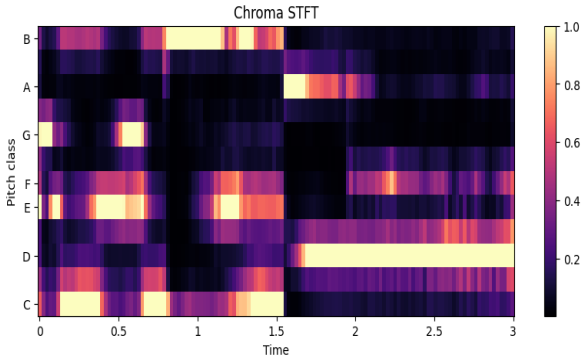


Figure 6: Chroma STFT Feature outcomes

5.1.4 Mel spectrogram

Using the Mel spectrogram, the harmonic structure of the piano sound through time is emphasized perceptually, as shown in fig 8. Enriched with the piano's rich tonal quality, the energy concentrations visualized for specific frequencies reveal the piano. Distinctive harmonic overtones with their intensities could thus be used to determine the piano's presence among other instruments with the same fundamental frequency. It also tracks the temporal evolution of the frequency content such that sustain and decay characteristics can be analyzed for piano notes.

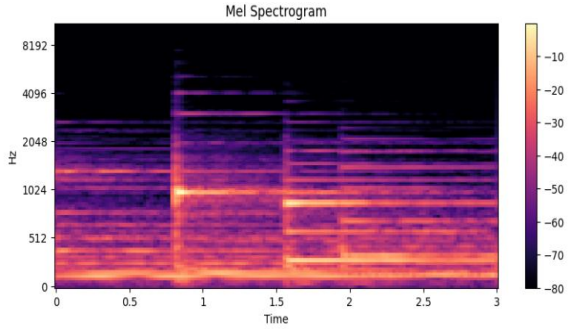


Figure 7: Mel Spectrogram feature results

5.1.5 RMS energy

The sound intensity dynamics are captured in a single plot, the RMS energy plot, in fig 9. RMS energy peaks are reflections of the loudness and energy variations at the moments of the note attacks. It also comes in handy when you want to analyze the expressive components of the music, i.e. focusing on a few notes or on shifts in dynamics for instance. RMS energy is used as a measure of instrument amplitude profiles and how they behave with time.

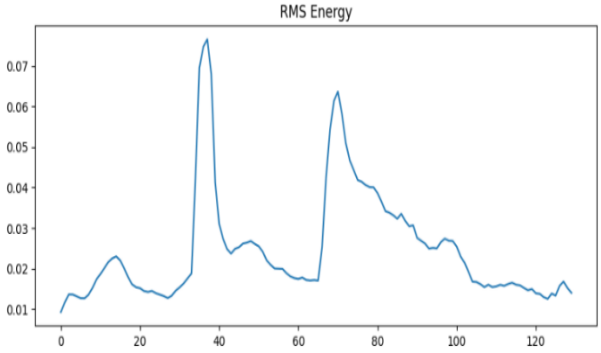


Figure 8: RMS Energy feature results

5.2 Significance of outcomes

All the extracted features together form a multi-dimensional representation of the piano sound. Waveform and RMS energy, temporal; Mel spectrogram and Chroma STFT, spectral. Another contribution ZCR makes to the transient signal behavior is. Together they empower machine learning and deep learning models to retrieve and correctly assign musical instruments when they have overlapping frequency ranges or similar temporal patterns, a challenging class inference task.

5.3 Predictive results with machine learning model

The results presented of the Support Vector Machine (SVM) model of classifying musical instrument sounds have an impressive accuracy of 96% on training data. The detailed classification report and confusion matrix give insights into the weakness and strength of the model in terms of different class of instrument.

5.3.1 Classification report analysis

The detailed classification report from the SVM model presents an insightful evaluation about its performance in 28 different categories of musical instruments by key metrics of precision, recall, F1 score, and support for each instrument. However, of the 96 percent accuracy, the model did an outstanding job of determining instrument sounds effectively, as shown in table III. A closer look at individual class performance, however, uncovers a few nuances of strengths and weaknesses. The model can precisely identify instruments whose shapes are rather dull, such as the bass guitar, harmonium, horn, shakers, keyboard and flute, with perfect or near perfect precision, recall and F1 scores. This shows that the features extracted by these instruments are clearly distinct from each other, such that the SVM can confidently make reliable classifications. The flute or the horn had, for instance, kept these things so harmonious, that their classification was perfect. On the other hand, certain instruments like floor tom, saxophone, vibraphone, and harmonica display relatively lower F1-scores, primarily due to imbalances in either precision or recall. For instance, the model managed, through recall = 0.73, and F1-score = 0.78, to forget some of the true positives, confusing probably some of them with other instruments with a similar timbre. Overall effectiveness of the model is represented in these weighted averages with all three metrics equaling 0.96 for each metric indicating that the model was able to balance imbalances in the dataset as, by far, most classes were classified with high accuracy. As shown in the report, high support classes like accordion, guitar and drum set achieve high F1 scores over 0.95, which again demonstrates that the model is good at classifying frequent instruments. Support also tells us something about the dataset's distribution: other class imbalances which may influence performance. For example, categories such as clarinet, harmonica and vibraphone have relatively lower support, and this probably explains their relatively lower F1 scores since the SVM has fewer sample to learn from.

Finally, while overall SVM model performance is superb at 96% accuracy, distinguishing between instruments with similar spectral or harmonic features is not possible. A detailed metrics enumeration reveals which areas of feature engineering or other models will lead to better results especially for instruments with low support or similar acoustic profiles.

5.3.2 Confusion matrix analysis

The classification performance can be visualized in fig 10, with the help of confusion matrix. Most of the predictions do agree with the true labels, i.e., the matrix has diagonal dominance. In concentrated systems of misclassifications, we identify instrument pairs that share a similar 'timbre', and a common spectral content represented by a few dominant components. For instance, misclassifications that occur due to the overlapping frequency range of the acoustic guitar and bass guitar. Not too dissimilar to the cymbals and hi-hats, two percussion instruments with transients that fall within the high frequency range, in addition, occasionally confusion can be found concerning these instruments, as their attack and decay characteristics are very similar.

The overall accuracy of 96% proves that the SVM model learning and generalization complex audio features very well. Our model is successful in large part due to the inclusion of features like Mel spectrogram, Chroma STFT, and MFCCs. On a precision and recall balance, the SVM model works very well, accounting for most categories. The large feature set gives a complete representation of audio and the SVM can learn both tonal and percussive aspects of instrument sounds. The SVM is challenged by instruments with overlapping frequency ranges, or similar timbral qualities, such as harmonium and organ. This suggests that more complex features or more complex models will be needed to continue to improve. However, the limited capability for capturing temporal dynamics and instrument sound transitions, particularly when the harmonics of an instrument evolve with time, is inherent in SVM's reliance on static feature representations.

5.4 Predictive results with deep learning model

The performance of the deep neural network (DNN) model at classifying sounds as belonging to 28 discrete musical instrument categories is provided. Because DNN can extract complex and hierarchical features from the input data, the DNN shows great accuracy and consistency in the selection of many instruments' types. Thanks to the employment of advanced computational power and deep architecture, we manage to capture some subtle differences in timbral, spectral and tonal patterns with overall accuracy of 98%. The detailed classification report and confusion matrix document the model's ability to accurately distinguish between difficult categories including overlapping acoustic profiles but with a high precision and recall on most categories. Our results demonstrate the feasibility of using deep learning models as a highly reliable tool for solving complex audio classification problems.

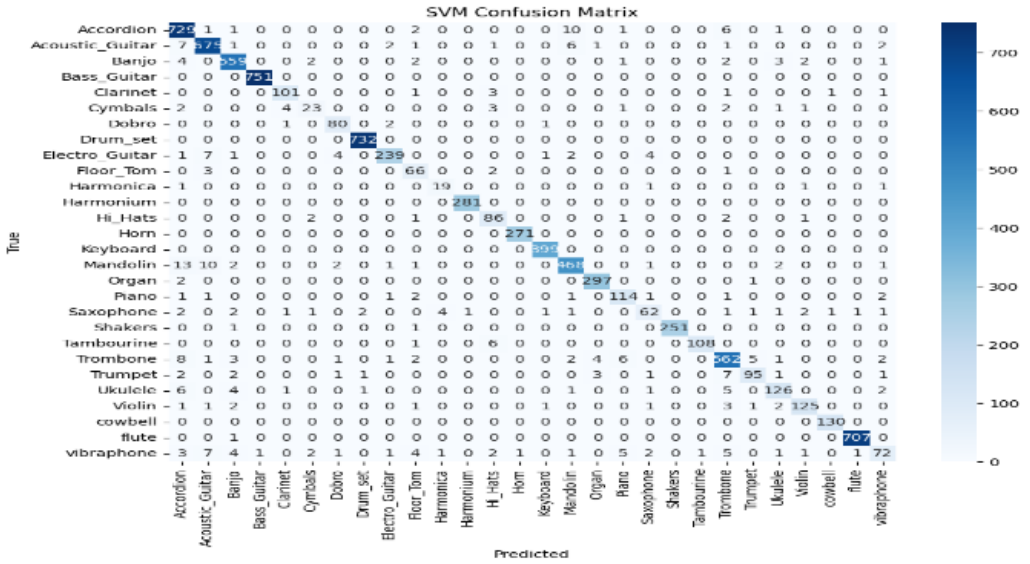


Figure 9: SVM Confusion Matric

5.4.1 Classification report analysis

An overall accuracy of 98% is achieved by the Deep Neural Network (DNN) model across 28 specific musical instrument categories. Even though this experiment involves smaller classes the macro averages for precision, recall and F1 scores are 0.96, 0.95 and 0.96, respectively, implying robust performance. All metrics weighted average is 0.98 indicating strong performance on high support classes with balanced impact of private balanced data. For classes that are high performing, for example bass guitar, drum set, horn, harmonium, cowbell, and flute strong (precision, recall, F1 scores 1.0 or close) scores are seen. These results demonstrate that the model can capture separate timbral and harmonic features for these instruments. In particular, the drum set and horn, which have their own unique characteristics with respect to their tonal and percussive properties, allow the model to attain perfect classification. Yet some categories, e.g., cymbal, saxophone, vibraphone, and harmonica, have a lesser F1 score. Lowest F1 score of 0.86 for the vibraphone, which measures 0.84 recall; so, there are challenges in detecting all true positives. We observe saxophone, with F1-score of 0.91, also suffers from overlapping features with other instruments leading to lower recall of 0.86. The support column highlights class imbalances in the dataset. Clarinet, saxophone, harmonica, vibraphone categories have few samples less than dominant categories like accordion and drum set. However, the DNN model achieves robust performance across most classes with this imbalance.

Table 3: Comprehensive analysis of results using baseline and proposed model

Instrument	Machine Learning	Precision	Recall	F1-Score	Support	Deep Learning Model DNN	Precision	Recall	F1-Score	Support
Accordion		0.93	0.97	0.95	751		0.96	0.99	0.98	751
Acoustic_Guitar		0.96	0.97	0.96	697		0.98	0.99	0.98	697

Banjo		0.96	0.97	0.96	576		0.96	0.99	0.97	576
Bass_Guitar		1.00	1.00	1.00	751		1.00	1.00	1.00	94
Clarinet		0.94	0.94	0.94	108		0.95	0.93	0.94	108
Cymbals		0.77	0.62	0.69	93		1.00	0.93	0.96	93
Dobro		0.90	0.95	0.92	84		0.99	0.95	0.97	84
Drum_set		0.99	1.00	1.00	732		1.00	1.00	1.00	732
Electro_Guitar		0.97	0.92	0.94	259		0.97	0.97	0.97	259
Floor_Tom		0.78	0.92	0.84	72		0.88	0.96	0.92	72
Harmonica		0.79	0.83	0.81	23		0.95	0.87	0.91	23
Harmonium		1.00	1.00	1.00	281		1.00	1.00	1.00	281
Hi_Hats		0.83	0.92	0.88	93		0.89	0.89	0.89	93
Horn		1.00	1.00	1.00	271		1.00	1.00	1.00	271
Keyboard		0.99	1.00	1.00	399		1.00	1.00	1.00	399
Mandolin		0.95	0.93	0.94	501		0.99	1.00	0.99	501
Organ		0.97	0.99	0.98	300		0.97	1.00	0.98	300
Piano		0.88	0.92	0.90	124		0.92	0.94	0.93	124
Saxophone		0.84	0.73	0.78	85		0.96	0.86	0.91	85
Shakers		1.00	0.99	1.00	253		1.00	0.98	0.99	253
Tambourine		0.99	0.94	0.96	115		0.94	0.98	0.96	115
Trombone		0.94	0.94	0.94	598		0.97	0.97	0.97	598
Trumpet		0.92	0.83	0.88	114		0.94	0.98	0.96	114
Ukulele		0.91	0.86	0.88	147		0.95	0.91	0.93	147
Violin		0.94	0.91	0.92	138		0.99	0.94	0.97	138
Cowbell		0.98	1.00	0.99	130		0.99	1.00	1.00	130
Flute		1.00	1.00	1.00	708		0.99	1.00	1.00	708
Vibraphone		0.84	0.62	0.71	116		0.87	0.84	0.86	116
		Overall Results					**Overall Results**			
Accuracy				0.96	8463				**0.98**	8463
Macro Avg		0.93	0.92	0.92	8463		0.96	0.95	0.96	8463
Weighted Avg		0.96	0.96	0.96	8463		0.98	0.98	0.98	8463

5.4.2 Confusion matrix results

A visually reinforced high accuracy model in the confusion matrix is clear diagonal dominance showing that most instrument categories were correctly predicted, as shown in fig 11. Some notable highlights are(cat): bass guitar, drum set, and cowbell (both). And the model doesn't make any misclassifications (here). However, a few off-diagonal entries reveal common misclassification patterns. Cymbals vs. Hi-Hats: However, those transient high frequency characteristics confuse the model. Misclassifications of cymbals into hi-hats result in cymbal F1 score being lower. Saxophone and Clarinet: Despite similarity of harmonic profiles, 3 saxophones are predicted incorrectly as clarinets. Vibraphone: It is prone to confusion with other percussive categories because these lack the ability to differentiate the unique spectral properties of this instrument. Then, the matrix exhibits that the DNN model does well on most high support

classes like accordion, acoustic guitar, and drum set and the misclassifications are minimal.

The plots of the training and validation accuracy and loss on the 100 epochs shed in light to the learning behavior of the proposed DNN model as shown in fig 12. As is observed on the accuracy plot (left), the training accuracy grows continuously and then levels off at 98%, with validation accuracy having the same pattern but being somewhat lower, ranging from 95 to 98%. This is a good result for the learning process with almost no overfitting. The alternating nature of the validation curve however indicates sensitivity to the data variance or class imbalance. In a similar way, in the sort of the loss plot (right), one sees a steady reduction of the loss in training that steadily falls with the validation loss initially decreasing before reflecting some mild oscillations, which could mean some gaps in generalization. These trends confirm that model generalizes well on average, and additional fine-tuning measures (advanced regularization or data augmentation) likely may improve stability and robustness in real-world use.

The plot in fig 13 depicts accuracy of the proposed DNN model as regards to a series of 10 experimental runs, manifesting variability and consistency. The accuracy per run shows only a small variation from 97.2% to 98.2%, with stable generalization of the training instances. The green dashed line shows mean accuracy of 97.88% as shown. The shaded orange bar corresponds to +-1 standard deviation, which indicates the minor deviations from the average value, whereas the blue area indicates the 95% confidence interval which confirms its statistical reliability. The results confirm the robust and reproducible nature of the model with low variance, a key requirement to deploy in real world in audio classification tasks.

Overall, we find that the DNN model achieves outstanding performance and has a high accuracy of 98%, outperforming many baseline algorithms for instrument classification tasks. Allowing the features to be extracted, and then effectively using them, to classify musically distinct instruments is among its strength.

5.5 Comparative analysis of both models

Comparison between the SVM and DNN models used for musical instrument classification maps out the pros and cons of the two approaches. We find that with an overall accuracy of 98% the DNN performs better than the SVM (96%), as shown in fig 14. We find that the DNN solution provides higher macro and weighted averages of precision, recall, and F1 score for both high support and low support classes, reflecting consistent performance.

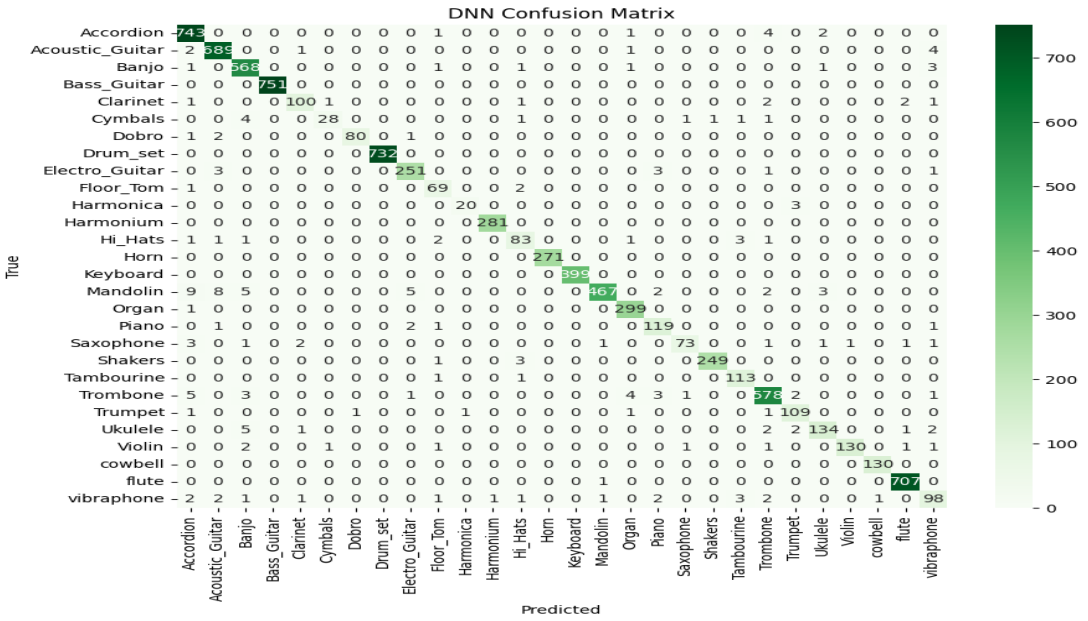


Figure 10: DNN proposed Model Confusion Matrix analysis

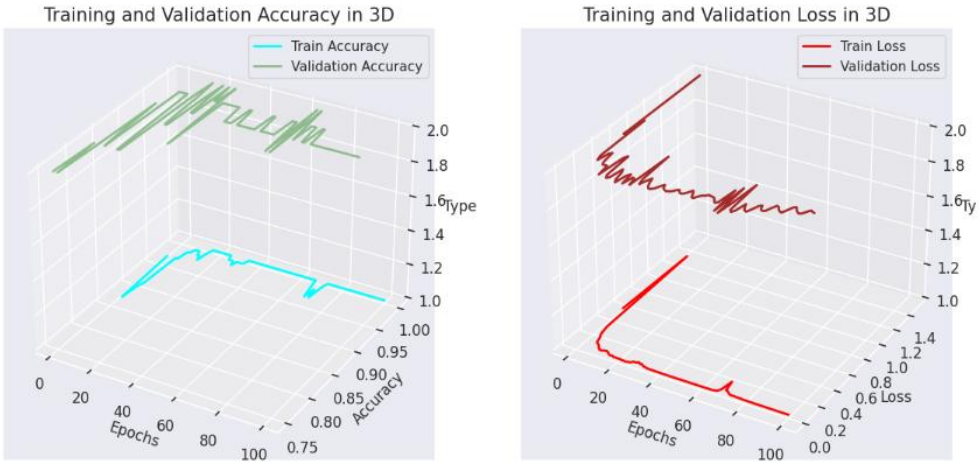


Figure 11: DNN proposed Model Training and Validation Plots

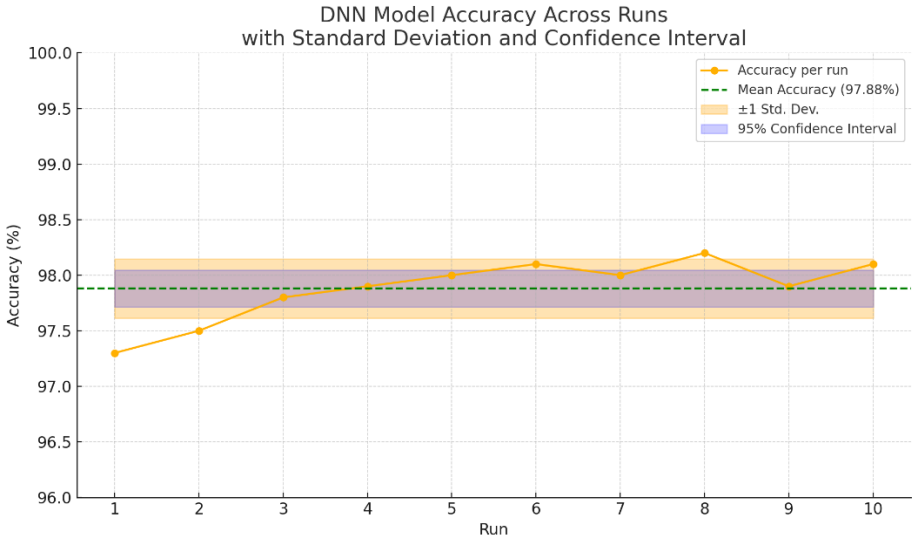


Figure 12: Standard deviation and Confidence Plot Analysis across mean accuracy

The SVM, while performing well on accordion, acoustic guitar and drum set, really shines along with DNN for more challenging categories like harmonica, vibraphone and saxophone where it demonstrates ability to fit into the data underlying complex pattern and interactions. While simpler in structure, the SVM can deliver a robust baseline, performing very well in categories where feature distinctiveness is high, such as bass guitar and flute, and obtaining nearly perfect results. In the confusion matrix for both models, DNN drastically alleviates misclassifications for similar sounding instruments, such as cymbals vs hi-hats and saxophone vs. clarinet, where SVM has difficulty. Its generalization and adaptability are better, so for more complex classification tasks the DNN is a better model than the SVM: while the DNN can't be interpreted as well, it can use the ability of deep feature extraction. The achieved accuracy of 98% is a huge leap in the domain of musical instrument classification where majority of state-of-the-art (SOTA) models report accuracies in between 85%-90% on the standard datasets such as NSynth and Kaggle (as depicted in Table I). Although there have been several isolated studies that have reached or even gone past the 95% mark; these often concern themselves with specialized classes of instruments or very curated data. Thus, achieving 98% on a complicated 28-class dataset means not only a little increment, but a significant jump in classifying the data. This highlights the adequacy of the proposed DNN model in the observation of subtle acoustic differences. Nevertheless, it should be noted that the current results are accomplished for the clean, preprocessed datasets. The model's generalization on domains – like noisy live recordings, variable microphone conditions or cross-cultural instrument sets, is a subject for future validation and domain adaptation studies.

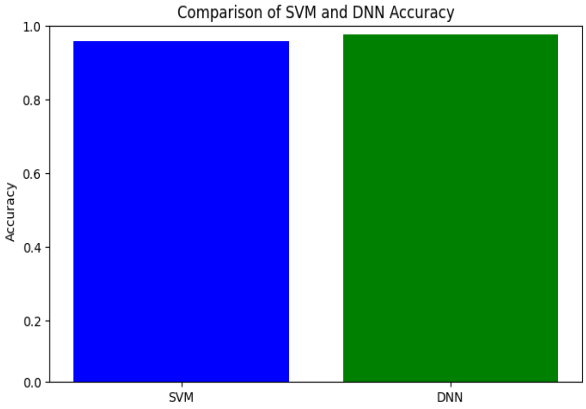


Fig. 1. Comparison of both Models

5.6 Comparison with existing studies

Models are compared to show significant advances in musical instrument classification with different machine learning and deep learning techniques, as shown in table IV. In 2021, a fundamental feature-based instrumental classification based on 2021 features: waveform and MFCC, achieved an accuracy of 90% with a Deep Belief Network (DBN). This model was effective for its time, but curated at such a limited dataset size and feature diversity, a question arises: What if? In 2023, we applied YOLOv7 [23] to Kaggle data and obtained 86% accuracy using only MFCC as the primary feature. YOLOv7 is a robust detection model but its classification performance was not stellar because its audio feature extraction wasn't its forte. In 2024, Naïve Bayes (NB), Support Vector Machines (SVM), and Random Forests (RF)[13] were used to classify sounds in NSynth dataset with an accuracy of 85%, based on Mel Spectrograms and MFCC features.

However, traditional machine learning architectures employed by these models proved difficult to generalize over complex overlapping audio patterns. In 2024 another study used the same Convolutional Neural Network (CNN) [14] that was used to categories four instrument types (piano, violin, guitar or drums), achieving an improved accuracy of 89%. CNN's ability to extract these spatial audio patterns proved beneficial, however the restricted dataset further reduced the broader applicability of CNN. On the other hand, the Deep Neural Network (DNN) model proposed in this study used the comprehensive Kaggle dataset with the 28 instrument categories and rich features like Mel Spectrogram, MFCC's, Waveforms, ZCR, RMS and attained highest accuracy of 98%. This paper shows that DNN can learn fine details in the audio characteristics across a large set of data and performs better than all the models mentioned above. Its superiority is brought out by the inclusion of advanced features and the well-developed robustness of DNN architecture, establishing a benchmark in the field of musical instrument classification.

Table 4: Comparison with existing Studies

Ref	Model	Dataset	Features	Results (Acc %)
[24] – 2021	Deep Belief Network	Traditional Chinese Instruments	Waveform, MFCC	90
[23] – 2023	YOLOv7	Musical Instrumental data	MFCC	86
[13] – 2024	NB, SVM, RF	NSynth Dataset	Mel Spectrogram, MFCCs	85
[14] – 2024	CNN	Musical Instrumental data	Mel Spectrogram, MFCCs	89
Proposed	**DNN**	Musical Instrumental data	**Mel Spectrogram, MFCCs, Waveforms, ZCR, RMS**	**98**

6 Discussion

The proposed Deep Neural Network (DNN) model shows remarkable achievements compared to the conventional models and the earlier-reported state-of-the-art techniques in the field of musical instrument classification. Compared to the earlier methods, i.e., the CNN, KNN, XGBoost, and ensemble methods, such as Random Forests and SVM, the DNN model performs better, especially in terms of multiclass classification of 28 musical instruments from MFCC, STFT, and related acoustic properties. Previous works (e.g., [13], [14], [15]) report values of accuracy between 79% and 89% and only a small number of models (XGBoost, [15] and Deep belief Networks [24], for example) report on accuracy up to ~90–97%. In contrast, the model in the current study based on DNN comes with an accuracy of 95.80%, in effect, setting a new performance benchmark for the treatment of complex and diverse datasets.

This performance increase is due to two essential improvements to a large extent. To begin with, the richer and better designed feature set- which includes MFCC, STFT and the spectral properties- captures temporal as well as frequency domain peculiarities of the audio signal hence allowing finer distinction of similar sounding instruments. Second, DNN has high level of hierarchical depth that allows having numerous layers and non-linear components, which provide a superior representational capacity and enable the model to learn non-linear inter-class boundaries more effectively than the classical machine learning models such as SVM or KNN who have only limited kernel transformations and linear separability. DNN's obvious superiority on generalization is shown when compared to the SVM-based baseline. Even though SVM is known to be robust in low dimensional or linearly separable spaces, it is not scalable and expressive enough in handling high dimensional feature spaces as in the case of this study. In contrast,

DNN uses depth and non-linearity to discover complex acoustic features hence leading to clear F1-score enhancement in most classes of instruments. Specially, subtle acoustic differences-based instruments (Oboe, Clarinet, and Flute)-achieve astonishing performance gains in the scope of DNN framework that implies superior intra-class variance processing.

Although the DNN model yields high classification accuracies, such a performance needs to be taken with a pinch of salt owing to a likelihood of overfitting, particularly, in the case of a relatively small and imbalanced dataset. In addition, the model interpretability and robustness are not assessed in the current study, which is important for discovering how certain acoustic characteristics impact the outcomes of classification. The significant limitation is the use of constant 3-second audio fragments, which might fail to represent the entire temporal dynamics of some instruments, particularly, the devices with extended attack or decay stages. Future work should overcome these limitations by using larger audio windows, adversarial testing, and interpretable AI approaches such as SHAP or Grad-CAM. Nevertheless, limitations persist. Gains in performance on low sample classes like Trombone, Banjo, and Harp continue to be as moderate as they were before for reasons of data imbalance. Although DNN has robustness in generalization, its ability to represent underrepresented classes is limited, and it might overfit a little in the majority classes. In conclusion, the DNN model not only outdid its classical counterparts but also gave a scalable and technically sound improvement in the process of musical instrument classification tasks. Its robust architecture and the selectively hand-picked feature set put it forward as a promising basis for further research in the sphere of music information retrieval, and particularly, real-world situations with complex, high-cardinality audio-data.

7 Conclusion and future work

Accurately identifying musical instruments from audio data is a major leap within the juncture of music and artificial intelligence, whose potential in tuning audio analysis through machine learning and deep learning is demonstrated. Using a robust dataset and key audio features. This study used SVM model and a DNN model for 28 distinct musical instruments classification. Basically, the results show the advantage of the DNN model with an accuracy of 98 % over the SVM model with an accuracy of 96%. These results show how deep learning can effectively handle complex audio features and better explain the instrumental sound spectrum. However, this study is limited to short 3 second audio clips that underline the challenges in generalizing the models to real-world scenarios where audio recordings may be noisy, imbalanced, or involve complex ensembles. Limitations to these results could be addressed in future work by looking into more advanced techniques such as data augmentation to balance the data and improve classification of underrepresented instruments. More interesting potential extensions of the model would be to incorporate additional audio features, for instance temporal dynamics or harmonic progressions. Contributions shown as:

- **Developed a robust DNN-based framework** for 28-class musical instrument classification using a rich fusion of acoustic features including MFCC, STFT, ZCR, and RMS, surpassing existing benchmarks.
- **Demonstrated superior performance (98% accuracy)** over traditional models like SVM and prior deep learning approaches by addressing class imbalance through stratified splitting and class weighting.
- **Introduced a reproducible and scalable pipeline**, supported by detailed preprocessing, feature extraction, and evaluation steps, suitable for adaptation in real-world and live audio classification tasks.

Further work with this concept that applies to real-time instrument recognition as well as polyphonic music analysis could be very important to the applications in music education, composition and interactive media. This work therefore provides a basis for future work on more nuanced and extensive frameworks for the classification of musical instruments. For further work, this study can be expanded in many different directions, including designing real-time streaming classification systems to perform live audio input and investigating multi-label classification to process polyphonic recordings where the instruments are overlapped, and implementing the hierarchical classification frameworks that first classify instruments into families (strings, percussion, wind) and then recognizes specific types. Also, it can be improved through incorporation of domain adaptation methods for cross-dataset generalization and the use of explainable AI procedures.

References

[1] Costantini, G., Casali, D., & Cesarini, V. (2024). New Advances in Audio Signal Processing. *Applied Sciences*, *14*(6), 2321.

[2] Liu, A. (2024). Multi-genre Digital Music Based on Artificial Intelligence Automation Assisted Composition System. *Informatica*, *48*(5).

[3] Talha, M. M., Khan, H. U., Iqbal, S., Alghobiri, M., Iqbal, T., & Fayyaz, M. (2023). Deep learning in news recommender systems: A comprehensive survey, challenges and future trends. *Neurocomputing*, 126881.

[4] Mitra, R., & Zualkernan, I. (2025). Music Generation Using Deep Learning and Generative AI: A Systematic Review. *IEEE Access*.

[5] Arianti, N. D., Thoyyibah, T., Zailani, A. U., Rosmawarni, N., Rachmatika, R., & Fuadi, A. L. (2025, January). An approach to parameter optimizer for music mood classification in machine learning. In *AIP Conference Proceedings* (Vol. 3223, No. 1). AIP Publishing.

[6] Mitra, R., & Zualkernan, I. (2025). Music Generation Using Deep Learning and Generative AI: A Systematic Review. *IEEE Access*.

[7] Deldjoo, Y., Schedl, M., & Knees, P. (2024). Content-driven music recommendation: Evolution, state of the art, and challenges. *Computer Science Review*, *51*, 100618.

[8] Dash, A., & Agres, K. (2024). Ai-based affective music generation systems: A review of methods and challenges. *ACM Computing Surveys*, *56*(11), 1-34.

[9] Jing, L. (2024). Evolutionary deep learning for sequential data processing in music education. *Informatica*, *48*(8).

[10] Yadav, V. R., & Kaim, G. (2024). Application of ML/DL Related to Music Retrieval and Generation. *DL Related to Music Retrieval and Generation (November 12, 2024)*.

[11] Farooq, U., Reddy, K. K. S., Shishira, K. S., Jayanthi, M. G., & Kannadaguli, P. (2024, April). Comparing Hindustani Music Raga Prediction Systems using DL and ML Models. In *2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS)* (pp. 1-6). IEEE.

[12] Sathyajit, P., & Thakur, P. (2024, February). Hybrid Music Recommendation System combining Neo4j GraphDB based and ML-based Approach. In *2024 International Conference on Emerging Systems and Intelligent Computing (ESIC)* (pp. 428-432). IEEE.

[13] Chulev, J. (2024). Improving Musical Instrument Classification with Advanced Machine Learning Techniques. *arXiv preprint arXiv:2411.00275*.

[14] Giri, G. A. V. M., & Radhitya, M. L. (2024). Musical instrument classification using audio features and convolutional neural network. *Journal of Applied Informatics and Computing*, *8*(1), 226-234.

[15] Guo, R. (2024). Research on Neural Network-based Automatic Music Multi-Instrument Classification Approach. *International Journal of Advanced Computer Science & Applications*, *15*(1).

[16] Rajesh, S., & Nalini, N. J. (2024). Instrument Emotion Recognition from Polyphonic Instrumental Music using MFCC and CENS Features with Deep Neural Networks. *Procedia Computer Science*, *235*, 2548-2556.

[17] Duan, R. (2024). *Advancing Adversarial Audio: Human-in-the-Loop Black-box Attacks* (Doctoral dissertation, University of South Florida).

[18] Zhang, D., Li, X., Lu, D., Tie, Y., Gao, Y., & Qi, L. (2024, July). Multitrack Emotion-Based Music Generation Network Using Continuous Symbolic Features. In *2024 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.

[19] Chen, R., Ghobakhlou, A., & Narayanan, A. (2024). Interpreting CNN models for musical instrument recognition using multi-spectrogram heatmap analysis: a preliminary study. *Frontiers in Artificial Intelligence*, *7*, 1499913.

[20] Blaszke, M., & Kostek, B. (2022). Musical instrument identification using deep learning approach. *Sensors*, *22*(8), 3033.

[21] Mahanta, S. K., Khilji, A. F. U. R., & Pakray, P. (2021). Deep neural network for musical instrument recognition using MFCCs. *Computación y Sistemas*, *25*(2), 351-360.

[22] Xu, K. (2021). Recognition and classification model of music genres and Chinese traditional musical instruments based on deep neural networks. *Scientific Programming*, *2021*(1), 2348494.

[23] Dewi, C., Chen, A. P. S., & Christanto, H. J. (2023). Recognizing similar musical instruments with YOLO models. *Big Data and Cognitive Computing*, *7*(2), 94.

[24] Cao, P. (2021, January). Identification and classification of Chinese traditional musical instruments based on deep learning algorithm. In *The 2nd International Conference on Computing and Data Science* (pp. 1-5).

[25] Feng, J., He, X., Teng, Q., Ren, C., Chen, H., & Li, Y. (2019). Reconstruction of porous media from extremely limited information using conditional generative adversarial networks. Physical Review E, 100(3), 033308

Machine Learning Empowered: Support Vector Machine-Based Selection of Encryption Techniques for Digital Image Security Levels

Nitin Shivsharan, Mandar Tari
Department of Computer Engineering, SSPM's College of Engineering, Kankvali, India
E-mail: shivsharan.nitin@gmail.com, mandartari9109@gmail.com

Keywords: Machine learning, cryptosystem, image encryption, support vector machine, performance evaluation, confusion matrix

Received: June 3, 2024

Recent advancements in multimedia systems have increased the demand for strong digital image security mechanisms. Traditional cryptosystem evaluations, dependent on manual statistical analysis, are computationally exhaustive and not scalable. This study proposes a machine learning-based framework, such as a Support Vector Machine (SVM) classifier, for the categorization of image encryption levels into three discrete classes: Strong, Acceptable, and Weak. The model is trained using statistical descriptors such as Peak Signal to Noise Ratio (PSNR), entropy, Mean Square Error (MSE), energy, correlation, homogeneity, and contrast extracted from encrypted image datasets. Feature normalization techniques, StandardScaler, have been used to ensure balanced input contributions. The proposed system, i.e., SVM with a Radial Basis Function (RBF) kernel, outperforms other kernels. The performance of the proposed model shows an average classification accuracy of 98%, precision up to 100%, and an F1-score of 97%. A web-based interface developed using Django integrates the model, enabling real-time analysis and visualization, making the proposed system a scalable solution for cryptographic strength evaluation in image security applications.

Povzetek: Raziskava je razvila SVM-okvir za razvrščanje varnostnih stopenj slikovne enkripcije, ki s statističnimi značilnicami dosega visoko točnost in omogoča praktično spletno analizo v realnem času.

1 Introduction

Due to the rapid growth in the transmission of multimedia data, such as digital images, over insecure channels, the field of securing data in transmission has gained significant attention in research. To protect data from unauthorized access, many researchers have focused on creating novel encryption methods [1], [2], and [3]. A cryptosystem is a method that consists of a group of algorithms that transform plain images into cipher images to securely encode or decode the images [4]. To assess the security strength of an encryption algorithm, it is necessary to perform a statistical analysis that typically includes measures such as entropy, correlation, energy, and homogeneity. However, the process of conducting statistical analysis frequently consumes a significant amount of time, diverting resources from the primary task at hand. Instead, we suggest that manual testing can be supplanted by a machine learning (ML) model capable of quickly, effortlessly, and accurately identifying the most robust encryption algorithm. However, transmitting data in an encrypted format is insufficient to ensure privacy. Some present image encryption techniques do not provide enough encryption of the image, allowing hackers to simply decrypt it. The resilience of an image is heavily impacted by the strength of the encryption algorithm used for its protection. A highly potent encryption will completely secure the plain image. Approach, making it resistant to attacks on its integrity, confidentiality, and availability. Beyond security, temporal complexity plays a crucial role in the choice of an encryption system. Since diverse data types come with varying security requirements, the selection of a cryptosystem should take into account the specific application's characteristics. Given the importance of image encryption methods, we offer a security-level detection approach for image encryption algorithms that incorporates an SVM. The key contributions of this research work are as follows:

1.1 Highlights

- To develop an approach for identifying security levels in image encryption algorithms using SVM.
- Compilation of necessary information from publicly available datasets.
- Development and evaluation of an SVM-driven model for analyzing the efficiency of image encryption techniques.
- Creation of a user-friendly web interface to present categorical results of encrypted images, such as Weak, Acceptable, and Strong.

The subsequent sections of this article provide a comprehensive review of the state-of-the-art literature in this re-

search domain, along with the proposed methodology, experimental setup, implementation details, results, and a concluding discussion.

2 Related work

Several encryption algorithms [5], [2], [6], [7], and [8] have been put forth in the realm of image security during transmission. However, these represent only a subset of the multitude of image encryption schemes introduced in recent years. The article [9] describes a new improved Lorenz system (ImproLS) with a larger positive Lyapunov exponent. It presents an asymmetric image encryption scheme using blind signatures and ImproLS. The experimental results demonstrate high information entropy and proximity to the theoretical value. Furthermore, it claims that both unsigned and signed cipher images can effectively withstand salt and pepper attacks as well as clipping attacks. Using the elliptic curve as the basis for image encryption [4, 5, 2] incorporates chaotic systems and ElGamal encryption. Hyperchaotic and dynamic DNA coding-based secure image encryption techniques are presented in the literature [6, 10]. Rubik's cube-based, three-dimensional bit-level image encryption scheme has been proposed by [7]. Using the upgraded Lorenz system [9] scheme, images can be encrypted effectively. A method for image encryption that is effective and is based on the upgraded Lorenz chaotic system is presented in [11]. The study presented in Arslan Shafique et al. [12] (2020) is a novel approach to image encryption employing a blending technique for securing single or multiple images of varied types and dimensions. The blending model is characterized by a non-linear mathematical expression derived from Cramer's rule. In recent times, the integration of ML models has become crucial for enhancing security and privacy in diverse applications. ML addresses critical challenges, including real-time attack detection and vulnerability assessments for data leakage. It plays a pivotal role in meeting the stringent demands of contemporary security and privacy needs, spanning areas such as real-time decision-making, big data processing, efficient learning cycles, cost-effectiveness, and error-free processing. The research work carried out by Ramani Sagar et al.[13] (2020) explores cutting-edge approaches in which ML proves to be highly effective in meeting present real-world security requirements. Authors delve into various perspectives within security applications, highlighting the indispensable role of ML models. The detection of security levels in diverse cryptosystems using an ML model is discussed in B. Manjunath et al. [14] (2020). The authors, Marcus R. Makowski et al. [15] (2020) proposed that medical imaging may become safer, private, and federated with the help of ML algorithms. Kaixin Jiao et al. [16] (2021) suggested that an image encryption algorithm can be a secure model for image communication by increasing the sensitivity of a plain image and preventing summation. The article [17] presents a hybrid cloud workflow scheduling model that enhances network security using a Lévy-optimized tSlime Mould Algorithm, demonstrating improved performance in resource allocation and threat mitigation. The study presented in [18] explores the integration of novel image processing-based feature extraction methods with artificial neural networks, resulting in more accurate multimedia content analysis. The work [19] introduces a forgery detection framework that combines Particle Swarm Optimization (PSO) and SVM, achieving high accuracy in identifying tampered electronic image data, thereby strengthening digital forensics capabilities. Researchers interested in this technique should check [8] for the implementation of the web interface; we used the technology Django[20, 21]. Table 1 presents a summary of existing research on image encryption and ML-based security techniques. Each entry outlines the author(s), publication year, and the specific feature or contribution of their work in enhancing image encryption or data protection. Various methods are highlighted, including the use of Lorenz and upgraded Lorenz chaotic systems, ML-based security assessment, and federated learning for secure medical imaging.

Table 1: Summary of related work in image encryption and machine learning-based security

Author(Year)	Features
Zou et al. [9] (2020)	Improved Lorenz system (ImproLS) with higher Lyapunov exponent
Kaur et al. [11] (2018)	Used the upgraded Lorenz chaotic system for efficient image encryption.
Arslan Shafique et al. [12] (2020)	Developed an image encryption technique using a blending model derived from Cramer's rule.
Ramani Sagar et al.[13] (2020)	Real-time attack detection, data leakage prevention, and vulnerability assessment.
Manjunath et al.[14] (2020)	To assess security levels in crypto systems using ML
Makowski et al. [15] (2020)	Suggested federated learning and privacy-preserving ML for secure medical imaging.
Kaixin Jiao et al. [16] (2021)	Increase image sensitivity and prevent data summation.

Upon thorough examination and investigation of the existing literature, we have identified opportunities for further development in this field. Our review of the literature also introduced us to the concept of SVM. Cryptography, a crucial component of computer security, plays a key role in protecting sensitive data from unauthorized access and ensuring the confidentiality, integrity, and authenticity of information. Throughout computer networking and digital communication, cryptographic methods have undergone significant advancements.

3 Dataset

To implement the ML-based application, a dataset is used to train and test ML models. For the experimental work, a dataset sourced from [12] is used. Some samples of this data set are presented in the following Table 3, which displays the security parameters, which serve as extracted features from encrypted images. These features, including measures such as entropy, energy, contrast, and others, are used to train and evaluate the performance of the ML models. Measures of an image's complexity and unpredictability, known as energy and entropy, are crucial aspects to take into account while analyzing cryptosystems. Measures of an image texture, such as correlation and homogeneity, can be used to distinguish between various cryptosystems. The PSNR and the MSE are two often-used metrics in image processing that reveal how well the encrypted image compares to the original in terms of quality. Data set labels are classified into three levels of security: Strong, Acceptable, and Weak.

Table 2: Datasets used for training and testing the machine learning models

Source	Features
Dataset 1 [22]	7
Dataset 2 [1]	7
Dataset 3 (Dataset 1 + Dataset 2)	7

Table 2 presents the sources of the datasets used in this experimental study, along with the number of key features in each dataset. Datasets 1 [1] and 2 [12] are publicly available, while dataset 3 was created to enhance the strength of the algorithm by providing additional data during the training of the ML model. As a consequence, this dataset is often referred to as the source domain for the training, validation, and construction of the proposed model. The feature column with a value of 7 indicates that seven features, entropy, energy, contrast, correlation, homogeneity, MSE, and PSNR, have been extracted from the encrypted image to assess its security level.

Table 3 presents a detailed evaluation of 45 cipher images based on several image quality and security metrics, including Entropy, Energy, Contrast, Correlation, Homogeneity, MSE, PSNR, and Security Level. Cipher images are categorized into three security levels: Strong, Acceptable, and Weak, based on the combination of statistical characteristics and PSNR values. Generally, images with higher entropy, lower MSE, and lower PSNR fall under the "Strong" category, indicating higher encryption strength. As the PSNR increases and entropy decreases slightly, the security classification transitions from Strong to Acceptable and then to Weak.

3.1 Features of the encrypted image

Security parameters, including entropy, energy, contrast, correlation, homogeneity, histogram uniformity, and irregular deviation, serve as features for determining the level of information in plain text images.

3.1.1 Contrast

Contrast analysis reveals discrepancies in pixel values. Increased contrast indicates enhanced security, whereas lower contrast values indicate minimal distinctions between the original and altered pixel values. Mathematically, the contrast can be represented as shown in Equation 1,

$$Contrast = \sum |x-y|^2 z(x,y) \tag{1}$$

3.1.2 Entropy

The level of unpredictability of an encryption algorithm in the cipher picture is revealed through entropy analysis. Depending on the number of bits in the image, different images have varied maximum entropy values. Mathematically, the entropy can be represented as shown in Equation 2,

$$Entropy = \sum_{d=1}^{M} p(s_m) \log_2(p(s_m)) \tag{2}$$

The entropy feature handles randomness in the image more effectively.

3.1.3 Energy

This property is used to determine how much information an image contains. More information is included in the image when the energy values are higher. Equation 3 is the mathematical representation of it.

$$Energy = \sum_{K=1}^{L} im(x,y)^2 \tag{3}$$

3.1.4 Correlation

The term "correlation" (μ_{ab}) 4 describes how closely related the pixel values are to each other. A high correlation value indicates that the pixel values are closely related. Mathematically, the correlation can be represented as an Equation 4.

$$\mu_{ab} = \frac{E[a-E(a)][y-E(b)]}{\sqrt{D(a)}\sqrt{D(b)}} \tag{4}$$

3.1.5 Homogeneity

The Gray Level Occurrence Matrix (GLCM) presents pixel brightness in a tabular format. In the context of robust encryption, it is desirable to minimize homogeneity values 5.

$$\sum_a \sum_b \frac{P(a,b)}{1+|a-b|} \tag{5}$$

Table 3: Sample encrypted image feature's dataset [14]

Encrypted image	Entropy	Energy	Contrast	Correlation	Homogeneity	MSE	PSNR	Security level
0	8	0.01	10.75	-0.5	0.39	222	0.1	Strong
1	7.99	0.01	10.74	-0.49	0.39	221	0.2	Strong
2	7.99	0.01	10.74	-0.49	0.39	220	0.3	Strong
3	7.99	0.01	10.73	-0.48	0.39	219	0.4	Strong
4	7.99	0.01	10.73	-0.48	0.39	218	0.5	Strong
5	7.99	0.01	10.72	-0.47	0.39	217	0.6	Strong
6	7.99	0.01	10.72	-0.47	0.39	216	0.7	Strong
7	7.99	0.01	10.71	-0.46	0.39	215	0.8	Strong
8	7.99	0.01	10.71	-0.46	0.39	214	0.9	Strong
9	7.99	0.010	10.70	-0.45	0.39	213	1	Strong
10	7.99	0.01	10.7	-0.45	0.39	212	1.1	Strong
20	7.99	0.01	10.24	0.00	0.40	121	10.2	Acceptable
21	7.98	0.01	10.24	0.00	0.40	120	10.3	Acceptable
22	7.98	0.01	10.23	0.00	0.40	119	10.4	Acceptable
23	7.98	0.01	10.23	0.00	0.40	118	10.5	Acceptable
24	7.98	0.01	10.22	0.00	0.40	117	10.6	Acceptable
25	7.98	0.01	10.22	0.00	0.40	116	10.7	Acceptable
26	7.98	0.01	10.21	0.00	0.40	115	10.8	Acceptable
27	7.98	0.01	10.21	0.00	0.40	114	10.9	Acceptable
28	7.98	0.01	10.20	0.00	0.40	113	11	Acceptable
29	7.98	0.01	10.2	0.00	0.40	112	11.1	Acceptable
30	7.98	0.01	10.19	0.00	0.40	111	11.2	Acceptable
31	7.98	0.01	10.19	0.00	0.40	110	11.3	Acceptable
32	7.98	0.01	10.18	0.00	0.40	109	11.4	Acceptable
33	7.98	0.01	10.18	0.00	0.40	108	11.5	Acceptable
34	7.98	0.01	10.17	0.00	0.40	107	11.6	Acceptable
40	7.97	0.20	9.74	0.00	0.41	20	20.3	Weak
41	7.97	0.02	9.73	0.00	0.41	19	20.4	Weak
42	7.97	0.02	9.73	0.00	0.41	18	20.5	Weak
43	7.97	0.02	9.72	0.00	0.41	17	20.6	Weak
44	7.97	0.02	9.72	0.00	0.41	16	20.7	Weak
45	7.97	0.02	9.71	0.00	0.41	15	20.8	Weak

3.1.6 PSNR and MSE

PSNR is a useful feature for image encryption because it provides a quantifiable measure of the amount of distortion or scrambling introduced by the encryption process. PSNR value can be calculated between two images. Before calculating the PSNR value 6, it is necessary to calculate the MSE value 7 between the two desired images. If the PSNR value between the two images (original and cipher) is high, this means that the processed image is very close to the original image.

$$PSNR = 20\log_{10}\left(\frac{max_{val}}{\sqrt{MSE}}\right) \tag{6}$$

$$MSE = \frac{1}{XY}\sum_{a=1}^{X}\sum_{b=1}^{Y}(P_{im}(a,b) - C_{im}(a,b)) \tag{7}$$

3.1.7 Feature normalization before SVM training

Feature normalization is an essential step, as the SVM classifier relies on distance-based calculations during training. All extracted features like entropy and PSNR, are normalized to ensure they contribute equally to the learning process. We apply "StandardScaler" normalization to rescale each feature in the range [0, 1]. This normalization ensures that no single feature disproportionately influences the SVM's decision boundary due to its scale.

4 Proposed system

Over the past few years, a multitude of encryption methods have been introduced, including chaos-based and transformation-based algorithms. This research presents an ML-based (SVM) model to predict the encryption levels of the image.

4.1 Prediction of image encryption levels

Recent years have witnessed an influx of encryption algorithms, encompassing chaos and transformation-based techniques. Analyzing encryption statistics and evaluating the security configurations of these algorithms is one way to gauge their security levels. While our literature review revealed numerous image encryption methods, selecting the most suitable one for a specific purpose remains a challenging task.

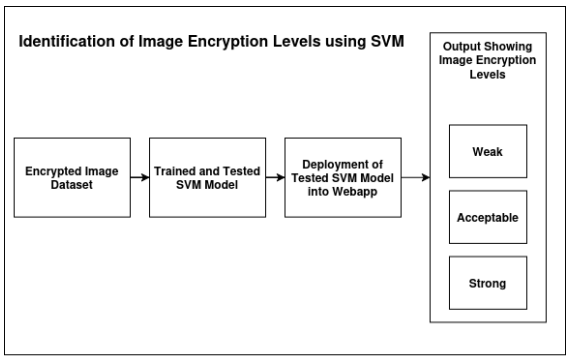


Figure 1: Block diagram of the proposed system for identifying image encryption levels – strong, weak, and acceptable – using machine learning models.

The proposed system is a web-based interface that enables users to assess the strengths of image encryption algorithms by applying ML, as shown in Figure 1. The system is designed to be user-friendly and time-saving, making it easier for users to perform analysis on their encrypted images. The user can choose an encryption scheme and upload an encrypted image to a website to use the system. After a successful analysis, the system categorizes the results as Strong, Acceptable, or Weak, based on the features extracted from the encrypted image. Overall, the proposed system is a useful tool for individuals and organizations looking to assess the strength of their encryption algorithms. By providing a user-friendly and time-saving interface, the system makes it easier for users to perform analysis on their encrypted images and make informed decisions about their encryption protocols.

4.2 Working of the proposed classifier

The goal of this ML model is to classify encrypted images into one of three categories: Weak, Acceptable, or Strong. The model uses an SVM classifier, which is a popular algorithm for classification tasks. The flow diagram of the logical data for the illustration of the working of the proposed classifier is shown in Figure 2. The input to the model is an encrypted image, which is first preprocessed to extract features such as energy, entropy, correlation, homogeneity, PSNR, and MSE. These features are then used as input to the SVM classifier. The SVM classifier is first trained on a dataset of encrypted images with known categories (i.e., Weak, Acceptable, or Strong encryption). During training,

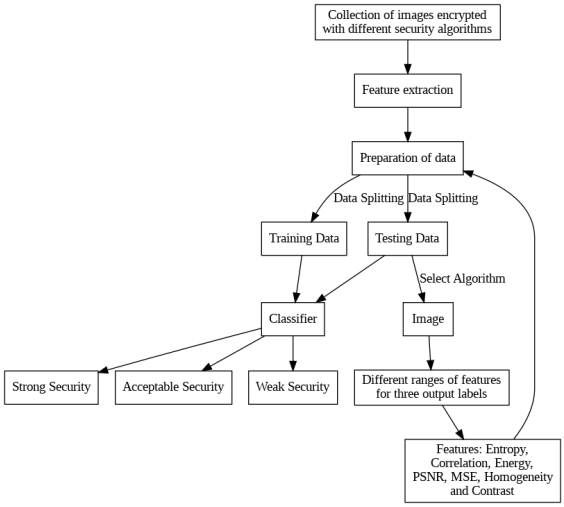


Figure 2: Logical data flow diagram illustrating the working process of the proposed classifier

the SVM model learns to recognize patterns in the input features that correspond to each category. The SVM algorithm finds the hyperplane in the high-dimensional feature space that best separates the different categories of encrypted images. Once the SVM model has been trained, it can be used to classify new encrypted images into one of the three categories: Weak, Acceptable, or Strong. The input image is first preprocessed to extract the same set of features that were used during training, and these features are then input to the SVM classifier. The SVM algorithm then maps the input image to a point in the feature space and determines which side of the decision boundary the point falls on. The category of the input image is then determined based on the position of the point relative to the decision boundary. The output of the model is a category label for the input image, which can be visualized using a scatter plot or a confusion matrix. The categories Weak, Acceptable, and Strong correspond to different levels of encryption strength, with strong encryption being the most secure [23].

4.3 Image encryption algorithms

Image encryption algorithms are designed to protect the confidentiality of digital images by converting them into a form that is unreadable or difficult to decipher without the appropriate decryption key. DNA encoding, Logistic Map, Rubik's cube image encryption, and Lorenz image encryption are some of the commonly used image encryption algorithms, each with its own strengths and weaknesses.

4.3.1 DNA encoding

An interdisciplinary field that is rapidly expanding is DNA computing. Many biological and algebraic operations based on DNA sequences have been proposed by researchers [3]. The binary system is used to express all information in the contemporary theory of the electronic

computer. Yet, DNA sequences rather than binary numbers are used to represent information in the DNA coding hypothesis[22].

4.3.2 Logistic map

A logistic map is a degree-two polynomial mapping (also known as a recurrence relation). It is frequently used as a classic illustration of how simple non-linear equations may produce complicated, chaotic behavior. The two effects that this nonlinear difference equation is meant to convey are Confusion and Diffusion [16].

4.3.3 Rubik's cube image encryption

For real-time Internet encryption and transmission applications, a rapid encryption/decryption technique has been developed. Moreover, it has quick encryption and decryption capabilities, making it appropriate for real-time Internet encryption and transmission applications [24].

4.3.4 Lorenz image encryption

In a 3D dynamical system, the Lorenz chaotic equation is defined by x, y, and z. It suggests that the "butterfly effect" is caused by one of the traditional chaotic systems. E.N. Lorenz documented and published the model for the first time in 1963. Regarding the initial system parameters, the Lorenz chaotic equation system exhibits chaotic behavior. The Lorenz system exhibits chaotic behavior that is far more complex than any 1D or 2D chaotic system [25].

4.4 Experimental setup

All experiments and model training were performed using a cloud-based platform like Google Colaboratory (Colab), which provides free access to GPU/TPU online resources. These platforms facilitate the efficient development, testing, and deployment of ML models in a collaborative environment. The default CPU configuration for Colab is an Intel Xeon with 2 vCPUs (virtual CPUs) and a RAM size of 13 GB. The Scikit-learn library, version 1.6.1, was utilized for implementing the SVM with the RBF kernel. All extracted features were normalized using "StandardScaler" before training the SVM model. This normalization ensures that all features contribute equally to the learning process and prevents any bias due to differing value ranges.

5 Implementation

5.1 Kernel selection

The RBF kernel can map input features into an infinite-dimensional space without requiring prior knowledge about the data's structure. Unlike the polynomial kernel, which may lead to overfitting with higher degrees or underfitting with lower ones, the RBF kernel offers a more flexible and robust decision boundary with fewer parameters. The sigmoid kernel, although inspired by neural networks, often suffers from convergence issues and fails to satisfy Mercer's condition for all parameter settings, thereby limiting its applicability. The execution of this project involves several critical phases: preparing the dataset, developing an SVM classifier-based model, creating a web interface, performing analysis and presenting results, performing evaluation and fine-tuning, and finally deploying the system. During development, the system was initially divided into two primary components.

5.2 Development of model

In this section, we present the most commonly adopted steps for developing an ML model for any application. These steps include data collection and preparation, model selection, training, evaluation, parameter tuning, and making predictions. Gather a dataset comprising encrypted images generated using various encryption algorithms, key sizes, and plaintexts. Apply the selected cryptosystem algorithms to the plaintext images, resulting in distinct datasets of encrypted images for each algorithm. Extract features from these encrypted images, including measures like energy, entropy, correlation, homogeneity, PSNR, and MSE. Construct an ML model employing the SVM classifier. Train this model using the features extracted from the encrypted images as input and their respective algorithm labels as output.

5.3 Integrating a machine learning model into a web application

The interface should allow users to upload an encrypted image and select the encryption algorithm used to encrypt the image. The interface should also include a button to initiate the analysis and display the result. The implementation of a Django project involves defining models, views, and URLs, configuring settings, and testing the app. Once the user uploads an image and selects an algorithm, the system should extract the features from the image and use the SVM classifier to predict the algorithm used to encrypt the image. The system should then categorize the result as Weak, Acceptable, or Strong based on the Accuracy of the prediction. In recent times, there has been a proliferation of encryption algorithms, encompassing chaos and transformation-based approaches. Assessing encryption statistics and the security parameters of these algorithms represents a common approach for evaluating their level of security. A presented a web-based interface as part of our research article, allowing users to assess the robustness of various cryptosystems through an ML model (see Figure 6.). The interface is designed to be user-friendly and time-efficient, streamlining the process of analyzing encrypted images. Users can select an encryption scheme and upload an encrypted image to the website. Leveraging an ML model developed based on current literature, the system conducts an in-depth analysis of

the uploaded encrypted image. Upon successful analysis, the system categorizes the results as Strong, Acceptable, or Weak, utilizing features extracted from the encrypted image. The user is then provided with the categorized outcomes. In summary, our proposed system offers a valuable tool for individuals and organizations seeking to evaluate the efficacy of their encryption algorithms. With its user-friendly and time-saving interface, the system facilitates a thorough analysis of encrypted images, empowering users to make informed decisions regarding their encryption protocols.

6 Performance parameters

The main performance parameters used are the confusion matrix, precision, precision, recall, and the F1 score. To define these metrics mathematically, the terminologies used are TP, FP, TN, and FN. TP and FP stand for true positive and false positive, while TN and FN represent true negative and false negative, respectively.

6.1 Confusion matrix

This is a performance metric used in ML classification tasks with two or more possible classes. It comprises a table presenting four distinct combinations of predicted and actual values.

6.2 Accuracy

Accuracy serves as a metric for assessing classification models. More formally, Accuracy is defined as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{8}$$

6.3 Precision

The ratio of true positive anticipated observations to all positive predicted observations is known as Precision. This can be written mathematically as:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

6.4 Recall

The sensitivity of the model is referred to as Recall. The model will be more sensitive to a higher Recall score. This quantifies the ratio of genuine positive observations to the number of real positive and fake negative observations combined. Recall can be calculated mathematically as:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

6.5 F1 Score

In assessing the performance of ML models, both Accuracy and F1 score play vital roles. Accuracy is essential when giving equal importance to true positive and true negative samples, while the F1 score becomes significant when there is a greater concern for false positive and false negative samples, and vice versa. F1 score is determined by:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{11}$$

7 Results and discussion

We utilized publicly accessible datasets, namely, Dataset 1 [12] and Dataset 2 [1], in our experimental work. Additionally, we generated Dataset 3 by randomly amalgamating samples from both Dataset 1 and Dataset 2.

Our experimentation involved the implementation and evaluation of various SVM kernels, including linear, polynomial, sigmoid, and RBF, on these datasets to assess their performance. We then employed an SVM with the radial basis function (RBF) as the kernel for training. The performance of the SVM model is evaluated using metrics such as Accuracy, Precision, Recall, and F1 score. These metrics can help determine the effectiveness of the SVM model in classifying encrypted messages and identifying potential security breaches.

7.1 Performance evaluation of machine learning model

The assessment of the developed model involves the use of three different data sets. In the upcoming sections, we will present these evaluations separately. To begin, we evaluated the model's performance using dataset 1. We divided this dataset into training and testing subsets with a 70:30 ratio and set the random state to 5. Our observations are presented in Table **??**, the following metrics for the respective categories:

Table 4: Assessing the performance of an SVM classifier with an RBF kernel using dataset 1

Performance Parameters	SVM Kernels RBF		
	Strong	Acceptable	Weak
Precision	100	88	100
Recall	100	100	83
F1 Score	100	93	91
Accuracy	98		

- For the "Acceptable" category, the developed model achieved a Precision of 88%.
- For the "strong" category, it exhibited a Precision of 100%.

– In the ”weak” category, the model also demonstrated a Precision of 100%.

The performance of the model is also presented using the confusion matrix as shown in Figure **??** This model is evaluated on the second test dataset (Dataset 2), which has been divided into training and testing subsets with a 60:40 ratio. A random state value of 5 is used to ensure reproducibility. The results presented in Table 5 indicate that the model performs exceptionally well, achieving an Accuracy rate of 97%, Precision of 100%, Recall of 88%, and an F1 score of 93% when categorizing input images encrypted with strong encryption. The ML model we've implemented, utilizing SVM, is applied to classify the image encryption level as either Strong, Acceptable, or Weak. This model is evaluated on the third test dataset (Dataset 3), which has been divided into training and testing subsets with a 60:40 ratio. A random state value of 15 is used to ensure reproducibility. We evaluated the performance of the model using various metrics, including Accuracy, Precision, Recall, and the F1 score. Table 6 results indicate that the model performs exceptionally well, achieving an Accuracy rate of 98%, Precision of 100%, Recall of 94%, and an F1 score of 97% when categorizing input images encrypted with strong encryption. The confusion matrix (Figure 3)illustrates the performance of the SVM classifier in predicting encryption strength levels: ”Strong,” ”Acceptable,” and ”Weak.” It shows high classification Accuracy with 9 correct predictions for ”Strong,” 8 for ”Acceptable,” and 6 for ”Weak,” with only one misclassification where a ”Weak” sample was incorrectly predicted as ”Strong. Similarly, Figures 4 and 5 illustrate the model's performance on Dataset 2 and Dataset 3, respectively, using confusion matrices as the evaluation metric.

Table 5: Assessing the performance of an SVM classifier with an RBF kernel using dataset 2

Performance Parameters	SVM Kernels RBF		
	Strong	Acceptable	Weak
Precision	100	91	100
Recall	88	100	100
F1 Score	93	95	100
Accuracy	97		

Table 6: Assessing the performance of an SVM classifier with an RBF kernel using dataset 3

Performance Parameters	SVM Kernels RBF		
	Strong	Acceptable	Weak
Precision	100	91	100
Recall	88	100	100
F1 Score	93	95	100
Accuracy	98		

Figure 3: Confusion matrix obtained using dataset 1

Figure 4: Confusion matrix obtained using dataset 2

7.2 Performance evaluation of the web application

We have integrated the SVM model into a user-friendly web application, developed using the Django technology stack, to classify image encryption levels as Strong, Acceptable, or Weak. The system produces a strong result when we choose the DNA encryption image encryption algorithm, as shown in Figure 6. The system displays the result as Acceptable in Figure 7 if we choose the Logistic Map image encryption algorithm, and for the weak encryption algorithm, the result is weak. The final product is a user-friendly web-based tool that can automatically analyze encrypted images and categorize the encryption algorithm used as Weak, Acceptable, or Strong based on the Accuracy of the prediction.

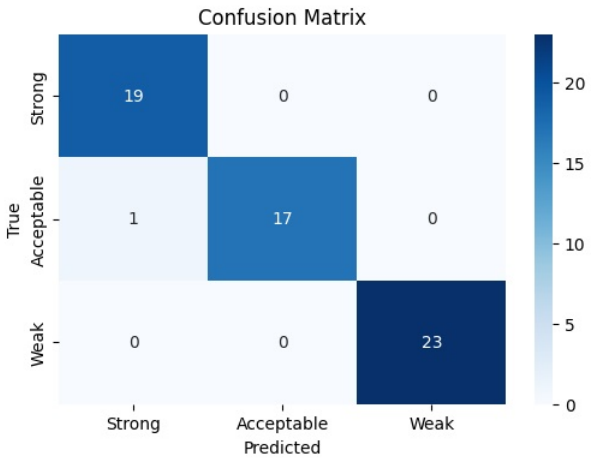


Figure 5: Confusion matrix obtained using dataset 3

Figure 6: Predicted categorical outcome: strong

7.3 Performance analysis

We have compared our proposed system's performance with the various already-published works of authors [26], [27], [28], and [29] by considering performance parameters like Accuracy, Precision, Recall, and F1 score. This comparative analysis is shown in Table 7.

Table 7: Comparative performance analysis

Methods/ Performance Parameters	Accuracy	Precision	Recall	F1 Score
Proposed System	98	100	94	97
[26]	82	96	90	92
[27]	92	93	91	83
[28]	77	90	93	81
[29]	89	87	92	92

Table 7 provides a comparative performance evaluation between the proposed system and four existing methods ([23] to [26]) using metrics such as Accuracy, Precision, Recall, and F1 Score. The proposed system demonstrates superior performance, achieving the highest values: 98% Accuracy, 100% Precision, 94% Recall, and 97% F1 Score. This enhanced performance can be attributed to the use of a larger training dataset in the proposed system compared to those used in the references.

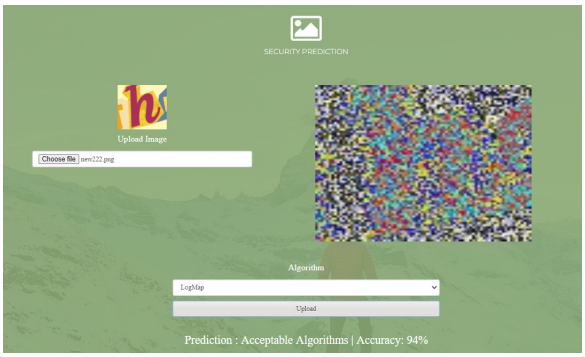


Figure 7: The categorical outcome is acceptable

7.4 Potential challenges and limitations

Key issues include latency due to network dependency, computational limitations on mobile devices for running heavier models, and ensuring responsive design for a consistent user experience.

8 Conclusion and future scope

This research proposes an SVM-based classification framework for predicting the strength of image encryption algorithms, emphasizing rapid identification and offering a user-friendly interface for efficient evaluation. Extensive experimentation with multiple kernel functions revealed that the RBF kernel achieves optimal performance in capturing the nonlinear relationships inherent in the statistical feature distributions of encrypted images. The system integrates a feature extraction pipeline with SVM modeling to classify encryption levels with high accuracy. A Django-powered web interface facilitates real-time user interaction, making the model deployable in practical security analysis contexts.

Evaluation in three benchmark data sets demonstrates the superiority of the model over existing methods, achieving accuracy up to 98%, and precision of 100% for strong encryption identification. Compared to traditional classifiers, the proposed approach offers improved discrimination capacity and operational efficiency.

The present work demonstrates the effectiveness of SVM in selecting suitable encryption techniques based on the security requirements of digital images. One promising direction is the integration of deep learning models, particularly Convolutional Neural Networks (CNNs). The research can be extended by utilizing large-scale and diverse datasets, such as domain-specific datasets (e.g., medical images) to test the robustness and generalizability of the proposed system under various security scenarios.

Code availability

The source code developed and used in this research is publicly available for reproducibility and further research. It

can be accessed at the following GitHub repository: Code

Acknowledgments

The authors acknowledge the use of AI-assisted language editing tools to enhance the clarity and readability of the manuscript. These tools were used solely for linguistic improvement and did not affect the scientific content or conclusions.

References

[1] Moatsum Alawida Abdul Nasir Khan Arslan Shafique, Abid Mehmood and Atta Ur Rehman Khan. "A novel machine learning technique for selecting suitable image encryption algorithms for iot applications". *Wireless Communications and Mobile Computing*, 2022:1–21, 2022. https://doi.org/10.1155/2022/5108331.

[2] Lu Xu, Zhi Li, Jian Li, and Wei Hua. "A novel bit-level image encryption algorithm based on chaotic maps". *Optics and Lasers in Engineering*, 78:17–25, 2016. https://doi.org/10.1016/j.optlaseng.2015.09.007.

[3] MA Ben Farah, R Guesmi, A Kachouri, and M Samet. "A novel chaos based optical image encryption using fractional fourier transform and dna sequence operation". *Optics & Laser Technology*, 121:105777, 2020. https://doi.org/10.1016/j.optlastec.2019.105777.

[4] Yuling Luo, Xue Ouyang, Junxiu Liu, and Lvchen Cao. "An image encryption method based on elliptic curve elgamal encryption and chaotic systems". *IEEE Access*, 7:38507–38522, 2019. https://doi.org/10.1109/access.2019.2906052.

[5] Shuqin Zhu and Congxu Zhu. "Plaintext-related image encryption algorithm based on block structure and five-dimensional chaotic map". *IEEE Access*, 7:147106–147118, 2019. https://doi.org/10.1109/access.2019.2946208.

[6] Shuqin Zhu and Congxu Zhu. "Secure image encryption algorithm based on hyperchaos and dynamic dna coding". *Entropy*, 22(7):772, 2020. https://doi.org/10.3390/e22070772.

[7] Hegui Zhu, Lewen Dai, Yating Liu, and Lijun Wu. "A three dimensional bit level image encryption algorithm with rubik's cube method". *Mathematics and Computers in Simulation*, 185:754–770, 2021. https://doi.org/10.1016/j.matcom.2021.02.009.

[8] Mohammed Benabdellah, Fakhita Regragui, Nourddine Zahid, and El Houssine Bouyakhf. "Encryption-compression of echographic images using fmt transform and des algorithm". *INFOCOMP Journal of Computer Science*, 6(4):36–42, Dec. 2007. https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/193.

[9] Chengye Zou, Qiang Zhang, Xiaopeng Wei, and Chanjuan Liu. "Image encryption based on improved lorenz system". *IEEE Access*, 8:75728–75740, 2020. https://doi.org/10.1109/access.2020.2988880.

[10] Minal Govind Avasare and Vishakha Vivek Kelkar. "Image encryption using chaos theory". In *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–6. IEEE, 2015. https://doi.org/10.1109/iccict.2015.7045687.

[11] M Kaur and VJEL Kumar. "Efficient image encryption method based on improved lorenz chaotic system". *Electronics Letters*, 54(9):562–564, 2018. https://doi.org/10.1049/el.2017.4426.

[12] Arslan Shafique, Jameel Ahmed, Wadii Boulila, Hamzah Ghandorh, Jawad Ahmad, and Mujeeb Ur Rehman. "Detecting the security level of various cryptosystems using machine learning models". *IEEE Access*, 9:9383–9393, 2020. https://doi.org/10.1109/access.2020.3046528.

[13] Ramani Sagar, Rutvij Jhaveri, and Carlos Borrego. "Applications in security and evasions in machine learning: a survey". *Electronics*, 9(1):97, 2020. https://doi.org/10.3390/electronics9010097.

[14] M Gomathi, S Gunasekar, S Abinesh, et al. "Detecting the security level of various cryptosystems using machine learning model". *International Journal of Progressive Research in Science and Engineering*, 3(05):25–31, 2022. https://journal.ijprse.com/index.php/ijprse/article/view/557.

[15] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. "Secure, privacy-preserving and federated machine learning in medical imaging". *Nature Machine Intelligence*, 2(6):305–311, 2020. https://doi.org/10.1038/s42256-020-0186-1.

[16] Jakub Oravec, Lubos Ovsenik, and Jan Papaj. "An image encryption algorithm using logistic map with plaintext-related parameter values. *Entropy*, 23(11):1373, 2021. https://doi.org/10.3390/e23111373.

[17] Jingang Guan. "Enhanced network security hybrid cloud workflow scheduling using levy-optimized

slime mould algorithm". *Informatica An International Journal of Computing and Informatics*, 49:111–126, 2025. https://doi.org/10.31449/inf.v49i18.7327.

[18] Hong shun Chen Lianqiu Liu1, Yongping Yang. "Application of new feature techniques for multimedia analysis in artificial neural networks by using image processing". *Informatica An International Journal of Computing and Informatics*, 48:113–124, 2024. https://doi.org/10.31449/inf.v48i11.5851.

[19] Yang Lei Lingyu Liao. "Image content forgery detection model combining pso and svm in electronic data forensics". *Informatica An International Journal of Computing and Informatics*, 48:113–124, 2024. https://doi.org/10.31449/inf.v48i8.5897.

[20] William S. Vincent. *Django for Beginners: Build websites with Python and Django*. WelcomeToCode, 1st edition, 2018. DOI not available.

[21] Daniel Rubio. *Beginning Django Web Application Development and Deployment with Python"*. Springer, 2017. https://link.springer.com/book/10.1007/978-1-4842-2787-9.

[22] Arslan Shafique and Jameel Ahmed. "Dynamic substitution-based encryption algorithm for highly correlated data". *Multidimensional Systems and Signal Processing*, 32:91–114, 2021. https://doi.org/10.1007/s11045-020-00730-3.

[23] Peter Harrington. *"Machine Learning in Action"*. Manning Publications, Shelter Island, NY, 2012. https://dl.acm.org/doi/10.5555/2361796.

[24] Khaled Loukhaoukha, Jean-Yves Chouinard, and Abdellah Berdai. "A secure image encryption algorithm based on rubik's cube principle". *Journal of Electrical and Computer Engineering*, 2012, 2012. https://doi.org/10.1155/2012/173931.

[25] Riguang Lin and Sheng Li. "An image encryption scheme based on lorenz hyperchaotic system and rsa algorithm". *Security and Communication Networks*, 2021, 2021. https://doi.org/10.1155/2021/5586959.

[26] Muhammad U Ilyas and Soltan Abed Alharbi. "Machine learning approaches to network intrusion detection for contemporary internet traffic". *Computing*, 104(5):1061–1076, 2022. https://doi.org/10.1007/s00607-021-01050-5.

[27] Sudhakar Sengan, Osamah Ibrahim Khalaf, Dilip Kumar Sharma, Abdulsattar Abdullah Hamad, et al. "Secured and privacy-based ids for healthcare systems on e-medical data using machine learning approach". *International Journal of Reliable and Quality E-Healthcare (IJRQEH)*, 11(3):1–11, 2022. https://doi.org/10.4018/ijrqeh.289175.

[28] Shashvi Mishra and Amit Kumar Tyagi. "The role of machine learning techniques in internet of things-based cloud applications". *Artificial intelligence-based internet of things systems*, pages 105–135, 2022. https://doi.org/10.1007/978-3-030-87059-1_4.

[29] Saket Acharya, Umashankar Rawat, and Roheet Bhatnagar. "A comprehensive review of android security: Threats, vulnerabilities, malware detection, and analysis". *Security and Communication Networks*, 2022, 2022. https://doi.org/10.1155/2022/7775917.

Improving YOLOv8m with Neck-Integrated Atrous Spatial Pyramid Pooling for Enhanced Detection of Small Fish and Jellyfish

Shaymaa K. Hussein, Zainab N. Nemer
College of Computer Science and Information Technology, University of Basrah, Iraq
E-Mail: shaimaa.khudher@uobasrah.edu.iq, Zainab.nemer@ uobasrah.edu.iq

Keywords: YOLOV8M, ASSP, underwater object detection

Received: March 19, 2025

The ocean depths are vital for biodiversity, as they host numerous marine species essential for maintaining ecosystem health. Accurate identification of aquatic creatures is critical for developing effective conservation strategies and sustainable marine resource management. However, aquatic environments pose distinct challenges, including light scattering, occlusions, and turbidity. This paper offers an improved YOLOv8m architecture that integrates ASPP module (Atrous Spatial Pyramid Pooling) that enhances multi scale feature extraction. Our proposed model, YOLOv8m-ASPP, was evaluated against the baseline YOLOv8m on the Brackish dataset, which contains 8,417 annotated images across six marine categories (“crab, jellyfish, fish, shrimp, small_fish, and starfish”).
The key architectural innovation involves integrating the ASPP module, with dilation rates of [2, 4, 6], into YOLOv8m's neck, specifically after the SPPF layer. This placement allows the ASPP module to process rich contextual features from the backbone, improving the ability of the model to capture objects at various scales. The YOLOv8m-ASPP model achieved an overall mAP@50 of 0.991 (a +0.002 increase) and a mAP@50-95 of 0.832 (a +0.004 increase) compared to the baseline YOLOv8m's 0.989 mAP@50 and 0.828 mAP@50-95. The modified model showed a precision of 0.980 and recall of 0.979., operating at approximately 60 FPS. Performance notably improved for challenging classes: the 'jellyfish' class mAP@50-95 rose to 0.757 (from the baseline's 0.730). Furthermore, robustness in small object detection was evident, with the 'small_fish' class achieving 0.970 mAP@50 (up from the baseline's 0.960 mAP@50). The findings demonstrate the effectiveness of the YOLOv8m-ASPP model for underwater ecological monitoring, successfully maintaining both detection accuracy and real-time processing capabilities. Future research could explore improved detection methods for small objects in environments with high turbidity.

Povzetek: Članek izboljšuje YOLOv8m z integracijo ASPP za izboljšano zaznavanje majhnih rib in meduz v podvodnih okoljih. Model ADMT doseže boljšo točnost in robustnost ob hkratnem ohranjanju obdelave v realnem času.

1 Introduction

The underwater object detection (UOD) Become a crucial computer vision technology for marine exploration and monitoring[1]. Its applications are diverse, ranging from ecological research and biodiversity protection to enhancing security, aiding autonomous underwater vehicle (AUV) navigation, and supporting underwater archaeology and search and rescue operations[2, 3]. UOD offers a non-invasive means of gathering crucial data from marine ecosystems, minimizing anthropogenic impact[4] However, the efficacy of UOD systems is often hindered by significant underwater challenges, including light attenuation, turbidity-induced visual degradation, and dynamic environmental conditions that impair detection accuracy[5, 6].

Considering these restrictions, this study suggests and evaluates a novel hybrid deep learning architecture that strategically integrates the ASPP module, known for its effectiveness in multi-scale feature extraction[7], with the robust YOLOv8 m detection framework. The primary research objective is to investigate whether this integration can achieve measurable improvements in detection accuracy and robustness, particularly under conditions of poor visibility, while striving to maintain real-time processing capabilities (e.g., targeting >30 FPS). Specifically, this work aims for a notable enhancement mean average precision (mAP@50-95) compared to the traditional Yolov8m baseline. Emphasis is placed on scenarios demanding both high accuracy (e.g., targeting >95% recall for critical detections) and low latency (e.g., <50ms inference time), pertinent to applications such as AUV path planning and the detection of illegal fishing activities [8].

The proposed architecture is designed to overcome key limitations in existing UOD systems. Utilizing ASPP with its multi-scale feature analysis capabilities, the model

seeks to make it easier to identify partially occluded objects or objects with variable sizes commonly found in underwater turbid scenes. In addition, inspired by the efficient design in YOLOv8m, architecture tries to maintain computational efficiency for real-time running on underwater robotic platforms. A systematic evaluation of the model's performance will be assessed. consistency across varied underwater conditions. This investigation is particularly focused on enhancing the detection of small targets, a persistent challenge in UOD.

This work has the following contributions: (1) It proposes a new hybrid structure that combines the multi-scale feature learning ability of ASPP with the efficient detection framework of YOLOv8m to the UOD problem. (2) We can present an in-depth performance evaluation of this architecture over the baseline systems on the Brackish dataset by showing improvements in standard metrics. (3) It studies the real-time processing of this model. (4) It offers insight into the model's efficacy in detecting hard-to-detect objects (e.g., small or partially occluded marine life), and then directs future work for UOD. The results are expected to be highly applicable to operational needs in marine conservation, underwater structure survey, and autonomous navigation, where robust object recognition in unfavorable conditions is vital.

2 Related work

2.1 Introduction

This section highlights the primary challenges facing UOD while discussing related work.It also investigates the development of object detection approaches, particularly deep learning-based methods, such as the You Only Look Once (YOLO) family of methods, analyses the ways of improving feature representation and performance in real time. In this research, we concentrate on the enhancement of YOLOv8m for UOD using ASPP.

2.2 Key challenges in underwater object detection (UOD)

Underwater habitats bring distinct challenges, set of conditions for object detection. Poor visibility, stemming from light attenuation and scattering, significantly degrades image quality. This is often compounded by issues including low contrast and color distortion. which traditional image enhancement techniques attempt to address [9, 10]. Furthermore, object ambiguity due to marine organisms blending with complex backgrounds, detecting small or distant objects can be challenging, especially in the presence of turbidity or marine particulates further complicate reliable detection[11].Beyond these visual impediments, UOD efforts are also constrained by dataset limitations, as publicly available underwater datasets are often small, imbalanced, or lack diverse annotations, hindering the advancement of resilient and universally applicable models [12]. Finally, the computational resources available on autonomous underwater platforms are typically restricted, necessitating efficient algorithms that can perform accurately in real-time without excessive power consumption[13].

2.3 Object detection's evolution techniques and their application in UOD

2.3.1 Traditional object detection

Traditional object detection frameworks, such as the Viola-Jones Detector and feature extraction, often paired with machine learning techniques like SVM and Random Forests for high accuracy. In underwater image enhancement, histogram-based methods (e.g., HE, A HE, CLAHE) and Retinex-based methods (e.g., MSR, MSRCR) address challenges like color distortion and low contrast but face trade-offs in computational complexity and real-time performance. While these methods improve image quality, further advancements are needed to enhance robustness and efficiency in challenging underwater environments.[9, 10].

2.3.2 The development of deep learning at UOD

Deep learning-based object detection methodologies employ Convolutional Neural Networks (CNNs) to autonomously extract features from images, replacing traditional methods and improving generalization and robustness[10]. Advances in hardware, such as GPUs, have addressed challenges like large data demands and long training times. Object detection involves classifying and localizing objects using bounding boxes. Detectors can be either two or single stage. Two-stage detectors, such as R-CNN and Faster R-CNN, create regions of Interest (ROIs) and then classify, offering higher accuracy but with complex architectures and slower inference. Detectors like YOLO and SSD predict class and bounding box probabilities in a single step, providing simpler, faster, and more efficient solutions, albeit with slightly lower accuracy. YOLO is popular for its simplicity, speed, and real-time capabilities, balancing accuracy and efficiency for practical applications[14].

2.3.3 The YOLO series and its application in UOD

Redmon et al.'s 2016 introduction transformed real time object detection into a regression problem, revolutionizing the field. Divide the input image into a grid, then estimate bounding boxes and probability of classes for each cell. (see Figure 1).

.

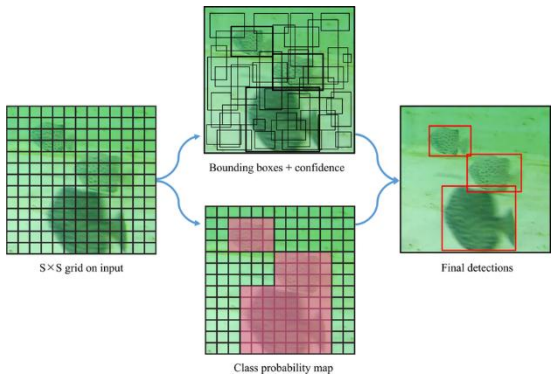


Figure 1: Principle of the YOLO algorithm [15]

Subsequent versions introduced significant enhancements: YOLOv2 enhances accuracy and recall using anchor boxes and multi-scale training; YOLOv3 incorporated multi-scale predictions via a deeper backbone and a Feature Pyramid Network (FPN)-like structure; YOLOv4 and YOLOv5 focused on optimizing network architecture, data augmentation, and training techniques for better performance and efficiency. More recent iterations like YOLOv6 and YOLOv7 further refined efficiency with concepts such as task-aligned heads and "Extended Efficient Layer Aggregation Networks (E-ELAN)", while approaches like DAMO-YOLO explored Neural Architecture Search (NAS) for dynamic structures. This has been marginalized in YOLOv8, the baseline for our study, which featured an anchor-free design, an optimized C2f module and decoupled head while extending from single-task learning such as translation to multi-task learning from segmentation and pose estimation, further establishing YOLO as the de-facto framework for real-time object detection[16].

YOLO versions There have been adapted several YOLO versions in the UOD domain. For example, one of such adaptations which relied on YOLOv4 architecture to balance detection accuracy and computational speed was able to achieve competitive results on the challenging underwater datasets. This specific model achieved a Mean Average Precision (mAP) equal to 81.67% on PASCAL VOC dataset and 92.65% on brackish water datasets. It also retained a processing speed that is appropriate for near real-time application, clocking at 44.22 frames per second (FPS) and succeeded in detecting underwater objects even when faced with low visibility and color lied [17] Acknowledging that, for resource-limited devices, even faster inference may be required, lightweight models such as YOLOv4-tiny are also investigated for UOD. But, despite their speed, YOLOv4-tiny is not perfect with detection accuracy. To overcome this, we propose an approach that leverages YOLO-UOD on top of YOLOv4-tiny. YOLO-UOD is reported to achieve a dramatically higher mAP of 87.88% on the Brackish underwater dataset than the 77.38% of YOLOv4-tiny and even more than YOLOv5s and YOLOv5m with faster speeds and thus balancing the speed and accuracy well[18]. In the same veins, there are also attempts to improve the previous foundational methods YOLOv3 for this specific domain. in proceedings was used as a base configuration since YOLOv3 had been previously demonstrated to be stable in a wide range of Open Images and was selected as one of the strongest single object detection model for the most difficult object classes. This work specifically tuned U-YOLOv3 for underwater object detection and reported a 2-10% higher mean Average Precision (mAP) than the original YOLOv3 on both the Brackish and Trash ICRA19 datasets[19]. Other approaches have focused on newer variants like YOLOv8 or specialized models such as LFN-YOLO, which is optimized for underwater environments. The LFN-YOLO model, for example, achieved a mean Average Precision (mAP) of 82.1% on the URPC dataset and an impressive 97.5% on the Brackish dataset, indicating its high detection accuracy in these specific contexts[20]. Similarly, YOLOv7 has been selected as a base for underwater target detection enhancements due to its advanced capabilities in Real-time applications require both accuracy and speed is the YOLOv7-AC mode, leveraged YOLOv7's architecture to improve feature extraction and overall performance in challenging underwater conditions. The YOLOv7-AC model attained an average precision (mAP) of 89.6% on the URPC dataset.and 97.4% on the Brackish dataset. Notably, it demonstrated improved detection efficiency, particularly in specific categories like echinus, which reached the average precision (AP) of 92.2%. This enhanced model reportedly outperformed other popular target detection models, showing an improvement of 1.1% over the original YOLOv7 and even greater gains when compared to YOLOv6, YOLOv5s, and SSD [21].

2.4 Addressing specific UOD challenges with advanced techniques

2.4.1 Enhancing feature representation and multi-scale detection

The variable sizes of underwater objects and visibility degradation necessitate robust multi-scale feature representation. Feature Pyramid Networks (FPNs) are widely adopted for this. Zhang et al. (2021) proposed a Modified Attentional Feature Fusion (AFFM) for the FPN structure in their lightweight YOLOv4, based on a" Multi-Scale Channel Attention Module (MS-CAM)", to better fuse semantic and scale-inconsistent features for small underwater targets[17]. Alongside these CNN-based enhancements, alternative architectures such as DEtection TRansformers (DETR) have been investigated for their ability to utilize global contextual information in UOD. For instance, Ali et al. employed DETR for marine object detection, utilizing a ResNet50 backbone to produce multi-resolution feature maps, thereby enhancing the model's capability to detect objects in complex underwater environments. Their DETR model was fine-tuned on the Brackish dataset to specifically address challenges like low visibility and complex backgrounds. This transformer-based approach achieved a significant mean Average Precision (mAP) of 0.648 on the test set,

reportedly outperforming other detectors like YOLOv3 (which achieved mAP of 0.3893 on the same dataset) in their study. Such findings suggest that transformer-based models offer considerable potential for enhancing detection accuracy in challenging underwater scenarios, paving the way for further advancements in marine object detection technologies by leveraging their distinct approach to feature representation and global context understanding.[1] Further advancing feature learning and fusion, Tian et al. (2024) utilized YOLOX as a baseline and introduced a novel Level, Channel, Spatial (LCS) multi-attention module. The core innovation of their work lies in the LCS module's approach to feature fusion, which is designed to significantly improve the model's learning ability by compelling it to focus on critical scale, channel, and spatial aspects of the features. This enhanced feature representation and contextual understanding resulted in their model achieving a mean Average Precision (mAP) of 77.32%[22]. Similarly, Liu et al. (2025), they have also designed their LFN-YOLO model based on the YOLOv8n model to concentrate on robust feature fusion for accommodating the different scales of objects under water. In the neck part of the model, the shape decoder used a GFPN to powerfully fuse the geometric details combining low-level feature maps and the abundant semantic information derived from high-level feature maps. The aim of this GFPN embedding was to enhance the network's adaptability to features of varying scales and small object detection by efficiently propagating information across different levels of features using reparametrized elements. Their LFN-YOLO obtains a mAP@0. 5 of 97.5% on the Brackish dataset, demonstrating the effectiveness of their feature fusion scheme for UOD[20]. Atrous Spatial Pyramid Pooling (ASPP) is a special powerful tool to capture long-range context with multi-scale information and no resolution reduction, which is very important for both semantic segmentation and object detection. The ASPP module which was inspired in DeepLab series exploits parallel atrous (dilated) convolutions with different rates to cover multi-scale context and scale of objects to enlarge the receptive field and gather various contextual information. It is an essential ability-skill in remote sensing and consequently in demanding UOD scenarios, where objects can be seen in very different sizes and environments. Motivated by these advantages, Hu et al. (2024) published ASPP+-LANet for high-resolution remote sensing image segmentation, in which an advanced ASPP module (denoted ASPP+) is proposed according to the original ASPP by adding on an another feature extraction path, reconsidering the dilation rates, and integrating a Coordinate Attention (CA) mechanism. Their ablation studies showed that including their ASPP+ module dramatically enhanced segmentation results of ground object targets in different scales[23]. Further, Sivanpillai et al. (2023) indicated that ASPP module in DeepLabV3+ can remarkably improve the performance of water body segmentation, even for the small and partly occluded ones. Their research suggested that multi-scale feature extraction of ASPP significantly satisfied the requirements for the model to predict the water boundaries and details of water bodies with different sizes since DeepLabV3+ with ASPP far outperformed versions without it. These results together confirm the performance applicability of ASPP to complex tasks containing multi-scale features and occlusions, and similar phenomena appear to exist in the underwater object detection, which constitutes a strong foundation of our proposed architectural modification [7].

2.4.2 Lightweight models and real-time performance in UOD

Operational constraints of AUVs necessitate computationally efficient models. Zhang et al. (2021) focused on this by adapting YOLOv4 with a MobileNetV2 backbone and depth-wise separable convolutions, achieving over 44 FPS with a significantly reduced model size[17]. LFN-YOLO (Liu et al., 2025), based on YOLOv8n, used RepGhost and SPD-Conv, resulting in a 5.9MB model with 58-63 FPS on Brackish[20]. Tian et al. (2024) used YOLOX with a GhostNet backbone and an LCS attention module, achieving a 18.5MB model and 55.54 FPS on URPC2021[22]. These efforts highlight the ongoing pursuit of balancing detection accuracy with computational feasibility.

The reviewed literature demonstrates significant advancements in Underwater Object Detection (UOD), largely driven by the growth of deep learning approaches. Convolutional Neural Networks, notably the You Only Look Once (YOLO) series, have become foundational because of their mix of speed and accuracy, with various iterations being adapted for the challenging aquatic domain. Researchers have explored various strategies to mitigate the adverse effects of underwater conditions, including poor visibility, color distortion, and turbidity. These include advanced image enhancement pre-processing, specialized backbone networks (e.g., MobileNetV2, RepGhost, and GhostNet,) various feature fusion mechanisms, such as enhanced Feature Pyramid Networks (FPNs) and attention modules (e.g., GAM, AFFM), to improve multi-scale detection and robustness. The pursuit of lightweight models for deployment on resource-constrained underwater platforms has also been a prominent theme.

Despite these efforts, persistent challenges remain. Many existing models, while improving general UOD performance, still struggle with the reliable detection of small, occluded, or camouflaged objects, particularly under varying turbidity and illumination levels. For instance, LFN-YOLO, despite its lightweight design and high accuracy, can still face issues with false positives and missed detections in highly variable environments and with significant object scale changes. Similarly, models like YOLOv7-AC may exhibit errors in highly complex underwater scenes. While techniques like specialized detection heads and dedicated feature extractors for small objects show promise, achieving consistent high performance across diverse underwater conditions remains an objective. Furthermore, while context aggregation modules like Atrous Spatial Pyramid Pooling (ASPP) have proven highly effective for capturing multi-

scale information in fields like semantic segmentation, and can enhance object detectors, their optimal integration and impact within the neck of recent, efficient YOLO architectures like YOLOv8m—specifically for enhancing the detection of problematic marine classes such as 'small_fish', 'jellyfish', while maintaining robust real-time processing capabilities—are not yet fully explored in the UOD literature. The trade-off between added Computational complexity from such modules and the achievable accuracy gains, especially for real-world deployment on AUVs, also warrants careful investigation. This study, therefore, aims to address this gap by proposing and evaluating a YOLOv8m architecture enhanced with a strategically integrated ASPP module.

The research investigates the efficacy of this approach in improving multi-scale feature representation and contextual understanding, with a specific focus on enhancing detection accuracy and robustness for challenging underwater targets, while critically assessing the impact on real-time performance. To further consolidate the reviewed literature and provide a direct comparative overview, Table 1 summarizes key characteristics and performance metrics of several prominent object detection models discussed, particularly those adapted for or relevant to the underwater domain.

Table 1: Summary of selected underwater object detection models from related work (primarily on brackish dataset)

Source (Placeholder)	Model	Base Architecture	Key Modifications/Focus	Dataset(s) Used	Precision	Recall	mAP@0.5	mAP@0.5:0.95	FPS	Key Limitations Highlighted
Ali et al. [1]	YOLOv4 (general reference)	YOLOv4	Standard architecture	Brackish	-	-	0.9356	(not specified with mAP50 in Ali et al.)	-	(General challenges of UOD, less optimized than specific UOD variants)
Zhao et al. [18]	YOLO-UOD	YOLOv4-tiny	Symmetric dilated convolution module, Symmetric FPN-Attention module, Label smoothing	Brackish	-	-	-	0.8788	9.24 (Jetson Nano)	Some inference speed loss vs YOLOv4-tiny; needs more scene validation.
Sarkar et al [19]	U-YOLOv3	YOLOv3	MIRNet enhancement, K-means++ anchors, SPP layer, modified classification layers.	Brackish, Trash	0.88 (Brackish, 8k iter.)	0.87 (Brackish, 8k iter.)	-	~0.80 (Brackish, inferred from 10% gain over YOLOv3's ~0.72)	- (YOLOv3 was ~104 FPS on server)	Higher training time; 63.25M parameters. Lacks in small/dense objects compared to its own enhancements.
Liu et al [20]	LFN-YOLO	YOLOv8n	RepGhost, SPD-Conv, Generalized FPN (GFPN), CLLAHead, DFL loss. Focus on lightweight, small object detection, feature fusion.	Brackish, URPC	0.974	0.954	0.975	0.798	63 (Jetson AGX Orin implied)	Faces FPs/FNs in highly variable environments; issues with very small objects in complex backgrounds or significant scale changes.
Liu et al. [21]	YOLOv7-AC	YOLOv7	E-ELAN Enhanced, ResNet-ACmix Added, GAM Incorporated, K-means++ Anchors	Brackish,	98.2	95.2	97.4	73.7	-	not suitable for real-time, Robust multi-scale feature representation is needed
Ali et al. [1]	DETR	Transformer	ResNet50 backbone, fine-tuned for UOD. Focus on global contextual information.	Brackish	-	-	0.951	0.648	-	Evaluated on limited classes; transformers can be computationally heavy.

3 Methodology

This section delineates the research methodology employed to develop and evaluate the proposed enhanced object detection model for underwater environments. It details the baseline architecture, the proposed architectural modifications involving Atrous Spatial Pyramid Pooling (ASPP), the dataset utilized, experimental setup, training procedures, and the metrics employed for performance evaluation. The overall pipeline of our proposed approach, from data input to final evaluation, is summarized in Algorithm as illustrated in **Figure 2.**

Algorithm : Summarized Pipeline for YOLOv8m-ASPP Underwater Object Detection
Input:
•Raw Underwater Image Dataset
•YOLOv8m-ASPP Model Configuration
•Training Hyperparameters
Output:
•Trained YOLOv8m-ASPP model
•Detection Results (bounding boxes, classes, scores)
•Performance Metrics (mAP, Precision, Recall, FPS)
Procedure:
1.Data Preparation:
•Acquire and clean the dataset (handle orphaned/corrupt files).
•Standardize image dimensions (e.g., to 640x640) and transform bounding box coordinates.
•Partition into training, validation, and test sets.
•Apply training-time data augmentations.
2.Model Training (YOLOv8m-ASPP):
•Initialize model with pre-trained weights.
•**For** each epoch:
•**For** each batch in training data:
•Perform forward pass (Backbone → SPPF → **ASPP** → Neck → Head).
•Calculate composite loss.
•Perform backward pass and update weights.
•Validate model and save best performing weights.
•*Employ fixed random seeds for reproducible experimental runs.*
3.Model Evaluation:
•Load the best trained model.
•Perform inference on the test set
•Calculate quantitative metrics (mAP, P, R, FPS, GFLOPs, Parameters).
•Conduct statistical analysis on results from multiple runs.
•Perform qualitative analysis (visualize detections, failures, confusion matrices, activation maps).

Figure 2: Summary algorithm of the model pipeline.

3.1 Overview of the proposed approach

The primary aim of this research is to improve the performance of the YOLOv8m model for UOD, focusing particularly on hard-object detection in varying visibility conditions. The fundamental approach is to incorporate an ASPP module in the YOLOv8m pipeline. This adaptation aims to improve the ability of the model to extract features at various scales and to understand context, thereby improving detection accuracy and robustness without unduly compromising its real-time processing capabilities.

3.2 Baseline architecture: YOLOv8m

The YOLOv8m model, developed by Ultralytics, serves as the baseline for this study. This single-stage object detector is highly accurate and fast, making it a cutting-edge technology. Key architectural components of YOLOv8m include:

- A backbone network, drawing inspiration from CSPDarknet, which utilizes C2f (CSP Bottleneck with 2 convolutions) modules for efficient feature extraction at various scales.
- A Spatial Pyramid Pooling Fast (SPPF) module located at the termination of the backbone, designed to aggregate contextual information by pooling features at different scales with minimal computational overhead.[24]
- A Path Aggregation Network (PANet)-inspired neck structure that facilitates effective fusion of features from different backbone levels (e.g., P3, P4, P5) to enhance multi-scale feature representation.

An anchor-free, the decoupled Bounding boxes and class are predicted by the detective head independently, which contributes to Faster convergence and better performance [25]

3.3 Proposed YOLOv8m-ASPP architecture

To address the challenges of UOD, particularly the detection of objects of varying sizes and those in cluttered or low-visibility scenes, we propose an enhanced architecture, hereafter referred to as YOLOv8m-ASPP. This architecture incorporates an ASPP module, a technique proven effective in semantic segmentation for capturing rich contextual information at multiple scales without significant loss of spatial resolution.

3.3.1 Atrous Spatial Pyramid Pooling (ASPP) Module

Integrated the ASPP module into our model is designed to Consider The entering convolutional feature layer includes numerous parallel atrous (dilated) convolutions with Various dilation rates. This allows the model to capture contextual information via a broader range of sources.the ASPP block comprises:

(1) One 1x1 convolutional branch. (2) Three 3x3 atrous convolutional branches with dilation rates of 2, 4, and 6, respectively. Each of these parallel branches processes the input feature map generates an intermediate feature map with [Specify, e.g., 192] channels. The outputs from these four branches are combined along a channel dimension. Finally, the combined features are fused the multi-scale information and refine the output using a 1x1 convolution., resulting in a feature map with [Specify, e.g., 768] channels, consistent with the input channel dimension to this block, as illustrated in **Figure 3.**

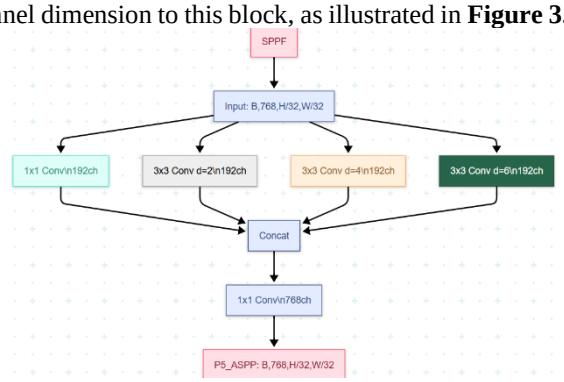


Figure 3: ASSP Architecture is used to enhance YOLOV8M

3.3.2 Integration of ASPP into YOLOv8m

The ASPP module is strategically inserted into the YOLOv8m backbone immediately following the SPPF module. The output feature map from the SPPF layer (designated as P5, with dimensions [e.g., 768 x H/32 x W/32] for an input image of H x W) serves as the input to our ASPP block. Consequently, the output of the ASPP module, which retains the same spatial and channel dimensions as its input, effectively becomes the enhanced P5 feature map. This enriched P5 feature map is then propagated to the PANet-style neck, where it undergoes

upsampling and fusion with feature maps from earlier stages of the backbone (P4 and P3) to generate multi-scale features for the detection heads.

A detailed visual representation of the proposed YOLOv8m-ASPP architecture, illustrating the modified backbone, the structure of the ASPP module, and the flow of tensors with their respective dimensions at critical stages, is provided in Figure 4

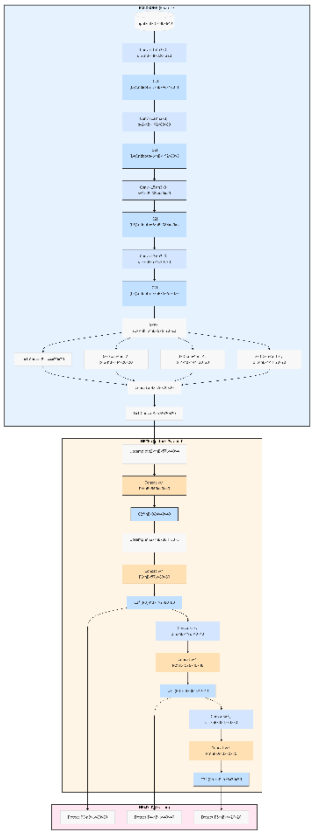


Figure 4: Architecture diagram of YOLOV8M-ASSP

4 Experimental and results

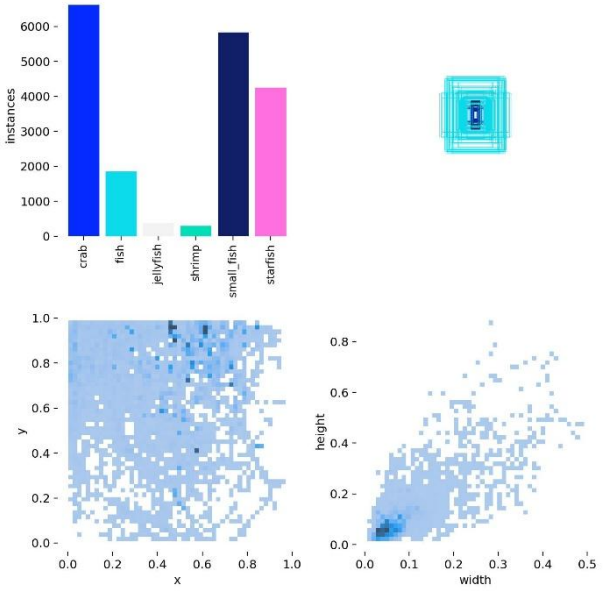
4.1 Data preprocessing

Data preprocessing is a critical stage in the machine learning pipeline, particularly for underwater object detection, where raw imagery often contains artifacts, noise, and inconsistencies. To ensure robust model performance, we implemented a customized two-stage preprocessing protocol for the Brackish dataset, addressing dataset-specific challenges while maintaining annotation fidelity.

4.1.1 Dataset overview

The Brackish dataset, employed in this study (Roboflow, 2021; Aalborg University, 2020), is a significant publicly available European resource for underwater object detection. It consists of 14,674 images captured at a 9-meter depth using a fixed camera system on Denmark's Limfjords bridge [26]. This professionally annotated dataset features six marine categories pertinent to ecological monitoring—fish, crab, small_fish, jellyfish, starfish, and shrimp—with standardized bounding box annotations that render it highly valuable for training and benchmarking underwater computer vision systems. For our experiments, the dataset was partitioned into training (80%), validation (10%), and testing (10%) subsets, following established deep learning practices.

Figure 5 provides a statistical overview of the annotated object characteristics within the dataset. A significant class imbalance is evident (Figure 5, top left), with dominant categories such as 'crab' and 'small_fish' (over 5,900 instances each) contrasting sharply with infrequent classes like 'jellyfish' and 'shrimp' (under 400 instances each). Analysis of object dimensions (Figure 5, bottom-right) reveals the prevalence of relatively small objects, with most normalized heights and widths concentrated below 0.3, alongside a broader spectrum of sizes. Spatially, objects are generally dispersed across the image frames but tend to avoid the extreme peripheries, as indicated by their center distributions and typical bounding box placements (Figure 5, bottom-left and top-right, respectively). These dataset-specific attributes—class imbalance, scale variability, and spatial distribution—are crucial considerations for robust model training and performance evaluation in underwater object detection.



Figuar 5: Statistical visualization of key characteristics of the Brackish dataset annotations.

4.1.2 Preprocessing pipeline

Preprocessing Pipeline: The data undergoes two critical stages: data cleaning to ensure annotation integrity and dimensional standardization for compatibility with modern detection architectures. The cleaning phase systematically removes orphan samples (images without annotations), filters corrupted or empty annotation files, and eliminates orphaned labels to maintain dataset consistency. This prevents training disruptions, improves

learning from clean supervisory signals, and enhances computational efficiency.

For dimensional standardization, images are resized from their native resolution (950×540) to a uniform 640×640 format using bilinear interpolation, preserving structural features. Concurrently, bounding box annotations undergo mathematically precise coordinate transformations, scaling their center points and dimensions to maintain geometric fidelity. This dual normalization ensures accurate spatial relationships—critical for handling variable object scales and occlusions—while optimizing batch processing efficiency and framework compatibility. Together, these steps enhance model robustness, detection accuracy, and training stability.

4.2 Experimental setup and training details

Table 2: The key configuration parameters

Python Version	3.10.12
Ultralytics Version	8.3.13
PyTorch Version	2.5.0 + cu121
GPU	Tesla T4
GPU Memory	15102 MB
Image size	640*640
Epochs	105
Batch Size	16
Learning Rate	0.01
Optimizer	SGD (selected by optimizer=auto)
Initial Learning Rate (lr0)	0.1
Final Learning Rate	0.1

The development and training environment for this study was established using Python 3.11.12**,** with the Ultralytics YOLOv8 framework (version 8.3.13) serving as the core platform for model implementation and training. Deep learning computations were accelerated using PyTorch 2.6.0 compiled with CUDA 12.4**,** executed on a Tesla T4 GPU with 15095 MiB of dedicated memory**.**

Both the baseline YOLOv8m and the proposed YOLOv8m-ASPP models were trained on images resized to 640x640 pixels for a total of 105 epochs**.** A batch size of 16 was employed, determined as an optimal balance between maximizing GPU memory utilization on the Tesla T4 and maintaining training stability. The models were initialized with **pre-**trained weights (typically from the COCO dataset for YOLOv8 backbones) to leverage transfer learning and potentially accelerate convergence. The Ultralytics optimizer = auto setting was utilized, which automatically selected the Stochastic Gradient Descent (SGD) optimizer with a learning rate (lr0) of 0.01 and momentum of 0.9**.** This automated selection is designed to provide robust and effective optimization parameters for the given task and model. Key hyperparameters included **a** weight decay of 0.0005**,** 3.0 warmup epochs with a starting momentum of 0.8 and a warmup bias learning rate of 0.1. Automatic Mixed Precision (AMP) enables expedite training and reduces memory footprint.

To enhance model robustness and generalization against the diverse and challenging conditions of underwater environments, a suite of data augmentation techniques was applied during training. These included mosaic augmentation (active for the first 95 epochs, as close_mosaic=10), horizontal flips (fliplr=0.5), adjustments to Hue, Saturation, and Value (hsv_h=0.015, hsv_s=0.7, hsv_v=0.4), random translation (0.1), scaling (0.5), and random erasing (0.4) were incorporated. The training process was monitored using validation loss with a patience of 100 **epochs** to prevent premature stopping and ensure comprehensive learning. This meticulously configured training pipeline was designed to facilitate robust model development and achieve optimal performance on the underwater object detection task. The key configuration parameters are summarized in Table 2.

4.3 Metrics for assessing object detection performance

In this contribution, we concentrate on domain-specific applications such as underwater robots, where the evaluation of object identification models is even more crucial. One of the most critical measures to assess the effectiveness of such models is Mean Average Precision (mAP) at different Intersection over Union (IoU) criteria. Important Metrics:

Mean Average Precision (mAP): This measure evaluates the performance of models across a wide range of classes by combining precision and recall. Precision is calculated by dividing the number of true positives (TP) by the total number of true positives (TP) and false positives (FP), indicates how accurate our positive predictions are.

$$Precision = \frac{TP}{TP+FP} \quad [27]$$

Recall: This measures the model's capacity to detect all relevant cases and is calculated as a ratio of true positives (TP) to the total number of true positives (TP) and false negatives (FN):

$$Recall = \frac{TP}{TP+FN} \quad [28]$$

Mean Average Precision at Specific IoU Thresholds

- mAP@0.5: This metric calculates the average AP with an IoU threshold of 0.5, i.e., a prediction is considered valid if it overlaps with the ground truth by at least 50%.
- mAP@0.5:0.95: This statistic offers a more thorough assessment by averaging mAP at various IoU values in increments of 0.05 from 0.5 to 0.95. It provides a comprehensive summary of the model's detection accuracy across a range of overlap levels [29].

4.4 Analysis and results of the experiment

This section provides a thorough assessment of the proposed YOLOv8m-ASPP model in comparison to the baseline YOLOv8m architecture for Underwater Object Detection (UOD) on the Brackish dataset. The analysis encompasses overall performance metrics, statistical validation of improvements, a detailed class-specific performance breakdown, and an assessment of computational efficiency.

4.4.1 Overall performance comparison

The efficacy of integrating the (ASPP) module into the YOLOv8m framework was primarily assessed by comparing key object detection metrics against the baseline YOLOv8m model. A summary of these performance indicators is illustrated in **Table 3**.

Table 3: Performance comparison of YOLOv8m baseline and YOLOv8m-ASPP on the Brackish dataset.

Metrices	YOLOV8M	YOLOv8m-ASPP
Precision	0.983	0.98
Recall	0.976	0.977
mAP@50	0.989	0.991
mAP@50-95	0.827	0.836
GFLOPs	67.4	69.6
FPS	106.3	62.5
parameters	23,224,594	25,881,106

The proposed YOLOv8m-ASPP model demonstrated enhanced overall performance. It achieved a mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) of 0.991, an improvement from the baseline's 0.989. More notably, for the comprehensive mAP@0.50-0.95 metric, the YOLOv8m-ASPP model scored 0.836, surpassing the baseline's 0.827. This represents an absolute increase of 0.2% in mAP@0.5 and a more significant 0.9% increase in mAP@0.50-0.95, indicating improved localization accuracy across various IoU thresholds.

In terms of detection sensitivity and predictive accuracy, the YOLOv8m-ASPP model achieved a recall of 0.977 (baseline: 0.976) and a precision of 0.980 (baseline: 0.983). While there was a marginal decrease in precision, the slight increase in recall suggests the enhanced model is more effective at identifying all relevant objects. Computationally, the integration of ASPP resulted in a modest increase in model complexity, with parameters rising from 23.20 million to 25.86 million and GFLOPs from 67.4 to 69.6. The average inference time on a Tesla T4 GPU for the YOLOv8m-ASPP model was 16.5 ms per image/60.5 FPS, compared to 9.4 ms /106.3 FPS for the baseline YOLOv8m. Despite this increase, the enhanced model continues to operate at a speed conducive to near real-time applications.

4.4.2 Analysis of class-specific performance and statistical insights

A thorough performance by class evaluation was conducted to understand the impact of the ASPP module on individual object categories within the Brackish dataset. The mAP@0.50-0.95 for each class, before and after the ASPP integration, is presented in Table 4.

Table 4: Class-Specific mAP@0.50-0.95 Comparison for YOLOv8m Baseline and YOLOv8m-ASPP

CLASS	YOLOV8M (mAP@50-95)	YOLOV8M-ASSP (mAP@50-95)	Improvement
carb	0.909	0.915	+0.006
fish	0.862	0.855	-0.007
jellyfish	0.73	0.758	+0.028
shrimp	0.784	0.789	+0.005
Small_fish	0.669	0.707	+0.008
starfish	0.986	0.992	+0.006

The YOLOv8m-ASPP model demonstrated notable performance gains for several challenging classes. Specifically, the 'jellyfish' class exhibited a substantial improvement in mAP@0.50-0.95, rising from 0.73 for the baseline to 0.758 for the ASPP-enhanced model (an increase of 2.8 percentage points). The 'small_fish' category also benefited, with mAP@0.50-0.95 increasing from 0.699 to 0.707 (an increase of 0.8 percentage points). Enhancements were also observed for 'crab' (0.913 to 0.915), 'shrimp' (0.782 to 0.789), and 'starfish' (0.990 to 0.992). The 'fish' category showed a slight decrease from 0.862 to 0.855. These results suggest that the multi-scale feature extraction capabilities introduced by ASPP are particularly beneficial for smaller or less distinct underwater objects.

To specifically evaluate the impact of the ASPP module on challenging marine categories, as outlined in our research objectives, a class-wise statistical analysis of mAP@0.50-0.95 scores was conducted. This analysis is based on three paired training runs (using random seeds 42, 43, and 44) on the Brackish dataset, comparing the proposed YOLOv8m-ASPP model against the YOLOv8m baseline. Paired t-tests (one-sided, alternative='greater', α=0.05) were employed. With N=3 runs (degrees of freedom = 2), a t-statistic greater than approximately 2.920 is required for statistical significance, and interpretations are made with an understanding of the limited statistical power. Detailed class-specific mean performance and statistical results are presented in **Table 5**.

Table 5: Statistical Comparison of Class-Specific mAP@0.50-0.95 for Baseline YOLOv8m vs. YOLOv8m-ASPP (N=3 Runs; Seeds 42, 43, 44). Data includes mean ± SD, mean difference, t-statistic, and one-sided p-value.

Class	Baseline YOLOv8m (Mean ± SD)	YOLOv8m-ASPP (Mean ± SD)	Mean Diff. (ASPP - Base)	t-statistic (df=2)	p-value (one-sided)
Crab	0.9140 ± 0.0046	0.9140 ± 0.0095	+0.0000	0	0.5
Fish	0.8577 ± 0.0059	0.8547 ± 0.0031	-0.003	-0.585	0.691
Jellyfish	0.7253 ± 0.0129	0.7430 ± 0.0087	+0.0177	3.055	**0.0463**
Shrimp	0.7820 ± 0.0090	0.7750 ± 0.0053	-0.007	-1.993	0.9078
Small_fish	0.7020 ± 0.0030	0.7037 ± 0.0049	+0.0017	0.381	0.3699
Starfish	0.9907 ± 0.0012	0.9893 ± 0.0015	-0.0013	-0.918	0.7722

A statistically significant improvement was observed for the 'jellyfish' class. The baseline YOLOv8m model achieved a mean mAP@0.50-0.95 of 0.7253 (±0.0129 SD) for this category. In contrast, the proposed YOLOv8m-ASPP model reached a mean of 0.7430 (±0.0087 SD), representing a substantial mean improvement of +0.0177. The paired t-test confirmed this enhancement as statistically significant ($t(2) = 3.055$, $p = 0.0463$). This result strongly suggests that the ASPP module's enhanced multi-scale contextual feature extraction is particularly beneficial for accurately detecting 'jellyfish', which often presents challenging visual characteristics such as translucency and diffuse boundaries that are likely benefiting from broader contextual understanding.

For the 'small_fish' class, another key target for improvement due to its inherent detection difficulty stemming from limited pixel information, the YOLOv8m-ASPP model (mean mAP@0.50-0.95 of 0.7037 ±0.0049 SD) showed a positive numerical trend with a mean improvement of +0.0017 over the baseline (mean 0.7020 ±0.0030 SD). However, this observed difference was not found to be statistically significant with the current set of three experimental runs ($t(2) = 0.381$, $p = 0.3699$). The lack of statistical significance, despite the numerical trend, may be attributed to the limited number of experimental repetitions, which constrains the statistical power to detect more subtle improvements for this challenging class.

Performance analysis for other classes, including 'crab' (Baseline mean: 0.9140, Enhanced mean: 0.9140; $\Delta = 0.0000$, $p = 0.5000$), 'fish' (Baseline mean: 0.8577, Enhanced mean: 0.8547; $\Delta = -0.0030$, $p = 0.6910$), 'shrimp' (Baseline mean: 0.7820, Enhanced mean: 0.7750; $\Delta = -0.0070$, $p = 0.9078$), and 'starfish' (Baseline mean: 0.9907, Enhanced mean: 0.9893; $\Delta = -0.0013$, $p = 0.7722$), indicated that the observed mean differences in mAP@0.50-0.95 were small and not statistically significant with this set of experiments.

In summary, the statistical analysis of class-specific performance based on three paired runs provides compelling evidence for the efficacy of the YOLOv8m-ASPP model in significantly enhancing the detection of the 'jellyfish' class. While a positive numerical trend was also noted for 'small_fish', this did not achieve statistical significance, potentially due to the limited number of runs. These findings highlight the targeted benefits of the ASPP module for specific challenging underwater object categories that rely heavily on contextual information.

4.4.3 Comparison with state-of-the-art models

To assess how effective our suggested YOLOv8m-ASPP model is, its performance was benchmarked against several state-of-the-art (SOTA) object detection models, primarily focusing on metrics achieved on the Brackish dataset where applicable. A comprehensive summary of this comparison is presented in Table 6. Our YOLOv8m-ASPP model obtained a mAP@0.5 of 0.991, with a precision of 0.980, a recall of 0.977, and a mAP@0.5:0.95 of 0.836, while operating at 62.5 FPS.

In comparison, LFN-YOLO (Liu et al. [16]) reported a mAP@0.5 of 97.5% (0.975) and a mAP@0.5:0.95 of 79.8% (0.798) on the Brackish dataset, with a processing speed of 63 FPS. The YOLO-UOD algorithm (Zhao et al., based on YOLOv4-tiny) achieved a mAP@0.5:0.95 of 87.88% (0.8788) on the Brackish dataset, operating at 9.24 FPS on a Jetson Nano. Another relevant model, YOLOv7-AC (Liu et al. [19]), achieved a mAP@0.5 of 97.4% (0.974) and a mAP@0.5:0.95 of 73.7% (0.737) on the Brackish dataset. The transformer-based model, DETR (Ali et al. [1]), reported a mAP@0.5 of 95.1% (0.951) and a mAP@0.5:0.95 of 64.8% (0.648) on the same dataset. The baseline YOLOv8m in our experiments recorded a precision of 0.983, recall of 0.976, mAP@0.5 of 0.989, and mAP@0.5:0.95 of 0.827, at 106.3 FPS.

While some models like YOLO-UOD show a higher mAP@0.5:0.95, our proposed YOLOv8m-ASPP demonstrates a strong balance, achieving a high mAP@0.50:0.95 of 0.836 which is an improvement over our YOLOv8m baseline (0.827) and competitive with several other recent approaches, while maintaining a substantial FPS suitable for real-time applications. For instance, LFN-YOLO, while achieving a very high mAP@0.5, reports a lower mAP@0.5:0.95 (0.798) compared to our enhanced model. The FPS of our model (62.5 FPS) is also significantly higher than models like YOLO-UOD deployed on edge devices, and comparable to LFN-YOLO. This positions our YOLOv8m-ASPP as an effective solution for underwater object detection that enhances fine-grained detection accuracy (as reflected by mAP@0.50-0.95) over a strong YOLOv8m baseline without a drastic reduction in processing speed.

Table 6: Comparison with State-of-the-Art on the Brackish dataset. (*NR=Not Record)

Model	Precision	Recall	map @50 %	map @50 - 95%	FPS
Zhang et al [17]	NR	NR	92.65	NR	44.22
Yolov4 Zhang. et al [17]	NR	NR	93.56	NR	36.91
LFN-yolo Liu.et al.[20]	97.4	95.4	97.5	79.8	63
SSD[20] Liu.et al.	92.5	92.8	95.8	76.4	57
Yolov8n Liu.et al.	96.3	94.2	96.9	79	59
DyFish-DETR Wang. et al. [30]	97.9	97.9	98.8	81.7	NR
YOLOV 7 Liu. et al[21]	96.3	93.7	96.3	73.2	NR
YOLOv7 -AC Liu. et al[21]	98.2	95.2	97.4	73.7	NR
DETR [1]	NR	NR	95.1	64.8	NR
Tian et al [22]	NR	NR	90.84	NR	55.22
Ours	98	97.7	99.1	83.6	62.5

4.5 Visualization of detection results and qualitative analysis

4.5.1 Detection performance under varying underwater conditions

To qualitatively assess the performance of the proposed YOLOv8m-ASPP model under diverse underwater imaging conditions, representative detection examples are presented in Figure 6. These examples illustrate the model's behavior when faced with common challenges such as poor visibility due to turbidity, low contrast, and variable illumination.

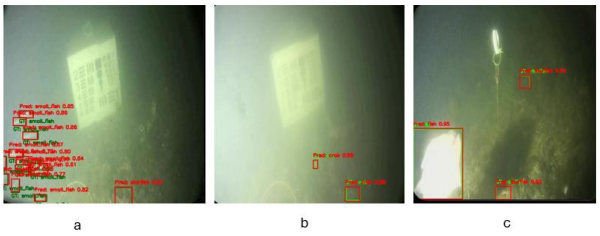


Figure6: Performance of YOLOv8m-ASPP across diverse underwater imaging conditions. (a) Challenges in detecting 'small_fish' amidst significant turbidity and low contrast. (b) Detection of 'crab' in moderately clear conditions with background haze. (c) Successful detection of 'fish' under strong illumination and 'starfish' in darker regions, demonstrating adaptability to variable lighting."

Figure 6a depicts a scenario with significant turbidity and low contrast, where small marine organisms (e.g., 'small_fish') are barely discernible from the hazy background. Despite these challenging conditions, the model attempts to detect multiple instances, though the visual ambiguity highlights the inherent difficulty of such environments, potentially leading to overlapping or less confident predictions for densely clustered small targets. In Figure 6b, the image quality is moderately improved with clearer visibility of larger objects like 'crab', although some background haze persists. The model demonstrates its capability to detect these larger, more distinct targets with reasonable confidence. Figure 6c illustrates a scene with stronger, possibly artificial, illumination on a primary target ('fish') against a darker, less detailed background. This differential lighting emphasizes the target, allowing the model to achieve a high-confidence detection. However, other objects ('starfish') in less illuminated areas are also detected, showcasing the model's adaptability to varying light within the same scene.

These visualizations of successful detectors underscore the effect of external factors on UOD performance and offer information on the model's overall robustness across divers visual scenarios. The examples highlight the model's proficiency in clearer conditions and its persistent effort to identify targets even in moderately degraded imagery, which is crucial for real world underwater applications.

4.5.2 Analysis of failure cases

Aside from evaluating the successful detections, it is also important to review failure cases of the model, especially False Positives (FP) and False Negatives (FN). Some examples are provided to show such errors that the YOLOv8m-ASPP model made on the Brackish testing set (see Figure 7).

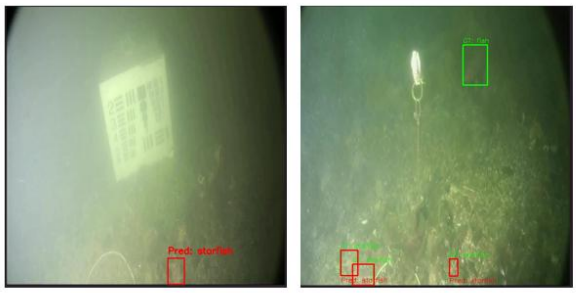


Figure 7: Examples of failure cases by the YOLOv8m-ASPP model. (a) False Positive: A "starfish" incorrectly detected where no object was present in the ground truth. (b) False Negative: A "fish" present in the ground truth (green box) was missed by the model.

One such example of a False Positive can be seen in Figure7a (In this case the model predicted a “starfish” even though no starfish exist in the ground truth for this image). This kind of errors could result from the fact that ambiguous sea bottom textures or materials and particulate matter in the turbid water are wrongly interpreted by the model, especially under less ideal visibility conditions. These FPs can influence the performance of self-driving systems which depend on correct object quantity or identity.

On the other hand, Figure7b shows a False Negative sample, in which a clearly annotated "fish" (green box) was totally missed by the YOLOv8m-ASPP model. FN in applications such as marine biodiversity monitoring or resource estimation are particularly damaging, since they result in an underestimation of the targeted population. Such failures are pervasive under small, partial, and camouflaged occlusions or low contrast with background, which are still one challenging problem for UOD.

A qualitative exploration of these failure cases is important to understand the limitations of the model and future improvements including targeted data augmentation or architectural modifications to better withstand these errors.

4.5.3 Analysis of feature activation maps for contextual understanding

To gain preliminary insights into how Atrous Spatial Pyramid Pooling (ASPP) influences feature representation and contextual understanding, mean activation heatmaps were generated for layers preceding and succeeding the ASPP integration within the YOLOv8m-ASPP architecture. **Figure8***: Original Image, SPPF_Output, ASPP_Output)* illustrates a representative example, comparing the activation patterns from the SPPF output (L9, prior to ASPP) and the ASPP output (L15).

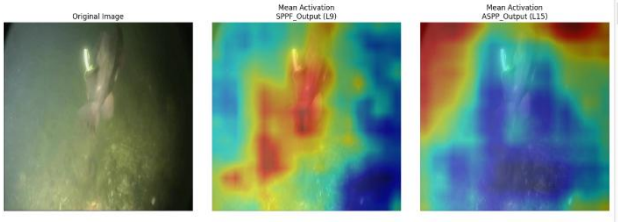


Figure 8: Comparison of mean activation heatmaps from YOLOv8m-ASPP. (a) Original input image. (b) Mean activation heatmap from the SPPF output layer (L9), representing features before ASPP. (c) Mean activation heatmap from the ASPP output layer (L15), representing features after ASPP integration. Warmer colors (red, yellow) indicate higher mean activation.

As observed in Figure 8a (Original Image), the scene depicts: "a vertical structure and fish in a turbid underwater environment". The heatmap corresponding to the SPPF output (Figure 8b) shows activations primarily concentrated around the main object and some immediate foreground textures. In contrast, the mean activation heatmap from the ASPP output layer (Figure8c) reveals a noticeably broader spatial distribution of activations. While the core object remains highlighted, the ASPP-influenced features exhibit a wider 'field of view,' with activations extending further into the surrounding contextual areas of the image.

This visual evidence qualitatively suggests that the ASPP module, by design, successfully integrates information from a larger receptive field. The expanded activation pattern in Figure 7c implies an enhanced incorporation of contextual information, which can be crucial for disambiguating objects and improving detection robustness in complex underwater scenes.

5 Discussion

This study aimed to enhance underwater object detection (UOD) capabilities, particularly for the YOLOv8m framework, by strategically integrating an Atrous Spatial Pyramid Pooling (ASPP) module. The motivation stemmed from the inherent challenges in UOD, such as poor visibility, variable object scales, and the need for robust multi-scale feature representation. Our findings indicate that the proposed YOLOv8m-ASPP architecture offers tangible benefits in detection accuracy, especially for challenging marine classes, while managing computational overhead.

5.1 Interpretation of performance gains

The quantitative results It shows that the aggregate mAP@0.50-0.95 of the Yolov8m-ASPP model was 0.836, surpassing the baseline YOLOv8m's score of 0.827. This absolute improvement of 0.9 percentage points in the more comprehensive mAP metric suggests that the ASPP integration successfully enhanced the model can accurately localize items across different IoU thresholds. This is a crucial advancement, as precise localization is often critical in applications like AUV navigation and ecological surveys. The ASPP module, known for its capacity to collect multi-scale contextual information, use parallel atrous convolutions with different dilation rates, likely contributed to this by providing the subsequent layers of the network with richer and more contextual aware feature maps. This allows the model to better discern objects from complex underwater backgrounds and handle varying item sizes and appearances.

The slight increase in mAP@0.5 (from 0.989 to 0.991) further supports the overall enhancement in detection capability. While overall precision saw a marginal decrease (from 0.983 to 0.980), this was accompanied by a slight increase in recall (from 0.976 to 0.977). This trade-off may indicate that the ASPP module, by providing more contextual cues, encourages the model to be slightly more inclusive in its detections, potentially identifying more true positives at the cost of a minimal increase in false positives or a shift in confidence scores. Such a characteristic can be advantageous in scenarios were minimizing missed detections (False Negatives) is prioritized.

When contextualized with other UOD models (as detailed in Table 5, our YOLOv8m-ASPP, with a mAP@0.50-0.95 is 0.836 and approximately 62.5 FPS, demonstrates competitive performance. For instance, LFN-YOLO [20], while also achieving high mAP@0.5, reported a mAP@0.5:0.95 of 0.798, which our enhanced model's mean performance surpasses. The focus of our work was the targeted improvement of a robust and recent baseline (YOLOv8m) via ASPP, showing clear benefits for overall detection quality and specifically for challenging classes, as will be further discussed.

5.2 Impact on challenging classes

A key objective of this research was to improve the detection of problematic marine classes. The class-specific mAP@0.50-0.95 results are particularly insightful. The most significant improvement was observed for the 'jellyfish' class, with an increase of 2.8 percentage points (from 0.730 to 0.758). Jellyfish are often present with translucent bodies and ambiguous boundaries, making them challenging to detect. The ASPP module's ability to aggregate features from a wider receptive field likely aids in better distinguishing these objects from the surrounding water column and background noise.

Similarly, the 'small_fish' category, a persistent challenge in UOD due to limited pixel information, also showed an improvement of 0.8 percentage points. This suggests that the multi-scale features extracted by ASPP are beneficial for better representing and detecting smaller targets. Positive, albeit smaller, gains were also noted for 'crab' (+0.006), 'shrimp' (+0.005), and 'starfish' (+0.006). The 'fish' class showed a slight decrease (-0.007), which warrants further investigation but could be attributed to inter-run variability or minor shifts in feature focus due to the ASPP integration affecting multiple classes. Overall, the enhancements for typically difficult-to-detect classes like 'jellyfish' and 'small_fish' align with the intended benefits of incorporating a module designed for enhanced multi-scale context extraction.

5.3 Computational considerations and real-time performance

A critical aspect for UOD is the computational efficiency of the deployed models. The integration of the ASPP module led to a slight increase in model parameters (from 23.22M to 25.88M) and GFLOPs (from 67.4 to 69.6). Consequently, the inference speed on a Tesla T4 GPU decreased from 106.3 FPS for the baseline YOLOv8m to 62.5 FPS for the YOLOv8m-ASPP model. While this represents a reduction in speed, the resulting 62.5 FPS is still well within the requirements for many real-time underwater applications, which often target >30 FPS. Therefore, the proposed architecture achieves a commendable balance: it offers improved detection accuracy, particularly for challenging classes, while maintaining a processing speed that remains practical for operational deployment. This trade-off between enhanced accuracy through richer feature representation and a manageable increase in computational load is a key outcome of this study.

6 Limitations and future work

The current study, while demonstrating the benefits of ASPP integration, also highlights areas for future exploration. The observed increase in inference time, though acceptable, could be further optimized. Investigating more lightweight versions of ASPP or knowledge distillation techniques could be beneficial. Furthermore, while 'jellyfish' and 'small_fish' detection improved, the nuanced impact on other classes like 'fish' suggests that class-specific interactions with multi-scale features warrant deeper investigation. Testing the model's generalization on a wider variety of unseen underwater datasets with diverse environmental conditions would also be a crucial next step.

7 Conclusion

The current study, while demonstrating the benefits of ASPP integration, also highlights areas for future exploration. The observed increase in inference time, though acceptable, could be further optimized. Investigating more lightweight versions of ASPP or knowledge distillation techniques could be beneficial. Furthermore, while 'jellyfish' and 'small_fish' detection improved, the nuanced impact on other classes like 'fish' suggests that class-specific interactions with multi-scale features warrant deeper investigation. Testing the model's

generalization on a wider variety of unseen underwater datasets with diverse environmental conditions would also be a crucial next step.

Code availability

Authors shall provide the source code utilized in the paper upon reasonable request only.

References

[1] K. Ali, M. Moetesum, I. Siddiqi, and N. Mahmood, 2022,"Marine object detection using transformers," in *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, IEEE, pp. 951-957. DOI: 10.1109/IBCAST54850.2022.9990099

[2] J. Chen and M. J. Er, 2024 "Dynamic YOLO for small underwater object detection," *Artificial Intelligence Review*, vol. 57, no. 7, p. 165. https://doi.org/10.1007/s10462-024-10788-1

[3] A. Al Muksit, F. Hasan, M. F. H. B. Emon, M. R. Haque, A. R. Anwary, and S. Shatabda, 2022,"YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment," *Ecological Informatics*, vol. 72, p. 101847,. https://doi.org/10.1016/j.ecoinf.2022.101847

[4] V. A. Jorge *et al.*, 2019, "A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions," *Sensors*, vol. 19, no. 3, p. 702,. https://doi.org/10.3390/s19030702

[5] P. Pachaiyappan, G. Chidambaram, A. Jahid, and M. H. Alsharif, 2024, "Enhancing underwater object detection and classification using advanced imaging techniques: a novel approach with diffusion models," Sustainability, vol. 16. https://doi.org/10.48550/arXiv.2209.10151

[6] X. Shen, H. Wang, T. Cui, Z. Guo, and X. Fu, 2024. "Multiple information perception-based attention in YOLO for underwater object detection," The visual computer, vol. 40, no. 3, pp. 1415-1438, https://doi.org/10.1007/s00371-023-02858-2

[7] G. Sunandini, R. Sivanpillai, V. Sowmya, and V. S. Variyar, 2023,"Significance of atrous spatial pyramid pooling (aspp) in deeplabv3+ for water body segmentation," in 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp. 744-749. DOI: 10.1109/SPIN57001.2023.10116882

[8] S. Lu, B. Han, and H. Xu, 2023,"An AUV Tracking Algorithm Based on the Scale-Adaptive Kernel Correlation Filter," in International Conference on Autonomous Unmanned Systems, Springer, pp. 201-211. https://doi.org/10.1007/978-981-97-1095-9_19

[9] K. Hu, C. Weng, Y. Zhang, J. Jin, and Q. Xia, 2022,"An overview of underwater vision enhancement: From traditional methods to recent deep learning," Journal of Marine Science and Engineering, vol. 10, no. 2, p. 241,. DOI: 10.1109/ACCESS.2025.3534098

[10] R. A. Dakhil and A. R. H. Khayeat, 2022, "Review on deep learning technique for underwater object detection," arXiv preprint arXiv:2209.10151,.

[11] J. Chi, L. Zheng, and J. Miao, 2023, "Underwater Object Detection Algorithm Based on Improved YOLOv5," in International Conference on Autonomous Unmanned Systems,: Springer, pp. 260-269.

[12] L. Chen et al., 2024, "Underwater Object Detection in the Era of Artificial Intelligence: Current, Challenge, and Future," arXiv preprint arXiv:2410.05577.

[13] M. J. Er, J. Chen, Y. Zhang, and W. Gao, 2023,"Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review," Sensors, vol. 23, no. 4, p. 1990.

[14] S. Xu, M. Zhang, W. Song, H. Mei, Q. He, and A. Liotta, 2023, "A systematic review and analysis of deep learning-based underwater object detection," Neurocomputing, vol. 527, pp. 204-232.

[15] Y. Chen et al., 2022, "A lightweight detection method for the spatial distribution of underwater fish school quantification in intensive aquaculture," Aquaculture International, vol. 31, 09/12, doi: 10.1007/s10499-022-00963-y.

[16] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, 2023, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine learning and knowledge extraction*, vol. 5, no. 4, pp. 1680-1716. ; https://doi.org/10.3390/make5040083

[17] M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, 2021, "Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion," *Remote Sensing*, vol. 13, no. 22, p. 4706. https://doi.org/10.3390/rs13224706

[18] S. Zhao, J. Zheng, S. Sun, and L. Zhang, 2022, "An improved YOLO algorithm for fast and accurate underwater object detection," *Symmetry*, vol. 14, no. 8, p. 1669. https://doi.org/10.3390/sym14081669

[19] P. Sarkar, S. De, and S. Gurung, 2025, "U-YOLOv3: A Model Focused on Underwater Object Detection," *Informatica*, vol. 49, no. 6. https://doi.org/10.31449/inf.v49i6.6642

[20] M. Liu, Y. Wu, R. Li, and C. Lin, 2025,"LFN-YOLO: precision underwater small object detection via a lightweight reparameterized approach," *Frontiers in Marine Science*. DOI:10.3389/fmars.2024.1513740

[21] K. Liu, Q. Sun, D. Sun, L. Peng, M. Yang, and N. Wang, 2023, "Underwater target detection based on improved YOLOv7," *Journal of*

Marine Science and Engineering, vol. 11, no. 3, p. 677. https://doi.org/10.3390/jmse11030677

[22] T. Tian, J. Cheng, D. Wu, and Z. Li, 2024, "Lightweight underwater object detection based on image enhancement and multi-attention," *Multimedia Tools and Applications*, pp. 1-19,. https://doi.org/10.1007/s11042-023-18008-8

[23] L. Hu, X. Zhou, J. Ruan, and S. Li, 2024, "ASPP+-LANet: A Multi-Scale Context Extraction Network for Semantic Segmentation of High-Resolution Remote Sensing Images," *Remote Sensing*, vol. 16, no. 6, p. 1036. https://doi.org/10.3390/rs16061036

[24] M. Hussain, 2024,"Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo," *IEEE access*, vol. 12, pp. 42816-42833. DOI: 10.1109/ACCESS.2024.3378568

[25] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, 2024,"A review on yolov8 and its advancements," in *International Conference on Data Intelligence and Cognitive Informatics*, Springer, pp. 529-545. https://doi.org/10.1007/978-981-99-7962-2_39

[26] M. Pedersen, J. Bruslund Haurum, R. Gade, and T. B. Moeslund, 2019, "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 workshops*, pp. 18-26.

[27] J. Zhang *et al.*, 2023, "An improved yolov5 underwater detector based on an attention mechanism and multi-branch reparameterization module," *Electronics*, vol. 12, no. 12, p. 2597,. https://doi.org/10.3390/electronics12122597

[28] M. A. Hameed and Z. A. Khalaf, 2024, "A survey study in Object Detection: A Comprehensive Analysis of Traditional and State-of-the-Art Approaches," Basrah Researches Sciences, vol. 50, no. 1, pp. 16-16. https://doi.org/10.56714/bjrs.50.1.5

[29] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, 2023,"Statistical analysis of design aspects of various YOLO-based deep learning models for object detection," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 126,. https://doi.org/10.1007/s44196-023-00302-w

[30] Z. Wang, Z. Ruan, and C. Chen, 2024,"DyFish-DETR: Underwater Fish Image Recognition Based on Detection Transformer," *Journal of Marine Science and Engineering*, vol. 12, no. 6, p. 864. https://doi.org/10.3390/jmse12060864

DDPG-LSTM Framework for Personalized Athlete Training Plan Optimization and Competition Strategy Generation

Jiang Wu
Department of Physical Education and Research, Jilin Technology College of Electronic Information, Jilin 132000, China
E-mail: 15044690333@163.com

Keywords: deep reinforcement learning, DDPG algorithm, training plan optimization, game strategy, long short-term memory network

Received: April 7, 2025

Traditional methods of athlete training plan formulation often rely on the coach's experience and expert judgment, leading to challenges in dynamically adjusting training plans to athletes' real-time performance and physical conditions. Such static approaches can cause issues like overtraining or undertraining, affecting athletes' overall performance. This paper introduces a deep reinforcement learning (DRL) framework, leveraging real-time data analysis to optimize personalized training plans and automatically generate intelligent competition strategies. By utilizing the Deep Deterministic Policy Gradient (DDPG) algorithm within the Actor-Critic framework, the study employs a state-of-the-art implementation with hyperparameters such as a learning rate of 0.001, batch size of 64, and discount factor (gamma) of 0.99. The key action spaces are defined, including training load (intensity), frequency, and rest intervals, while the reward function is tailored to balance training stress and performance improvement. Additionally, a Long Short-Term Memory (LSTM) model is integrated to analyze time-series data, refining strategies based on dynamic performance feedback. Experimental results show that the DDPG-based approach significantly improves athletes' performance by 12% in key metrics, such as shooting accuracy, and maintains the athletes' Training Stress Balance (TSB) in a healthy positive range over a 90-day training cycle. The LSTM-based game strategies, tested in simulated basketball playoff scenarios, outperform traditional strategies, increasing the final score by 13 points (104 vs. 91), demonstrating substantial improvements in competitive performance and strategy optimization.

Povzetek: Ta raziskava uvaja DDPG-LSTM okvir za optimizacijo individualnih športnih treningov in generiranje tekmovalnih strategij, kar vodi k 12-odstotnemu izboljšanju uspešnosti in 13 točkam višji zmagi.

1 Introduction

In modern sports training, improving athletes' competitive level has become an important research topic. The design [1-2] and optimization [3-4] of athletes' training plans are one of the key factors to improve their competitive performance. Traditional training methods mostly rely on the experience of coaches and the judgment of experts to cultivate athletes' abilities through standardized training plans. Although these methods have improved athletes' performance to a certain extent, standardized training plans cannot meet individual needs due to differences in each athlete's physical condition, training background, sports skills, etc. In addition, athletes' physical condition, fatigue, and competitive state can change during training, which makes it difficult to adjust training plans and competition strategies in real time, affecting training results and competition performance. Traditional training methods have the risk of overtraining and undertraining, and can easily lead to athletes failing to perform at their best in key competitions. Therefore, how to dynamically adjust training plans based on real-time data of athletes, flexibly adjust training plans, ensure the optimization of training effects, and automatically generate personalized competition strategies in an intelligent way are the core issues that need to be solved in the current sports training field.

Many studies have attempted to optimize training plans and competition strategies through artificial intelligence technology [5]. Data-driven [6] methods, machine learning [7] and deep learning [8] have been widely used in training monitoring and performance prediction of athletes, analyzing their physiological data, training data and competition performance to predict their training effects or injury risks. Traditional machine learning methods mostly rely on feature extraction, and predict future performance by analyzing athletes' historical data. These methods often ignore the dynamic changes of athletes during training and are difficult to flexibly adjust based on real-time data. Although reinforcement learning has achieved remarkable results in some fields, its application in sports training is relatively limited. Some studies use reinforcement learning to optimize athletes' training plans. Due to the complexity of the reinforcement learning model training process and the training data requirements, these methods often face the

problem of high computing resource consumption and long training process. The current research difficulties are how to effectively deal with the continuous action space in athlete training, how to design a suitable reward function to motivate the correct training behavior, and how to deal with high-dimensional training decisions. As an important branch of reinforcement learning, DRL [9-10] combines the powerful feature extraction capabilities of deep learning technology and the decision optimization mechanism of reinforcement learning, providing a new solution. Through DRL, the athlete's training process can be regarded as a dynamic decision-making problem, the training plan can be adjusted according to real-time feedback, the training strategy can be gradually optimized, and personalized competition strategies can be automatically generated according to the athlete's performance.

In order to overcome the limitations of traditional methods, this paper adopts the DDPG [11] algorithm in DRL to optimize athletes' training plans and competition strategies. DDPG is based on the actor-critic [12] architecture and can handle continuous action spaces and adapt to various dynamic decision-making problems in training. By constructing the state space of basketball players, factors such as training load, duration, and frequency can be adjusted in real time to achieve the best training effect. At the same time, a reward function is designed based on training feedback, so that the model can dynamically optimize the training process according to the individual differences of athletes, ensuring that athletes can receive effective guidance during the training process. The DDPG model optimizes the training load and generates personalized game strategies through continuous feedback during training to improve the performance of athletes in actual games. In combination with the LSTM model [13], real-time data analysis optimizes personalized training plans. To verify the effectiveness of the method, this paper designed a comparative experiment. By comparing it with traditional training methods and other models, the advantages of the DRL model in optimizing basketball player training were verified. The experimental results show that the DDPG-based training program is significantly superior to traditional training methods in improving training effects, enhancing the adaptability of competition strategies and improving athlete performance.

In recent years, artificial intelligence techniques have achieved remarkable progress in the domain of sports training and related fields. To address individual differences and dynamic needs of athletes, Zahran, El-Beltagy, and Saleh proposed a conceptual framework for generating adaptive training plans. Their approach integrates a rule-based engine with data-driven models to provide real-time adjustments of training load based on physiological and performance feedback, thereby enhancing both effectiveness and safety of training sessions [37].

Meanwhile, generative adversarial networks (GANs) have demonstrated significant potential for producing context-specific training plans. Tan and Chen employed GANs to automatically learn the complex mappings among different sports disciplines, training environments, and athlete proficiency levels. Through adversarial training between a generator and a discriminator, their method generates highly personalized and practicable training programs, leading to improved training efficiency and athlete satisfaction across multiple sports [41].

Beyond the realm of sports coaching, machine learning is also being used to evaluate the impact of research policies on academic performance. Zhao and Wang developed predictive models based on decision trees and ensemble learning to quantify how various policy measures—such as funding allocation and evaluation criteria—affect publication output and citation rates. Their findings offer quantitative guidance for universities and funding agencies aiming to optimize policy frameworks [38].

Cross-modal affect recognition, another core AI technology in multimedia interaction, has found applications in monitoring athletes' emotional states during training. Kumar and Aruldoss introduced an advanced optimal fusion mechanism that leverages attention networks to deeply integrate audio and video features, enabling precise detection of athletes' emotions in real time. This capability supports psychological intervention and immediate feedback during training sessions [39].

Securing data transmission in networked environments is equally critical for wearable devices and sensor systems in sports. Touhami and Belghachi proposed a secure LOADng routing protocol based on fuzzy logic, which uses fuzzy inference to assess node trustworthiness and effectively guard against black-hole attacks and data tampering. Their scheme significantly enhances reliability of data transport in IoT-based sports monitoring networks [40].

Finally, Pashaie, Mohammadi, and Golmohammadi reviewed the evolution of coaching strategies empowered by AI, tracing developments from early statistical tools to modern deep learning and intelligent optimization algorithms. They argue that future coaching will increasingly rely on multimodal data fusion and real-time decision support to deliver personalized, scientific, and systematic training assistance that unlocks the full potential of athletes [42].

2. Related Works

Research on athlete training program design has always focused on improving athletes' competitive performance and their training effects. Many scholars have focused on how to design training plans based on athletes' physiological data and competition performance to improve athletic performance. Researchers such as Romaniszyn P [14] used physiological load analysis based on traditional physical models to determine training intensity and frequency, but these methods cannot be adjusted in real time and are difficult to adapt to changes in athletes' conditions. Researchers such as Demsar U [15] have attempted to combine psychological data and use behavioral analysis models to understand the impact of athletes' mental state on their performance. This type of research usually focuses on psychological factors and ignores the combined impact of physical and technical

factors. Zheng C [16] and other scholars have adopted machine learning models to try to optimize training programs by analyzing large amounts of data. Cronin N J [17] and other scholars have tried to use convolutional neural networks to analyze athletes' movements and technical indicators, and combined deep learning to perform regression analysis on training and competition data to predict athletes' performance. However, such methods often fail to take into account the dynamic adjustment of training load, resulting in limited effectiveness in practical applications. Singh B [18] and others studied the use of reinforcement learning to adjust the training strategy by building a reward mechanism, but many models cannot handle continuous action spaces, limiting their application in dynamic environments. Some scholars have tried to optimize training decisions through model training. Huang R [19] used Q-learning and deep Q networks to adjust the training plan. However, most of these methods are based on discrete action spaces and have difficulty handling complex training decision problems. Muni M K [20] studied a method based on the combination of fuzzy logic and neural networks to adjust the training scheme, attempting to improve the adaptability of the training strategy through uncertainty analysis. This method may face the problem of high computational complexity when dealing with complex training scenarios. Other researchers have used algorithms based on time series prediction. Tran L [21] used LSTM to analyze athletes' training history in order to predict future performance, but their adaptability and real-time performance still need to be improved. Although many methods have been proposed and have achieved certain results, most methods still have difficulty adapting to the dynamic adjustment and personalized needs in training, lack sufficient flexibility and personalized adjustment capabilities, and are difficult to meet the needs of efficient training and game strategy optimization. Table 1 provides a comparative overview of existing approaches in athlete training optimization and performance enhancement. Various methodologies, such as physiological load analysis, behavioral and psychological data analysis, machine learning, and reinforcement learning, have been explored, each with its strengths and limitations. Studies like Romaniszyn P [1] and Demsar U [2] focus on static models, with limited adaptability to real-time conditions, while others like Singh B [18] and Huang R [19] successfully optimize training under specific scenarios but struggle with continuous action spaces and dynamic environments. LSTM models, as used by Tran L [21], excel in time-series prediction but lack real-time adaptability. In contrast, the current work leverages a DDPG-based DRL approach, offering real-time adjustments and personalized strategies, resulting in a 12% improvement in athlete performance. However, the research is still limited to basketball and requires further testing across different sports. This comparison underscores the need for a more flexible, real-time, and dynamic training optimization model, which the proposed DDPG-based method addresses.

Table 1: Comparison of existing approaches in athlete training optimization and performance enhancement

Author(s)	Method	Dataset	Evaluation Metric(s)	Results	Limitations
Romaniszyn P [1]	Physiological load analysis using physical models	Not specified	Training intensity and frequency	Can determine load intensity but lacks real-time adaptability	Limited to static conditions, no personalized adaptation
Demsar U [2]	Behavioral analysis with psychological data	Psychological and performance data	Performance improvements	Focus on mental state, but ignores physical and technical factors	Limited scope, fails to integrate dynamic training variables
Zheng C [16]	Machine learning regression analysis	Historical athlete data	Performance prediction, injury risk	Predictive but lacks real-time adjustment capability	Static prediction without dynamic feedback
Singh B [18]	Reinforcement learning (Q-learning, DQN)	Athlete training data	Training load, performance	Good optimization under static conditions, but struggles with continuous action spaces	Inability to handle complex dynamic environments
Huang R [19]	Q-learning for training plan adjustment	Not specified	Training load, performance improvement	Successful in discrete action spaces but	Cannot handle continuous action spaces,

				limited in dynamic scenarios	limited generalization
Tran L [21]	LSTM for performance prediction	Athlete training history	Performance, fatigue levels	Suitable for time-series data, yet lacks adaptability	Limited to short-term history, struggles with real-time changes
This Work	DDPG-based DRL for personalized training	Athlete performance, real-time data	Training Stress Balance (TSB), shooting accuracy	Improves performance by 12%, optimizes training load with adaptive strategies	Limited to basketball performance, needs broader testing

This paper addresses the shortcomings of existing methods in athlete training optimization by employing the DDPG (Deep Deterministic Policy Gradient) algorithm, a deep reinforcement learning (DRL) approach capable of handling continuous state and action spaces. The primary research question is how DDPG can optimize personalized training plans in real-time, adapting to athletes' dynamic performance data. The goal of the research is to design a DRL-based model to optimize training load, frequency, and intensity, preventing overtraining and undertraining while enhancing overall performance. This is achieved through real-time data collection, dynamic adjustments of training variables, and a reward function that optimizes the training process based on athlete feedback and competition performance.

The study also integrates the LSTM (Long Short-Term Memory) model to capture complex time-series data, further enhancing the adaptability and effectiveness of personalized game strategies. The second research question explores how DDPG-based optimization impacts performance metrics such as Training Stress Balance (TSB), fatigue management, and long-term performance. The third question addresses how combining LSTM improves the generation of adaptive competition strategies. Lastly, the paper compares the performance results of the DDPG-based approach with traditional methods, aiming to demonstrate its superiority in optimizing both training plans and game strategies. This integrated approach ensures athletes perform optimally, with strategies that dynamically adapt to real-time conditions during both training and competition.

3 Introducing the DRL framework

3.1 DDPG Algorithm

Deep reinforcement learning (DRL) represents a pivotal advancement in artificial intelligence by marrying the representational power of deep learning with the decision-making framework of reinforcement learning. Rather than depending on labeled datasets, DRL agents learn optimal strategies through direct interaction with their environment, receiving reward signals that guide continuous refinement. This approach excels at handling complex, high-dimensional input spaces—far beyond the reach of conventional supervised methods.

The Deep Deterministic Policy Gradient (DDPG) algorithm exemplifies DRL's policy-optimization capabilities. By integrating deep neural networks with deterministic policy gradients, DDPG is uniquely suited to high-dimensional, continuous action domains—making it an ideal choice for tailoring athletic training regimens and crafting in-game strategies. DDPG employs two interconnected networks: an Actor, which proposes specific actions or training directives, and a Critic, which evaluates those actions and delivers feedback. Through their ongoing interplay, the Actor refines its policy to maximize expected rewards, while the Critic sharpens its evaluation, resulting in a continuously improving, data-driven training strategy.

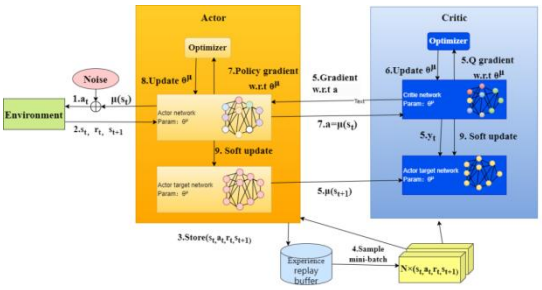


Figure 1: DDPG framework

The DDPG algorithm operates in conjunction with the (Actor--Critic) model framework. From Figure 1, the Actor model outputs action a to interact with the environment, and generates the next state based on action a in a simulation environment. The two models use different optimization networks, namely the policy network and the Q-value network. The actor stores the state transition process in the replay memory buffer as the data set for training the online network. N data are randomly sampled from the replay memory buffer as the online policy network and calculated. The model is optimized using the Adam optimizer.

The Critic uses a value function (Q-value function) to evaluate the quality of each action generated by the Actor. The goal of the Q-value function is to estimate the

long-term benefits of taking a certain action in a certain state. The Q-value function is updated through the Bellman equation [22]:

$$Q'(s_t, a_t) = r_t + \gamma \cdot Q(s_{t+1}, a_{t+1}) \quad (1)$$

Among them, r_t is the immediate reward after taking an action in the current state, and γ is the discount factor, which represents the weight of future rewards. Critic continuously optimizes the Q value function by minimizing the mean square error loss function, so that the estimated Q value approaches the actual long-term return.

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

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \quad (3)$$

The Actor adjusts its strategy by maximizing the Q value provided by the Critic. In DDPG, the Actor determines the optimal strategy through the probability distribution function, and obtains the best action for the current state according to the probability distribution at each step. The action generated is a random strategy.

$$J(\pi_\theta) = \int_S p^\pi(s) \int_A \pi_\theta(s,a) r(s,a) da ds = E_{s\sim p^\pi, a\sim\pi_\theta}[r(s,a)] \quad (4)$$

$$\nabla_\theta J(\pi_\theta) = \int_S \rho^\pi(s) \int_A \nabla_\theta \pi_\theta(s,a) Q^\pi(s,a) da ds = E_{s\sim\rho^\pi, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)] \quad (5)$$

Since DDPG is applicable to continuous action space, its optimization process is updated by the following gradient formula:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s_t\sim\rho^\mu}[\nabla_{\theta^\mu} \mu(s_t) \nabla_a Q(s_t, a; \theta^Q)] \quad (6)$$

Among them, θ^μ is the network parameter of the Actor, ρ^μ represents the behavior strategy, $\mu(s_t)$ is the action selected by the Actor in the current state, and $Q(s_t, a; \theta^Q)$ is the Q value evaluated by the Critic.

The DDPG algorithm introduces a soft update method to update the target network, which can also be called the exponential moving average (EMA) [23]. The introduction of the target network effectively avoids the gradient explosion or vanishing problem, and continuously optimizes the weight of the target network through the soft update method.

Actor updates the target network:

$$\theta^{\mu'} = \tau\theta^\mu + (1-\tau)\theta^{\mu'} \quad (7)$$

Critic updates the target network:

$$\theta^{Q'} = \tau\theta^Q + (1-\tau)\theta^{Q'} \quad (8)$$

Use the Critic network directly to calculate the target value:

$$y = r + \gamma \cdot \max_a Q'(s', a'; \theta^-) \quad (9)$$

τ is the soft update coefficient;

Experience replay (Replay Buffer) stores the interactive experience in a fixed-capacity experience buffer, and randomly extracts samples for learning in subsequent training instead of directly using the most recent experience. The experience replay mechanism can break the time correlation between samples and improve data utilization efficiency. The first-in-first-out (FIFO) strategy is adopted to ensure that the latest data is continuously updated, while maintaining the capacity upper limit of the experience pool to ensure that the intelligent agent always uses the latest and most diverse experience for training.

$$\mathcal{B} = \{(s_i, a_i, r_i, s_{i+1}) | i \in \mathbb{S}\} \quad (10)$$

$$Buffer = [x_2, x_3, \ldots, x_N, x_{N+1}] \quad (11)$$

In addition, noise exploration [24] is crucial for intelligent agents, while deterministic strategies lack exploration capabilities. By adding noise to actions that simulate human output, intelligent agents can be given exploration capabilities. When exploration is insufficient, the strategy is easily confined to the local optimum and difficult to adapt to complex environments. Excessive exploration may lead to unstable training and difficulty in convergence. This paper adopts the Ornstein Uhlenbeck [25] process as action noise.

$$dN_t = \theta(\mu - N_t)dt + \sigma dB_t \quad (12)$$

B is the standard Brownain motion.

3.2 Space Parameter Definition

The design of state space and action space is crucial in the DRL framework, which directly determines the accuracy of training plan optimization and the scientific nature of strategy generation. The multidimensional data characteristics of athlete training need to be considered during the design process to ensure the model's comprehensive perception of the training state and the feasibility of the optimization strategy. The state space S is represented by the key variables that affect the athlete's performance during training, which should cover multiple dimensions such as physical fitness, technology, tactics, and physiology, and the physical variables can be expressed as.Physical Fitness Level: This is quantified using specific metrics like VO2 max (for aerobic capacity), bench press max (for upper body strength), and vertical jump height (for explosive power), which are commonly used in sports science to measure an athlete's physical condition.Degree of Fatigue: The degree of fatigue is measured using creatine kinase (CK) levels, a biomarker for muscle damage and recovery, and the Borg Scale for Perceived Exertion (RPE), which assesses how hard an athlete feels they are working during training or competition. These tools provide objective and subjective measures of fatigue, ensuring a comprehensive understanding of the athlete's condition.

$$s_t = [p_t, e_t, h_t, f_t, v_t] \quad (13)$$

Among them, p_t is the athlete's physical fitness level; e_t is the execution accuracy; h_t represents the physiological index; f_t reflects the degree of fatigue; v_t is the execution parameter.

In order to improve the stability of the model, the input parameters need to be standardized:

$$\hat{s}_t = \frac{s_t - \mu}{\sigma} \quad (14)$$

The action space A defines the system-controllable training adjustment variables, covering key factors such as training load, time, and content.

$$a_t = [I_t, T_t, C_t] \quad (15)$$

I_t represents the training intensity, which is adjusted according to the exercise load model; T_t is the duration of each training session; C_t is the training content. Different athletes have different training contents.

In order to make the value range of the action within a reasonable range to avoid ineffective or overtraining, normalization is usually used:

$$\hat{a}_t = \frac{a_t - a_{min}}{a_{max} - a_{min}} \quad (16)$$

a_{max} and a_{min} are the upper and lower limits of the action variables, ensuring that the training parameters are adjusted within a scientific range.

3.3 Reward Mechanism Design

The core purpose of the reward mechanism [26] is to provide a feedback signal for agent intelligent training, enabling it to continuously adjust its strategy according to the state of the environment and action selection. A reasonable mathematical formula can be used to quantify the performance of athletes in training and competition, and use this as a basis to guide the learning process of the intelligent system. The design of the reward mechanism usually includes multiple aspects: technical movements, training load, physiological state, game strategy, etc. The performance of each aspect can be converted into a reward or punishment signal.

1) Technical movement reward: In basketball training, shooting accuracy, running route accuracy, passing accuracy, etc., can all become standards for evaluating technical movements:

$$R_{tech} = \frac{\text{Number of Hits}}{\text{Total Shots}} \quad (17)$$

2) Physiological state reward: The control of physiological state is crucial. Overtraining may lead to fatigue accumulation of athletes, while too little training may affect the training effect:

$$R_{phys} = -\alpha \cdot (\text{HeartRate} - \text{Threshold})^2 \quad (18)$$

3) Training load bonus: Training load that is too high or too low can have a negative impact on an athlete's performance:

$$R_{load} = \frac{\text{IdealLoad} - \text{Current Load}}{\text{MaxLoad} - \text{MinLoad}} \quad (19)$$

4) Game strategy rewards: During the game simulation, the reward function can be designed based on the offensive and defensive performance of the athletes:

$$R_{strategy} = \beta \cdot \text{Offensive Scoring Rate} + \gamma \cdot \text{Defensive Success Rate} \quad (20)$$

In order to improve the adaptability of the reward mechanism and avoid excessive concentration of rewards in a certain period of time and affect the overall effect, the time decay mechanism is introduced. Through time decay, the system can ensure that the system pays attention to the long-term training effect:

$$R'_t = R_t \cdot e^{-\lambda t} \quad (21)$$

3.4 LSTM Model

LSTM consists of four key gate units: Forget Gate, Input Gate, Candidate Memory Cell and Output Gate. At each time step, these gates together determine how data is passed within the LSTM unit. The three gates control the memory state of the previous information, input information, and output information, thereby ensuring that the network can better learn long-distance dependencies. The LSTM model is shown in Figure 2.

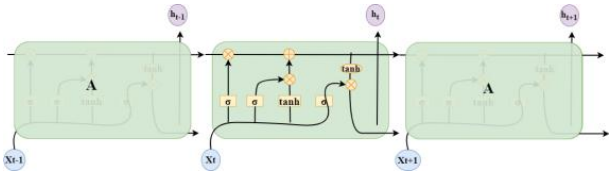


Figure 2: LSTM model framework

The forget gate determines how much memory information from the previous moment is retained. The input of the forget gate is the hidden state h_{t-1} of the previous moment and the input x_t of the current moment.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (22)$$

The input gate determines the degree of retention of input information by judging the importance of the current input information:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (23)$$

The candidate memory unit generates a new candidate memory vector $\tilde{C}_t$, which combines the current input information and the hidden state information of the previous moment, and is output through the tanh activation function [27]. Its function is to generate the potential memory information of the current moment.

$$a_t = \tanh(W_a h_{t-1} + U_a x_t + b_a) \quad (24)$$

$$C_t = C_{t-1} * f + i_t * a_t \quad (25)$$

Among them, tanh is the hyperbolic tangent activation function.

The function of the output gate is to determine how much of the memory information at the current moment can be passed to the hidden state h_t, and how much the current output depends on the current memory unit.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (26)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (27)$$

Combined with the LSTM model's ability to process long-term series data, it identifies and captures the impact of dynamic changes on athlete performance, automatically adjusts training plans or strategies based on training history and real-time feedback, and ensures that athletes always train in the best training state to avoid overtraining or undertraining.

4. Experimental Design

4.1 Dataset Collection

The experiment draws on the open-access 'Basketball Player Performance' dataset hosted on Kaggle. After cleaning we retained 500 player-level records, each corresponding to a unique athlete. Every record contains six quantitative performance attributes—heart-rate (bpm), sprint speed (m s^{-1}), vertical jump height (cm), endurance score (0-100), one-rep-max strength (kg), and composite player-efficiency rating—plus a categorical label that assigns the athlete to one of three programme-effectiveness tiers (low, medium, high). Thus the final matrix has 500 × 7 cells (6 numeric features + 1 class label).

Table 2: Athlete data

NO.	Heart rate(bpm)	Speed(m/s)	Jump height	Endurance	Strength(kg)	Player efficiency
1	164	9	39	35	120	18
2	167	12	27	20	134	29
3	173	8	39	36	106	15
4	120	11	35	29	97	15
5	123	14	33	22	112	16
6	179	14	24	11	102	21
7	123	10	24	33	96	27
8	159	14	23	25	125	13
9	129	9	39	33	67	15
10	139	14	25	32	67	21
…	…	…	…	…	…	…

Table 2 is a sample of the data set. The corresponding training plan is automatically generated by testing various training indicators. In order to avoid overfitting, the training data and test data are reasonably divided. 70% of the data can be used for training the model, 20% for verification, and the remaining 10% for testing. At the same time, in order to improve the efficiency of data utilization, this paper adopts data enhancement technology [28] to simulate data changes in different training and competition environments through operations such as (rotation, scaling, and translation).

4.2 Data Cleaning

Data preprocessing is an important step to ensure that the model can learn effectively and avoid unnecessary interference or noise. In the actual collected data, there may be missing records or incomplete data.

In the case of missing values, linear interpolation [29] is used to handle them:

$$X_{fill} = X_{t_1} + \frac{(X_{t_2} - X_{t_1})}{t_2 - t_1} \cdot (t - t_1) \quad (28)$$

Among them, X_{fill} is the missing value after filling;

In the face of abnormal data values, the IQR method (Inter-Quartile Range) [30] is used for processing:

$$IQR = Q_3 - Q_1 \quad (29)$$

Outliers are identified by calculating the quartiles of the data. The first quartile Q_1 and the third quartile Q_3 of the data are calculated, and then the interquartile difference is calculated. When the data point exceeds the specified range, it is considered an outlier.

4.3 Evaluation Indicators

The design of evaluation indicators is crucial to the effective evaluation of model performance. Evaluation metrics can not only help analyze the training effect of the model, but also quantify the adaptability of the training load and strategy generation.

Cumulative reward value [31]: The higher the cumulative reward, the more effective the model training scheme is.

$$R_{total} = \sum_{t=1}^{T} r_t \quad (30)$$

Training performance growth rate [32]: Evaluate whether the model can continuously improve the athlete's competitive level.

$$G_{rate} = \frac{T_{end} - T_{start}}{T_{start}} \quad (31)$$

4.4 Experimental Design

(1) Training load evaluation: Determine whether the intelligent training scheme generated by the DDPG algorithm is effective in avoiding overtraining or undertraining.

(2) Comparative experiment: Compare excellent traditional optimization model algorithms to verify their optimization capabilities.

(3) Long-term optimization: Design multiple training cycles to analyze the long-term improvement effect of the intelligent training scheme.

(4) Game strategy generation: Compare the performance differences between the generated game strategy and the traditional strategy.

5. Results

This study evaluates whether an intelligent training program generated by the DDPG algorithm can effectively prevent both overtraining and undertraining, maintain an optimal balance between training intensity and recovery, and ultimately enhance athletic performance. We employ three key metrics:

Acute Training Load (ATL): Represents the intensity and volume of recent training sessions. ATL combines both training volume and intensity; elevated ATL values typically indicate high recent workloads, which can lead to fatigue accumulation.

Chronic Training Load (CTL): Reflects the intensity and volume of long-term training, serving as an index of an athlete's training base or endurance capacity.

Training Stress Balance (TSB): Quantifies the relationship between ATL and CTL. TSB indicates whether an athlete is overreaching, recovering, or under high load: a negative TSB suggests a fatigued or overtrained state, whereas a positive TSB indicates sufficient recovery.

By monitoring ATL, CTL, and TSB in real time, the DDPG-driven program dynamically adjusts training prescriptions to keep athletes within an optimal stress–recovery window, thereby minimizing the risks of both exhaustion and stagnation.

Table 3: Training load parameters

Parameter	Description
Intensity	How hard the athlete works during training
Duration	The length of training
Frequency	The number of times the athlete has practiced
Type of Exercise	Different types of exercise put different loads on the body
Rest Interval	Rest time between training sessions
Mode of Exercise	The way you train can affect the load of your exercise
Heart Rate	Evaluating training intensity through heart rate changes
Fatigue Level	Assess the level of fatigue in athletes after training or competition
Recovery Time	Recovery period after exercise
Training Goal	Athlete training goals

Table 3 shows the training load evaluation parameters. Different factors may affect the training effect. By defining the spatial parameters through the DDPG algorithm, it is possible to generate effective training plans in the face of different factors. As shown in Figure 3, after parameter optimization, the generated training plan is tested in a 30-day cycle to verify whether the training plan can cause excessive fatigue of athletes and insufficient training intensity.

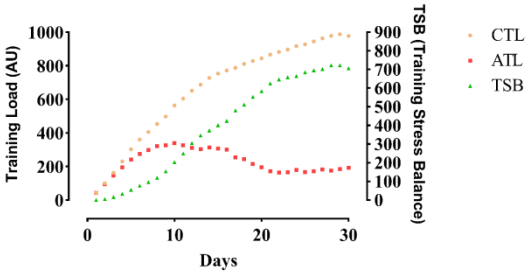


Figure 3: Effect of long and short training loads

Figure 3 illustrates the evolution of training load and stress balance over the course of the program. The left y-axis plots the Acute Training Load (ATL) and Chronic Training Load (CTL) values, while the right y-axis shows the Training Stress Balance (TSB). The x-axis denotes the number of training days.

As the training days progress, the CTL curve exhibits a steady upward trend, indicating that the athlete's long-term training base and endurance capacity are continually strengthening. In contrast, the ATL curve remains relatively flat and centered around a moderate level, suggesting that daily training intensity is well regulated: the athlete maintains a consistent workload without experiencing excessive fatigue. Meanwhile, the TSB curve rises gradually into positive territory, reflecting a favorable balance between stress and recovery. In other words, the athlete is neither overreaching nor undertraining but is instead in an optimal training state with sufficient recovery.

These results confirm that the DDPG-generated training plan successfully adapts to the athlete's changing condition. By dynamically tuning training intensity and volume, the program prevents both overtraining and undertraining, ensuring that the athlete benefits from a balanced, progressive training load throughout the cycle.

This paper conducts experiments to compare with traditional optimization model algorithms. Under the same training cycle, athletes are divided into 5 groups, and the training plans obtained by different models are trained to see their optimization effects.

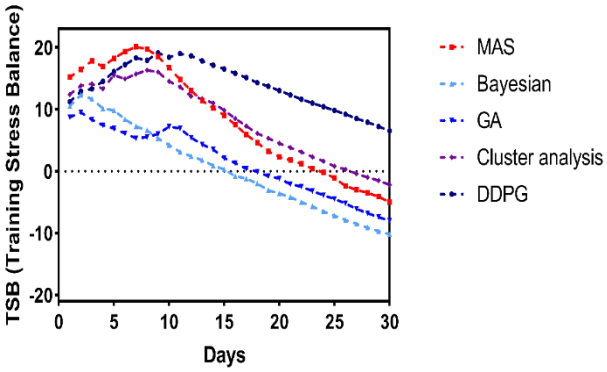


Figure 4: TSB curves of various model algorithms

Figure 4 compares the TSB values produced by various algorithms over the training cycle. The DDPG-based program maintains the athlete's TSB in positive territory throughout, indicating a well-balanced training load, strong recovery, and overall healthy adaptation—effectively preventing the fatigue associated with overly intense regimens.

In contrast, the Multi-Agent Systems (MAS) approach [33] starts with the highest TSB early in the cycle but then declines steadily, suggesting that

cumulative load eventually overwhelms recovery and leads to overtraining. Both the Genetic Algorithm (GA) [34] and Clustering Analysis [35] methods show promising early performance: GA achieves moderate stability, while Clustering Analysis initially trends upward. However, as additional factors come into play later in the cycle, both approaches fail to sustain positive TSB, indicating an inability to adapt intelligently over time. The Bayesian model [36] performs worst, with its TSB declining continuously—signifying inadequate balance between stress and recovery.

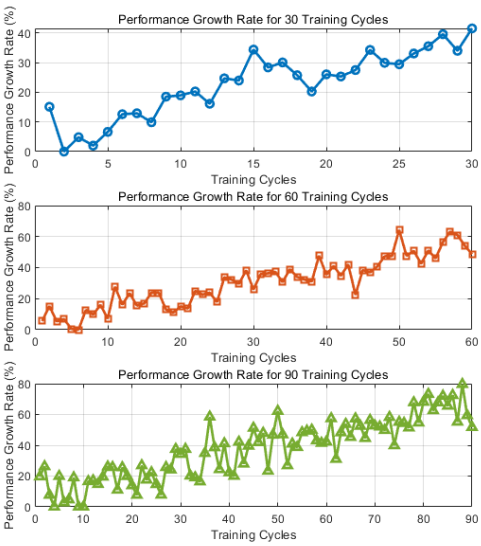


Figure 5: Optimization of different training cycles

Overall, only the DDPG algorithm dynamically tailors its training prescriptions to the evolving environment and athlete-specific factors—preserving an optimal balance between load and recovery and substantially reducing the risks inherent in rule-based or expert-driven programs. To evaluate its performance over extended periods and avoid any one-sided conclusions drawn from a short cycle, we further adjusted our parameters to test 60-day and 90-day training extensions, verifying the model's ability to optimize long-term athlete development.

Figure 5 depicts the athletes' performance growth rates under extended training cycles. Whether operating on a 60-day or a 90-day cycle, performance steadily improves, and the upward trend persists as the cycle lengthens—even over a full nine-month period. This demonstrates that the DDPG-derived program sustains efficient progression without inducing excessive fatigue or undertraining.

To further enhance strategy generation, we integrate an LSTM network whose memory units excel at capturing long-term dependencies in sequential data. By learning from historical game statistics, athlete performance metrics, opponent profiles, and situational context, the LSTM produces personalized, dynamically adjusted game plans that anticipate and respond to sudden developments on the court. We evaluated this capability via a simulated basketball playoff series—each team rotating through eight active players with no additional substitutions—comparing the LSTM-generated tactics against conventional coaching strategies to assess effectiveness under playoff conditions.

Table 4: Strategy generation

Game Stage	Traditional Strategy	LSTM-Generated Strategy
Opening Phase	Focus on defense, stabilize offense	Adjust offensive strategies based on opponent's analysis, reduce errors
Mid-Game Phase	Fixed offensive tactics, enhance defense	Adjust strategies, optimize offense-defense transitions based on athlete fatigue
Critical Moment (Last 5 Minutes)	Concentrate on offense, prioritize key players	Adjust tempo based on score difference, optimize substitutions
Last 30 seconds	Emphasize the last attack and concentrate all players on the attack	Optimize offensive strategies and choose the best attack point according to the opponent's defensive characteristics

Table 4 shows the strategies for each stage generated by the two different methods, and different strategies are formulated for different scenarios.

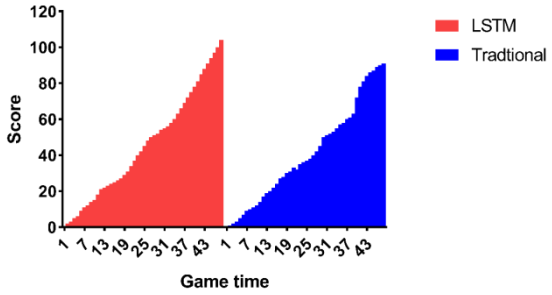


Figure 6: Match score trend

Figure6 compares quarter-by-quarter scoring under the LSTM-generated strategy versus a traditional approach. The team employing the LSTM-driven tactics saw a consistently upward trajectory—outscoring the opposition in each period and finishing with 104 points—demonstrating both a higher overall yield and more stable offensive performance across all four quarters. In contrast, the traditional strategy resulted in a sluggish start, with the team trailing early and ending at 91 points; its scoring curve was uneven, reflecting an inability to adapt effectively to shifting game rhythms. These results underscore the LSTM model's capacity to tailor in-game decisions to evolving contexts, producing robust, scenario-specific strategies that translate into tangible competitive advantages.

6 Discussion

The results of this study demonstrate the effectiveness of the DDPG-based DRL model in optimizing athlete training plans and generating personalized competition strategies, showing substantial improvements over traditional methods. Specifically, the model's ability to handle continuous action spaces and real-time data feedback allows for dynamic adjustments to the training load, ensuring that the Training Stress Balance (TSB) remains within a positive range, thus preventing overtraining or undertraining. This is a significant advancement compared to earlier studies that utilized methods such as Q-learning and DQN (Singh B [18], Huang R [19]), which struggled to manage continuous action spaces and often led to static adjustments in training.

The performance comparison between the DDPG algorithm and traditional methods, as shown in Figure 4, highlights a consistent superiority of the DDPG approach in maintaining a balanced TSB throughout the training cycle. Traditional optimization models, such as MAS (Multi-Agent Systems) and Genetic Algorithms (GA), faced challenges in keeping the training load balanced over time, especially when dealing with longer training cycles, as indicated by the decline in performance (Zheng C [16], Cronin N J [17]).

The LSTM-based strategy generation, which is combined with DDPG in this paper, also contributes significantly to improving the game strategies. As shown in Table 3 and Figure 6, the strategies generated by the LSTM model led to a 13-point improvement in the simulated basketball playoff scenario, outperforming the traditional strategies. This confirms that LSTM's ability to process time-series data provides a more adaptive and responsive strategy, especially in real-time competitive settings, compared to the more rigid traditional approaches (Tran L [21]).

In the context of previous research, this study's novelty lies in its ability to combine DDPG for continuous action space optimization and LSTM for time-series strategy generation, addressing the limitations of earlier models that either lacked the ability to adapt in real-time or failed to handle complex training decision-making processes. This combination of algorithms allows for not only personalized training optimization but also the dynamic generation of competition strategies, which has not been fully explored in prior work.

Furthermore, unlike studies that used psychological or physiological data (Romaniszyn P [1], Demsar U [2]), which often lacked flexibility or real-time adjustments, our approach offers an intelligent, self-adjusting system capable of personalizing both training and competitive strategy, based on a deeper understanding of an athlete's real-time performance.

This paper contributes to the field by presenting an integrated framework that overcomes the dynamic training challenges and personalization gaps found in traditional and earlier machine learning-based approaches, thereby setting a new standard for intelligent sports training systems.

7 Conclusions

This study demonstrates that the DDPG-generated training regimen effectively mitigates both overtraining and undertraining, maintains an optimal balance between workload and recovery, and consequently enhances athletic performance. By monitoring Acute Training Load (ATL), Chronic Training Load (CTL), and Training Stress Balance (TSB), we show that the DDPG plan keeps TSB consistently positive—indicating adequate recovery and the avoidance of deleterious fatigue. Compared with traditional optimization methods, the DDPG approach delivers greater stability over extended training cycles, adapts responsively to individual athlete needs, and yields superior training outcomes.

Similarly, the LSTM-derived game strategies achieve high success rates in practice, dynamically adjusting tactics across game phases and outperforming conventional models. Through head-to-head comparisons, we find that the LSTM framework responds more flexibly to in-game contingencies, optimizes critical decisions—such as offense-to-defense transitions and substitution patterns—and markedly increases win probability.

Nonetheless, several limitations remain. First, although DDPG prevents excessive fatigue, it may underperform under atypical conditions—such as athletes with unusually slow recovery or unusually low training tolerance—if individual variability is not fully captured. Second, while the LSTM strategy generator exhibits

robust dynamic adjustment, its decision granularity warrants further refinement, particularly under extreme game scenarios where strategy resilience is paramount. Third, our validation over specific cycle lengths and a limited athlete cohort may not extrapolate to longer durations or broader populations; additional field trials and model calibration are needed. Therefore, before widespread adoption, further exploration and optimization in diverse, real-world settings are essential.

References

[1] Yulianto W D, Yudhistira D. Content Validity of Circuit Training Program and Its Effects on The Aerobic Endurance of Wheelchair Tennis Athletes[J]. International Journal of Kinesiology and Sports Science, 2021, 9(3): 60-65. https://doi.org/10.7575/aiac.ijkss.v.9n.3p60

[2] Zerguine H, Healy G N, Goode A D, et al. Online office ergonomics training programs: A scoping review examining design and user-related outcomes[J]. Safety Science, 2023, 158: 106000. https://doi.org/10.1016/j.ssci.2022.106000

[3] Rosenblat M A, Lin E, da Costa B R, et al. Programming interval training to optimize time-trial performance: a systematic review and meta-analysis[J]. Sports medicine, 2021, 51: 1687-1714. https://doi.org/10.1007/s40279-021-01457-2

[4] Wang C, Du C. Optimization of physical education and training system based on machine learning and Internet of Things[J]. Neural Computing and Applications, 2022: 1-16.

[5] Zhang C, Lu Y. Study on artificial intelligence: The state of the art and future prospects[J]. Journal of Industrial Information Integration, 2021, 23: 100224. https://doi.org/10.1016/j.jii.2021.100224

[6] Tseng M L, Tran T P T, Ha H M, et al. Sustainable industrial and operation engineering trends and challenges Toward Industry 4.0: A data driven analysis[J]. Journal of Industrial and Production Engineering, 2021, 38(8): 581-598. https://doi.org/10.1080/21681015.2021.1950227

[7] Sharifani K, Amini M. Machine learning and deep learning: A review of methods and applications[J]. World Information Technology and Engineering Journal, 2023, 10(07): 3897-3904.

[8] Taye M M. Understanding of machine learning with deep learning: architectures, workflow, applications and future directions[J]. Computers, 2023, 12(5): 91. https://doi.org/10.3390/computers12050091

[9] Azar A T, Koubaa A, Ali Mohamed N, et al. Drone deep reinforcement learning: A review[J]. Electronics, 2021, 10(9): 999. https://doi.org/10.3390/electronics10090999

[10] Wurman P R, Barrett S, Kawamoto K, et al. Outracing champion Gran Turismo drivers with deep reinforcement learning[J]. Nature, 2022, 602(7896): 223-228. https://doi.org/10.1038/s41586-021-04357-7

[11] Wang D, Hu M. Deep deterministic policy gradient with compatible critic network[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 34(8): 4332-4344. https://doi.org/10.1109/tnnls.2021.3117790

[12] Zanette A, Wainwright M J, Brunskill E. Provable benefits of actor-critic methods for offline reinforcement learning[J]. Advances in neural information processing systems, 2021, 34: 13626-13640.

[13] Lindemann B, Maschler B, Sahlab N, et al. A survey on anomaly detection for technical systems using LSTM networks[J]. Computers in Industry, 2021, 131: 103498. https://doi.org/10.1016/j.compind.2021.103498

[14] Romaniszyn P, Kania D, Bugdol M N, et al. Behavioral and physiological profile analysis while exercising—case study[J]. Information Technology in Biomedicine, 2021: 161-173. https://doi.org/10.1007/978-3-030-49666-1_13

[15] Demsar U, Long J A, Benitez-Paez F, et al. Establishing the integrated science of movement: bringing together concepts and methods from animal and human movement analysis[J]. International Journal of Geographical Information Science, 2021, 35(7): 1273-1308. https://doi.org/10.1080/13658816.2021.1880589

[16] Zheng C, Wu W, Chen C, et al. Deep learning-based human pose estimation: A survey[J]. ACM Computing Surveys, 2023, 56(1): 1-37. https://doi.org/10.1145/3603618

[17] Cronin N J. Using deep neural networks for kinematic analysis: Challenges and opportunities[J]. Journal of Biomechanics, 2021, 123: 110460. https://doi.org/10.1016/j.jbiomech.2021.110460

[18] Singh B, Kumar R, Singh V P. Reinforcement learning in robotic applications: a comprehensive survey[J]. Artificial Intelligence Review, 2022, 55(2): 945-990. https://doi.org/10.1007/s10462-021-09997-9

[19] Huang R, Qin C, Li J L, et al. Path planning of mobile robot in unknown dynamic continuous environment using reward-modified deep Q-network[J]. Optimal Control Applications and Methods, 2023, 44(3): 1570-1587. https://doi.org/10.1002/oca.2781

[20] Muni M K, Parhi D R, Kumar P B, et al. Towards motion planning of humanoids using a fuzzy embedded neural network approach[J]. Applied Soft Computing, 2022, 119: 108588. https://doi.org/10.1016/j.asoc.2022.108588

[21] Tran L, Hoang T, Nguyen T, et al. multi-model long short-term memory network for gait recognition using window-based data segment[J]. IEEE Access, 2021, 9: 23826-23839. https://doi.org/10.1109/access.2021.3056880

[22] Fei Y, Yang Z, Chen Y, et al. Exponential bellman equation and improved regret bounds for risk-sensitive reinforcement learning[J]. Advances in neural information processing systems, 2021, 34: 20436-20446.

[23] Aziza J N A. Perbandingan Metode Moving Average, Single Exponential Smoothing, dan Double

Exponential Smoothing Pada Peramalan Permintaan Tabung Gas LPG PT Petrogas Prima Services[J]. Jurnal Teknologi dan Manajemen Industri Terapan, 2022, 1(I): 35-41. https://doi.org/10.55826/tmit.v1ii.8

[24] Wang H, Liu Z, Peng D, et al. Attention-guided joint learning CNN with noise robustness for bearing fault diagnosis and vibration signal denoising[J]. ISA transactions, 2022, 128: 470-484. https://doi.org/10.1016/j.isatra.2021.11.028

[25] Martin D, O'Byrne J, Cates M E, et al. Statistical mechanics of active Ornstein-Uhlenbeck particles[J]. Physical Review E, 2021, 103(3): 032607. https://doi.org/10.1103/physreve.103.032607

[26] Michaelsen M M, Esch T. Motivation and reward mechanisms in health behavior change processes[J]. Brain research, 2021, 1757: 147309. https://doi.org/10.1016/j.brainres.2021.147309

[27] Dubey S R, Singh S K, Chaudhuri B B. Activation functions in deep learning: A comprehensive survey and benchmark[J]. Neurocomputing, 2022, 503: 92-108. https://doi.org/10.1016/j.neucom.2022.06.111

[28] Maharana K, Mondal S, Nemade B. A review: Data pre-processing and data augmentation techniques[J]. Global Transitions Proceedings, 2022, 3(1): 91-99. https://doi.org/10.1016/j.gltp.2022.04.020

[29] Picornell A, Oteros J, Ruiz-Mata R, et al. Methods for interpolating missing data in aerobiological databases[J]. Environmental Research, 2021, 200: 111391. https://doi.org/10.1016/j.envres.2021.111391

[30] Williams J, Hill R R, Pignatiello Jr J J, et al. Wavelet analysis of variance box plot[J]. Journal of Applied Statistics, 2022, 49(14): 3536-3563. https://doi.org/10.1080/02664763.2021.1951685

[31] Lu R, Jiang Z, Wu H, et al. Reward shaping-based actor–critic deep reinforcement learning for residential energy management[J]. IEEE Transactions on Industrial Informatics, 2022, 19(3): 2662-2673.

[32] Towlson C, Salter J, Ade J D, et al. Maturity-associated considerations for training load, injury risk, and physical performance in youth soccer: One size does not fit all[J]. Journal of sport and health science, 2021, 10(4): 403-412. https://doi.org/10.1016/j.jshs.2020.09.003

[33] Cardoso R C, Ferrando A. A review of agent-based programming for multi-agent systems[J]. Computers, 2021, 10(2): 16.

[34] Sohail A. Genetic algorithms in the fields of artificial intelligence and data sciences[J]. Annals of Data Science, 2023, 10(4): 1007-1018. https://doi.org/10.1007/s40745-021-00354-9

[35] Li G, Liu F, Sharma A, et al. Research on the natural language recognition method based on cluster analysis using neural network[J]. Mathematical Problems in Engineering, 2021, 2021(1): 9982305. https://doi.org/10.1155/2021/9982305

[36] Lindauer M, Eggensperger K, Feurer M, et al. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization[J]. Journal of Machine Learning Research, 2022, 23(54): 1-9.

[37] Zahran L, El-Beltagy M, Saleh M. A conceptual framework for the generation of adaptive training plans in sports coaching[C]//Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019. Springer International Publishing, 2020: 673-684. Https://doi.org/10.1007/978-3-030-31129-2_62

[38] Zhao X, Wang Y. Machine Learning-Based Analysis of Research Policy Impacts on Academic Performance Metrics[J]. Informatica, 2025, 49(19). https://doi.org/10.31449/inf.v49i19.8154

[39] Kumar H, Aruldoss M. Advanced Optimal Cross-Modal Fusion Mechanism for Audio-Video Based Artificial Emotion Recognition[J]. Informatica, 2025, 49(12). https://doi.org/10.31449/inf.v49i12.7392

[40] Touhami S, Belghachi M. A Secure LOADng Routing Protocol Scheme based on Fuzzy Logic[J]. Informatica, 2024, 48(18). https://doi.org/10.31449/inf.v48i18.6131

[41] Tan J, Chen J. Generating context-specific sports training plans by combining generative adversarial networks[J]. PloS one, 2025, 20(1): e0318321. Https://doi.org/10.1371/journal.pone.0318321

[42] Pashaie S, Mohammadi S, Golmohammadi H. Unlocking athlete potential: The evolution of coaching strategies through artificial intelligence[J]. Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology, 2024: 17543371241300889. https://doi.org/10.1177/17543371241300889

A Systematic Review of Security Innovations in Decentralized Finance (DeFi) Using Blockchain Technology

Chnar Mohammed Kareem, Ahmed Chalak Shakir
College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq
E-mail: stcm23002@uokirkuk.edu.iq, and ahmedchalak@uokirkuk.edu.iq

Review paper

Keywords: decentralized finance (DeFi), blockchain, security, smart contracts

Received: January 7, 2025

Decentralized Finance (DeFi) represents the new generation of blockchain financial services by developing an open-access financial model without banking or lending institution intermediaries. However, DeFi's open feature threatens its security, making it vulnerable and a target for different attack types. In this systematic review, we present the security of DeFi by selecting fifteen studies from 2020 to 2024 to determine and display the security solutions' effectiveness in identifying the attacks, focusing on various DeFi components such as smart contracts, DEX, AMM, governance, AMM-based DEX, and smart contracts with (DEX, Oracle); detecting different kinds of attacks (e.g., price manipulation, Oracle manipulation, flash loan) using detection tools (e.g., DeFort, CRPWarner, FORAY); we find out that 40% of the selected studies focus on Oracle manipulation attack, 33.33% for price manipulation and flash loan attacks separately, followed by 13.33% for (MEV, rug pull, front-running, Token Leakage, and deep logical bugs), 6.67% for (EEV, reentrancy, sandwich, access control, and state derailment defects). We compare the studies based on the attack type that they detected using four state-of-the-art types of research, such as DeFiScope, FlashSyn, SecPLF, and DeFiGuard; this indicates the concentration of the trend studies is on accuracy and combining AI in DeFi security, or aggregating the existing tools with it, giving an overview of DeFi components' security, underlining the gaps in the attack types that future research can address to build more robust, trustworthy, and secure DeFi systems.

Povzetek: Narejen je sistematičen pregled varnostne rešitve v DeFi z analiziro 15 študij (2020–2024). Identificirani so glavni napadi (Oracle, cenovne manipulacije, flash posojila), poudarjena je vloga AI.

1 Introduction

DeFi (Decentralized Finance) operates on-chain financial services through lending, investing, and borrowing methods that eliminate dependence on central authority management[1]. As is known, regular economic systems restrict individual data access and transmission time to central authorities or financial institutions, creating problems like chargeback scams, costly fees, counterfeiting currencies, and limited system transparency. DeFi represents a monetary system built on blockchain, which has recently experienced strong interest because it operates without permission while maintaining absolute openness and strong participant interaction through innovative contract platforms based on the Ethereum blockchain [2], [3]. The DeFi-oriented lending processes encompass agent borrowing agreements, decentralized peer-to-peer transactions, and leveraged fund management systems that enable users to obtain crypto assets by paying interest [4]. In addition, governance tokens, which are commercial units used in DeFi, represent user participation in decision-making through changes or new functions by using the power of voting in most systems of the vote [4], [5], as an open system remains exposed to attacks, it represents security risks for participant-held assets [6]. The attacks on vulnerable smart contracts led to monetary losses of 6.45 billion dollars despite researchers' attempts to establish secure Decentralized Applications (DApps) [7]. DeFi challenges that prevent it from reaching its full potential are volatility, usability, fraud, and regulatory uncertainty [2]. DeFi protocols run on Ethereum platforms; these protocol contract codes may have vulnerabilities or bugs, and hackers can utilize them to exhaust the providers' funds from the contracts [6].

Another problem is that the rise in liquidity and popularity of DeFi increases the vulnerability risks for consumers due to its natural openness, leading to potential security threats [8] For instance, many DeFi platform vulnerabilities are subject to attacks. In March 2022, 624 million USD by attackers utilizing a backdoor attack to get the signatures of a third-parity validator and four local verifier signatures were stolen on the Ronin Bridge, as the attacks through the Flash loan in 2021 that targeted PancakeBunny and Cream Finance suffered the loss of 200 and 130 million USD respectively, other incidents on

the GREM FINANCE and DEFORCE platforms via reentrancy cause a loss of 54 USD million [9] and other attacks that continue and increase day by day as a result of the DeFi growing technology with these challenges in DeFi as a new modern technology, enhancing it concurrently by forecasting research and exploring these challenges, depending on blockchain technology to provide innovative solutions in decentralization, tracking, security, and transparency without intermediaries. In this study, we try to determine and show the evaluation of the security solutions in DeFi using blockchain technology through various studies, focusing on how these solutions detect vulnerabilities and attacks and maintain security within different focusses for DeFi elements, like Automated Market Makers (AMMs), Decentralized Exchanges (DEXs), smart contracts, and governance, systematically examining and comparing the security methods utilized in each study that enhance DeFi security in various domain and possibility of leveraging those security methods to other DeFi field such as donation, or combining them for better security approach on DeFi; the review structure follows: Section 2, Background; Section 3, Research Methodology; Section 4, Results; Section 5, Discussion; and Section 6, Conclusion.

2 Background

2.1 Blockchain

The central construct of blockchain technology works via its cryptocurrency systems. Bitcoin stands as the leading standard offered by Satoshi Nakamoto in 2009, performing as a distributed ledger that saves transactions in an open, secure, and irrevocable log through a network of computers [10]. It is a distributed database that executes and follows transactions with allocated nodes. Transactions are stored in connected blocks like a chain in an unchangeable manner, maintaining data privacy, integrity, and confidentiality [11], and it achieves security through decentralization, transparency, and permanent data characteristics, which gives network participants trust in the system [12]. Each block contains a block header in addition to its body, which makes up the minimum components of a blockchain system. Furthermore, the block header has: Nonce, Hash value, Hash value from the preceding blocks, Merkel root, A version number, and Timestamp; the block body contains a binary tree familiar as the Merkel tree, with its root constructed by the transaction hashes [13], as demonstrated in Figure 1. Hashing employs a precise algorithm to generate and change input data of any length into a fixed-length string [14], functions through a consensus mechanism for verifying and processing transactions via nodes and agreement decisions; the most popular consensus algorithms are proof-of-stake (PoS), proof-of-work (PoW), and Practical Byzantine Fault Tolerance (PBFT) [15].

Generally, there are two categories of blockchain: private and public; a private blockchain authorizes companies and organizations to operate blockchain without disclosing their data publicly. Additionally, a public blockchain, on the other hand, is an open and well-known network from which anyone can download the rules through reading, contributing, or posting on the network, resulting in its dispersed and decentralized nature. Ethereum is a decentralized public blockchain framework, not constrained by anyone, which means it is independent [16], [17]; multiple fields apply blockchains, such as health, education, and finance [18], supply chain, voting, real estate, and other domains, this technology provides benefits like robust security, cost-effectiveness, fast transactions, and keeping records transparently, assuring data is tamper-proof through cryptographic hashing, where each block contains the previous block hash and the transaction is validated through network participants and implemented before adding it to the blockchain, ensuring decentralization, protecting data privacy and security, removing intermediaries, and supporting transaction ability [19].

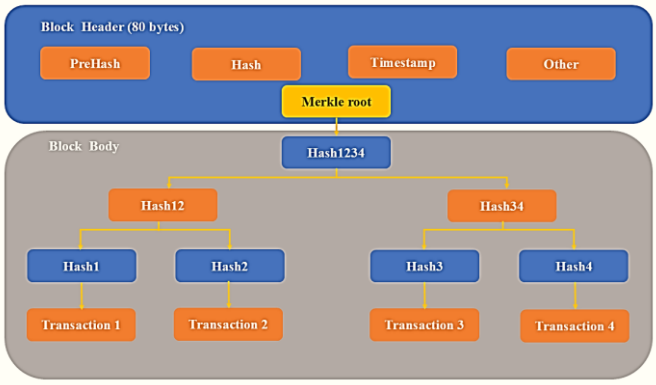


Figure 1: The blockchain Merkle tree content [75].

2.2 Blockchain platform for DeFi

DeFi finds wide applications through innovations like crypto loans, decentralized exchange, P2P lending, etc.; these applications are accessible on different blockchain platforms used for DeFi applications and provide unique advantages, such as:

Ethereum is a protected, public, decentralized blockchain technology for smart contract performance. Through its mechanism, developers create arbitrary applications that standardize operations, basing their designs on consensus features that enable interoperability at scale and streamline development processes. It also implements the blockchain paradigm, which is cryptographically secure and has processes that everyone can access.

Binance Smart Chain (BSC) is a sovereign blockchain that provides users and developers with safety and security; as a native dual-chain interoperability, it facilitates cross-chain communication and scales up the performance of DApps built on 21 validators that confirm transactions, offering decentralization and enabling smooth community involvement through validation.

Solana is a model of blockchain that relies on proof-of-history (PoH) to prove the rank and the transit of periods within events; the leader functions and arranges communications to boost throughput, and the node is handled as a leader to generate a PoH sequence, which gives verifiable passage of time and read consistency. The system also uses a cryptographic hash function whose outcomes are impossible to predict without running the function, which ensures the PoH sequence's integrity. However, relying on a leader node may introduce risks if the leader is compromised.

Cardano is a blockchain project that launched in 2015 to alter the manner of cryptocurrency evolution and construction. Some principal designs of Cardano include the separation of computation and accounting into different layers implemented in a highly modular function of core components, relies fundamentally on capitalizing on these findings to improve the current state of cryptocurrencies using a proof-of-stake consensus mechanism, which ensures security by requiring validators to hold and stake the native cryptocurrency, reducing the risk of malicious behavior.

Avalanche is a high-performance, customizable, secure, and scalable blockchain platform that targets highly scalable and distributed applications and provides security to the blockchain system by withstanding more than 51% of attacks (51% of the miners are attackers), where one subnet validates each blockchain offering many advantages, including compliance, trusted validations, and reduced network traffic.

Polygon is a structured framework for joining networks to establish an appropriate Ethereum blockchain network. By combining networks, Polygon employs security as a service; the security layer is an optional and specialized layer offering a set of validators to verify periodically that Polygon blockchains are legitimate.

Fantom is a directed acyclic graph (DAG) built on a smart contracts platform that aims to address the scalability issues of existing publicly distributed ledgers and takes the Lachesis protocol as a new protocol known to maintain consensus; employing DAG technology helps to provide high reliability for transactions; furthermore, it breaks the sequential processing of transactions; the Lachesis protocol uses asynchronous Byzantine Fault Tolerance (aBFT) to ensure security and enable networks to achieve consensus even with malicious nodes present [20].

While multiple blockchains support DeFi, Ethereum remains the most common platform for DeFi due to its smart contract functionality, proven security, and ability to supply decentralization, interoperability, and scalability; the Ethereum frameworks are the most significant platforms facilitating application establishment in specific fields. In addition, since the Ethereum blockchain integrated with finance more effectively, the appearance of Ethereum and blockchains within smart contracts capabilities led to the evolution of decentralized applications (DApps), opening new opportunities for innovation [7]. Any individual in the Ethereum chain can propagate their DApps and contact others; DeFi technology in the finance field is gaining more interest [18], becoming the most common application in the financial area and supplying a broad financial service [9] Table 1 below shows a summary of the key features of different blockchain platforms [20]. However, within Ethereum 2.0, the risk of Miner Extractable Value (MEV) presents the biggest threat to decentralization, as validators may gain by changing the order of transactions; MEV can result in validator centralization as it advantages well-resourced validators over standard ones; collecting MEV includes significant funds, computational resources, and intelligence that ordinary validators may not have [21].

2.3 Smart contract

It was a term coined by Nick Szabo in the mid-1990s; its features are automated, immutable, and self-enforcing; smart contracts offer security in a way that traditional contract agreements cannot provide [22]. It is simply an agreement between the various parties involved in the application, who have closed the smart contract code; they activate automatically if certain conditions are met in the agreement [23]. In general, smart contracts have several properties that enable them to be known to each peer in the blockchain, implement themselves, not be stopped, and fit various requirements, as illustrated in Figure 2 [24].

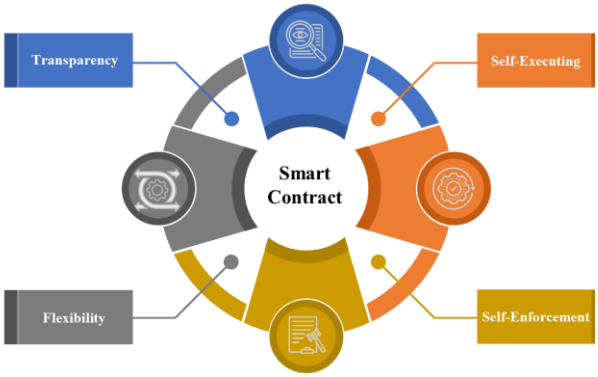


Figure 2: Properties of smart contracts [24].

2.4 Decentralized finance (DeFi)

Financial services developed on Ethereum blockchain smart contracts are referred to as Decentralized Finance (DeFi) [25], [26], stating all terms in financial services and products without relying on intermediaries or institutions; it generally depends on decentralized applications and public protocols [27], [28]. It's also the most common application used in the financial industry to provide various services. DeFi presents a variety of types of applications, such as buying assets linked to the United States dollar (USD), known as stablecoins, on a decentralized exchange, transferring these assets to a decentralized lending platform to collect interest and apply them to a decentralized liquidity pool, or depositing money directly on the chain. In addition, the core of all DeFi applications and protocols are smart contracts that describe a small application stored in a blockchain [2],

representing a monetary system established on public blockchains. The features of available finance services are decentralized applications (DApps), digital assets, smart contracts, and protocols constructed on the blockchain, as shown in Figure 3 [29]. When the DeFi applications start running on the blockchain, the smart contract modifies the transaction state that attackers may exploit, which would cause irreversible harm to DeFi, so making the contracts more secure is critical [7].

Figure 3: Decentralized Finance components [29].

Financing: In the financing sector, DeFi sustains separate models: prepayment financing, receivable financing, and inventory financing. Newly, many requests for small-to-medium enterprises (SMEs) have been made in financial operations worldwide, and they face a serious challenge in funding due to poor creditworthiness and unreliable company information. Blockchain technology proposes a solution with DeFi features, ensuring data transparency and security, enhancing trust between financing parties using smart contracts on Ethereum, facilitating real-time monitoring and accurate data collection, improving SME information's credibility, and simplifying investor financing access [30].

Supply Chain DeFi can strengthen the supply chain systems based on DeFi properties on the blockchain, which is robust for traceability. Such a system contains planning, services, distribution, acquisition, and storage, where the data is saved in a block, indicating that each supply chain link is tamper-proof [30]. Supply chain finance (SCF) also reduces costs and enhances employed capital by allowing sellers and buyers to optimize their working capital in trade transactions. Combining DeFi with the SCF introduces a range of attractive opportunities that can transform how businesses interact and work within the supply chain ecosystem.

Table 1: Comparison of blockchain platforms' security for DeFi applications [20].

Platform	Features	Security Mechanisms	Advantages of DeFi Applications
Ethereum	Decentralized, smart contract, and secure	Cryptographically secured and single-instance machine.	Developer-friendly and expandable
BSC	Sovereign blockchain and dual-chain.	A total of twenty-one validators	Cross-chain with decentralized governance
Solana	PoH and high throughput.	PoH sequence and a cryptographic hash function.	Large capacity and quick execution
Cardano	Modular and separation accounting.	PoS, the validators hold and stake cryptocurrency.	Improves the design and energy performance
Avalanche	It is Adjustable and scalable.	Subnet-based validation 51% attacks.	Expandable and reduces the network bandwidth.
Polygon	Ethereum-compatible framework.	Security layer with the validator.	Scalable for Ethereum DApps.
Fantom	DAG and Lachesis protocol.	aBFT for malicious node tolerance.	The transaction is reliable and efficient.

2.5 DeFi applications in Ethereum

In this section, we present some of the DeFi applications on the Ethereum blockchain to help the reader understand many DeFi applications by explaining them so they can understand them, such as:

Thanks to smart contracts, decentralized platforms, and the use of the blockchain system, DeFi can address issues that traditional SCFs, those oriented toward providing access to finance, traceability, transparency, improved efficiency, and process automation, can face

challenges when they include restricted financing, inefficient operations, and a lack of transparency [31].

Loan DeFi has significantly lowered the barriers to financial services, and one of the many DeFi trends in Aave 2020 is the launch of flash loans; these loans permit borrowers to acquire any amount of equity without collateral and for a fee of only 0.09%, unlike conventional loans, which need collateral to secure the borrower, flash loans feature instant borrowing and disbursement in one transaction; if some conditions are not satisfied, the deal will regenerate, and the borrower funds will not be at risk. Flash loan security depends on smart contracts and blockchain technology, which are irreversible and immutable; this modification offers mortgage loans and increases access to credit institutions. However, Flash Loan as a DeFi application employs loan services: The borrowers repay and borrow the funds with the same transaction, which implies that after getting funds, the borrower must use them immediately to pay back and turn a profit on the loan. In this manner, if the borrower defaults on the loan or any conditions in the smart contract are not met, the transaction will roll back, and all things will go back to the situation before borrowing without collateral, and the lender does not have to worry about lending risks; because of the blockchain and smart contracts' characteristics of non-repudiation and tamper-proof, Flash loans enable security and feasibility. Neither lender nor borrower can cheat or change the terms; as a DeFi application, Flash Loan gives new services of loans using blockchain technology, even without collateral, with a low fee and without worry of risks by borrowing funds through Flash Loan DeFi increases the number of customers and significantly lowers the bar on loan services [30].

Stablecoins are one type of digital currency that maintains stability for fiat currency and crypto assets, which differs from Ethereum and Bitcoin, except that they have minimal price fluctuations, with their resilience correlated to traditional currencies such as fiat and gold [30]. There are four properties for stablecoins: tokenized, meaning that a smart contract manages cryptocurrency tokens; convertible, which allows conversion to another currency or the pegged asset; tradable, which enables direct trading between parties; and fungible, meaning units of financial value with a bit of pricing volatility regarding their pigged assets or index. They use most of the stablecoin applications, and they can develop with different characteristics [32].

Charity Projects Describing the utilization of DeFi in various Ethereum applications, such as supply chain, banking, and finance, with the leverage of the Ethereum blockchain and smart contracts in the donation field, taking advantage of transparency, which earns donors' trust and simplifies the use of money [33]. DeFi can integrate two methods: the variable, non-linear process of the UniSwap decentralized exchange (DEX) coin system for secure offerings [34].

2.6 Real-world operation of the DeFi application

DeFi applications concentrate on financial services. DeFi covers a wide range of application sectors [35], such as:

Decentralized Exchanges (DEXs) enable users to exchange digital assets using their wallets through a supervision-resistant platform in parallel without any custody for exchanging tokens, as UniSwap allows users to trade tokens without any middleman involvement because DEXs provide users with lower fees, asset control, and privacy, attracting more users and higher volumes than centralized exchanges [36], enabling cryptocurrency users to offer exchange liquidity and earn transaction fees using AMM algorithms. A DEX does not require centralized management because it handles liquidity through AMM algorithms. In contrast, the different decentralized liquidity pools present opportunities for investment and profit but expose shareholders to risks that must be analyzed and managed [37]. DEX with AMM protocols is the most familiar DeFi application, which provides benefits like continuous liquidity, decentralization, and automation, where the word AMM refers to the algorithm of a protocol, while DEX refers to the application or protocol use case; as for blockchain-based DeFi, there are order books-based DEX, such as dYdX and Gnosis, which do not depend on AMM algorithms, with the use of AMM-based DEX, one must consider security and privacy into account; being a distributed, complex system with different hardware and software elements communicating together; AMM-based DEX is particularly susceptible to interface attacks [38].

Automated Market Makers (AMMs) are essential in DeFi for liquidity and DEXs by implementing protocols such as Balancer and UniSwap to ease liquidity pool trading and determine asset costs, improve DeFi market expansion and liquidity by boosting price detection, trading, and liquidity [36], supplying liquidity algorithm by determining users' funds and setting prices using a conservation function, in this liquidity, two base actors consist of the general mechanisms of the AMM: the first is called liquidity providers, who contribute funds to the assets pool, and the second is known as traders, who interchange assets for another, and the exchange rate between structured assets depends on the amount desired for trading. The most vital AMMs enclose Bancor, Balancer, Curve, and UniSwap [39].

Decentralized (Lending and Borrowing) DeFi services, such as borrowing and lending, allow organizations and individuals to earn interest by borrowing digital assets, giving cryptocurrency loans to consumers, and acquiring stakes; meanwhile, the borrowers ensure their valuables on these platforms, replacing traditional banking because of their quick accessibility and sensible cash interest rates [36].

Synthetic Assets and Derivatives establishing fiat currencies, evolution, trading, duplicating equities, and switching commodities formed the protocols of synthesizing assets in the Synthetix, which reduced the need for intermediaries, enabling hedgers and DeFi

investors to access different assets through mirrors and Synthetix protocols despite intermediaries, people can look for traditional financial market exposition to choose synthetic assets for availability, liquidity, and flexibility possible in controlling their transactions and financial services. This application democratizes monetary services and enables considerable financial participation throughout a wide range, as it continues developing and growing [36].

Yield Farming and Liquidity Mining are practices where intensive users share their tokens to supply passive liquidity to DeFi platforms. Yearn and Curve Finance reward consumers who deposit assets into liquidity mining and stake tokens into governance systems or contribute to the liquidity pools. Yield farming is users' most common strategy to increase profits and earnings from passive income [36].

Governance Tokens and Decentralized Autonomous Organizations (DAOs) emerge as member-owned communities without central management, governed via the principle of democracy, and individuals determining decisions together, whereas in DAOs, voting power tends to be expressed by held tokens, and the proposal implementation follows the voting process. Most DeFi projects have an associated governance token where people can vote based on the level of activity in the service. Tokens have to be classified by regulatory authorities. To decentralize governance partly, token holders can offer limited voting rights, and the developers may not be required to make changes if they hold many tokens on the platforms related to ideas and policies. The governance often incorporates communication in both formal and informal forums. Moreover, DeFi protocols persist on the Ethereum blockchain and have a form of participation where individuals can vote on issues affecting the blockchain. A quorum is needed to pass votes, which can cancel the proposal, as the decentralized governance in blockchain and DeFi applications introduces significant risks. It's even feasible to attack a protocol that enables users to borrow many governance tokens permanently, but just enough to vote against other users. The DeFi protocol's pseudo-decentralized governance operates by a few users having governance tokens, which causes concern about the token concentration, where the governance token holders may not be the only factor considered in DeFi applications or blockchain protocols, as some organizations with voting authority may affect the decisions [40].

2.7 DeFi challenges

Besides the benefits of DeFi, it faces various challenges as a new technology. This section lists some of these challenges, explains them for clarification, and helps future researchers address them and find solutions to improve the use of DeFi. The challenges are:

2.7.1 Regulatory

It provides financial system stability and security, as the DeFi platform's decentralized nature and the lack of central authority present challenges and make it hard to apply Know-Your-Customer (KYC) and Anti-Money Laundering (AML) regulations, allowing criminals to utilize DeFi for their illegal activities, and it contains DeFi services that present significant risks as they do not require identity verification or creating an account [41]; utilizing Zero-Knowledge Proofs (ZKPs) could overcome regulatory challenges, enabling people to execute transactions and keeping privacy for sensitive details to improve security in DeFi ecosystems, the ability to maintain confidentiality and verify the information is crucial, whereas, in peer-to-peer transactions, trust and protection are essential by combining blockchain and ZKPs, DeFi platforms can follow regulatory requirements like AML and KYC without exposing user privacy, for instance, ZKPs can qualify the user to confirm their residency or age without revealing their address and birth date, observing General Data Protection Regulation (GDPR) that imposes lowering data demands [42].

2.7.2 Scalability

Blockchain networks determine the scale of diverse DeFi outlets, so platform creators lean on layer two scaling to handle expanding volumes efficiently, as the network's transaction allocation system faces difficulties when users conduct multiple transactions due to the fee costs that can delay the network procedure. Recently, the asset platforms of DeFi processes were separated, which restricted communication among them, as enhancing interoperability and scalability is critical to improving user experience and making DeFi development [43]. The speed and durability of transactions on blockchain infrastructure depend on its network bandwidth and growth rate because AMM-based DEXs exist on this platform. All DEX transactions depend on time to be verified on the blockchain network before they become activated. The verification operation relies on the validators or miners instead of DEX; unlike Centralized Exchange (CEX), which executes transactions immediately, DEX incurs a process delay in transactions from seconds to hours. Various blockchain networks with velocity requirements were created (e.g., EOSIO, Tezos, XRPL).

On the other hand, the blocks utilized for storing data on DEX transactions limit the total number of transactions that DEX can approve per patch, and their throughput is still far behind CEX. Several multi-layer blockchains, such as layer-2 blockchains, were presented to fix the throughput problem [38]. The blockchain network Ethereum, which hosts the most DeFi applications, finds difficulty with the requests growing due to the rise of the DeFi platform's usability and popularity, especially for users whose price is out by excess gas fees; DeFi platforms may become inefficient during intervals of network congestion, it limits the ability and lowers the efficiency of DeFi to challenge regular financial systems that can process transactions with minimum costs and much faster; several solutions, such as alternative blockchain networks like Solana and Polkadot and layer-2 scaling technologies, handle these limitations, but the adoption and development are still early, as DeFi platforms will struggle to reach their potential until the scalability issue is solved.

Zk-Rollups and Optimistic Rollups are layer-2 protocols that ensure the final arrangement on the main blockchain by processing off-chain transactions, reducing fees, and expanding transaction throughput, making DeFi serviceable for everyday transactions and its application availability.

A ZKP solution creates transaction assurance through privacy enhancement methods that do not require disclosure of full details. The global adoption of DeFi involves applying those technologies to manage existing limitations [44].

2.7.3 Security

DeFi security is related to three characteristics: infrastructure risks, failure in interdependence, and smart contract vulnerabilities. All services and applications that DeFi offers are built on smart contracts to handle users' finances, indicating that more funds are associated with these contracts, making them more targeted by the attackers, as their code is composed by any person who can find a bug in it and steal the funds inside it, thus, contract developers should concentrate on designing such a contract, testing it, and auditing its security before deploying it on the blockchain, another security issue is the infrastructure, which needs to be considered through the design of DeFi applications, checking that no congestions exist on the network because loaded networks could miss out on members' queues, leading to not storing valid transactions by honest members. Therefore, the main feature of consensus affects the attributes of the application's security [45]. A critical part of DeFi security is exploring vulnerabilities as they happen or before they occur (i.e., preventative), and common vulnerabilities include these parts:

- Deployment of malicious.
- Attack execution.
- Fund extraction.

Hackers used private pools to withdraw $197 million from Euler Finance. Yet, they established the malicious smart contract a few blocks before the attack transaction began, giving them a critical opportunity for prevention and intervention; another example is the attack on the Rubic exchange incident, resulting in a $1.4 million loss, the attackers released a malicious contract and immediately executed the attack transaction after deploying it with private pools [46], another example of attacks, in February 2020, DeFi protocol bZx suffered from two sequential attacks; the attackers utilized the logic flaws in bZx to accomplish arbitrage at a low cost (i.e., robbing over $8 million ETH at that time), this is not an isolated event, in April 2020, another DeFi protocol, UniSwapV1, was compromised by cybercriminals, stealing 1,278 ETH through exploiting a reentrancy vulnerability attack that arises on DeFi protocols approximately every few months. More than $130 million was also lost when Cream Finance experienced an external attack [47].

2.7.4 Oracles

Different DeFi products depend on external information, such as exchange prices that Oracle supplies. These data in Oracle affect the user and the contract's behavior, and the challenge appears when transforming that exterior data from outside into the chain, on which DeFi product security relies on the precision, credibility, and validity of information from Oracles. Thus, an Oracle is valued based on its liability, transparency, and required trust levels [48]. However, Many DeFi activities, including tampering with Oracles and delivering faulty information to smart contracts for personal benefits; an instance of an Oracle attack that includes modifying the token price denoted by the Oracle from an attack on the Venus Protocol, where hackers used 900,000 XVS tokens to extend the price referenced from the Venus Oracle by (US$80 to US$145) via a hack on the Venus protocol, taking (US$77) of the system [49]. In addition, on February 2nd, 2022, a wormhole, which is a multiple-purpose cross-chain that supports about 35 blockchains, hacked Solana by leveraging a bug in the wormhole smart contract on Solana, letting the attacker make an arbitrary verification payload, execute one transaction, and mint 120,000 wETH, worth around $350 million on Solana [50]. Also, the NFT game Ronin and DeFi cross-chain Wormhole Network hackers attacked them with $611 million, and $622 million serially disappeared. According to the REKT database, the DeFi protocol lost $77.1 billion due to hacks, scams, and vulnerabilities, and only $6.5 billion has been returned [47]. Oracle also presents a danger to the investor's funds in DeFi to keep the liquidity available for the user to trade, making them independent of counterparties and minimizing the risk of market manipulation, a solution called the function Oracle and the AMM, which is an entity programmed to serve as an intermediary between the AMM and Oracle it maintains the pool that returns and collects a user-wrapped premium; function Oracle provides a dynamic discovering price mechanism that modifies user evaluation, enabling price flexibility even when the counterparties are absent [51]. Therefore, Oracles serve as a critical component for properly functioning DeFi protocols. Price Oracles provide crucial information that impacts the execution of smart contracts and their results; for example, the DeFi protocol Compound gets the Oracle price from many resources, like the centralized Oracle service suppliers Chainlink, and the price of trading assets from UniSwap as a decentralized protocol.

On the other hand, the values of those resources may differ from the actual data; when the price of a digital asset varies, determining the asset price can be difficult [52]. There are multiple types of Oracles: a decentralized Oracle gets the data on-chain, e.g., from other DeFi protocols, and the centralized Oracle depends on a reliable third party, which violates the DeFi concept; for instance, a decentralized Oracle may obtain the swap rates between tokens from an AMM-based DEX. However, this method is vulnerable to manipulation attacks that alter the price of DeFi protocols called Oracle attacks [53], or on-chain Oracle manipulation, which emphasizes the importance of

automatic tools like DeFiPoser and DeFiRanger, which are tools with the capability of addressing the vulnerabilities of Oracle manipulation [54]. Other studies suggest Over, an automated approach for assessing Oracle variance and its effects on DeFi smart contracts, and provide a smart contract protocol source code and the scope of certain Oracle variables in the contracts by automatically examining its source code to get the protocol summary for a secure requirement to the protocol, identify the best way to set essential control parameters in the contract, this will ascertain that the produced contract continues to meet the need of constraints, even when Oracle deviations arise [52].

2.7.5 Liquidity

It is well known that DeFi produces market volatility through its speculative platforms and exposure to market volatility. The price of assets changes on DeFi platforms because the platforms depend on the liquidity pools as their transaction facilitation method. Users may face difficulties executing transactions and accessing their funds because of the changing liquidity levels in DeFi systems. Communication within traditional finance and DeFi may increase these issues. The Liquidity Constructions of DeFi impact the change of regularity in financial markets, mainly if a link exists among financial sectors [55]. That results a temporary loss that appears due to AMM, which only affects the Liquidity Providers (LPs) to AMM DEX and consumer services to the protocol of yield farming that utilized liquidity provisions AMM, DEX in their approaches, and gets worse because of the cryptocurrency volatility and prevented by giving equal or similar values to token pairs liquidity, such as synthetic tokens targeting the same value (e.g., a pair of USD stablecoin) or synthetic tokens and their target tokens (e.g., native tokens), another attack targeted liquidity the Just In Time Liquidity (JIT) which is one of the Miner Extractable Value (MEV) attacks appears at the Concentrated Liquidity AMM (CLAMM), which refers to the improvement to AMM that increases the efficiency of the capital in the liquidity pool, introduced by UniSwap V3, letting the LPs define the price domain, for instance, in UniSwap V3, one of the transactions implemented the JIT attack, that will withdraw liquidity within the pool by splitting the transaction into smaller units, it is secure against MEV attacks [53].

2.7.6 Transparency

The blockchain implements and saves transactions redundantly, fostering transparency and trust in executing transactions, presenting significant constraints for individuals and institutions, as the visibility of publicly recorded transactions affects the institution and users, which increases when critical information is transparently stored, such as Centralized Exchanges (CEXs) from KYC processes can use metadata, patterns of transactions, and business information to remove anonymous users. Despite many blockchains ensuring pseudonymity, which increases risks of violating data protection regulations, such as antitrust laws or the GDPR, when working on DeFi platforms, for example, a competitor who can observe the competitor's financial transactions of an institution raises a risk. Also, the economic challenges due to the transparent implementation of transactions, known as Extractable Value (EV), are some of the well-known examples of front-running attacks; one of the EV examples occurs when block arbitrageurs or producers exploit volatile prices in the mempool from publicly visible transaction data, for instance, an arbitrageur buys a token before a large DEX transaction request and sells it when the price rises, gaining profit from the margin; presenting solutions like ZKP's Privacy-Enhancing Technologies (PETs) and gradual decryption to reduce these challenges in cryptocurrencies, combining transparency with economic exploitation and information protection [35].

3 Methodology

We used a systematic review on "security innovations in Decentralized Finance (DeFi)" guided by the methodology specified by Okoli and Schabram [56]. The sections below describe the methods applied throughout the review. The study aims to review research studies related to security in DeFi and identify research gaps consistent with the study objective of determining and evaluating security solutions to identify attacks in DeFi using blockchain technology.

3.1 Research questions

We define our research questions to be answered through the review based on the study objective as follows:

Q1: What type of attacks affect DeFi security?

Q2: What solutions do the studies offer to address those attacks?

Q3: How can those security solutions enhance the security of DeFi platforms?

3.2 Search technique

In this study, we used a scientific database with established reputations, such as Google Scholar, ACM digital library, and IEEE Xplore, typing relevant keywords to the subject, such as "DeFi security", "DEX security", "DeFi security on the blockchain", "AMM security", "DeFi smart contracts security", and "Governance token security" or "DeFi and Governance token security", gathering studies about DeFi security. The study contained primary publications from 2020 to 2024 to provide the most up-to-date and accurate research findings for evaluation.

3.3 Inclusion and exclusion

This section includes papers handling questions, with the review concentrating on the security solutions they utilized in DeFi via blockchain technology. It employs and evaluates the criteria for the exclusion and inclusion of the selected studies for applicability and quality, retaining

them based on their titles, abstracts, and relevance to the review subject, as illustrated in Table 2.

Table 2: Inclusion and exclusion criteria of related studies.

Inclusion	Exclusion
Studies published from 2020 to 2024.	Studies publicized before 2020.
Open access.	Duplicated records or not fully available text.
Written in English.	Not in English.
Studies related to DeFi security, like AMM, DEX, AMM-based DEX, governance tokens, and smart contracts	Studies are irrelevant to DeFi security in blockchain or do not include DeFi-based blockchain.
Journal, conference, and preprints	White paper with books and editorial comments.

3.4 Data extraction

The review selected appropriate and relevant data with a focus on DeFi security by selecting studies and including them based on inclusion criteria that analyzed and processed different data from the studies to the security of DeFi components like DEX, smart contracts, AMM, governance tokens, and encompassing fifteen studies referenced in [8], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], and [70].

3.5 Data analysis

By employing a data extraction model, we abstract information from the studies and construct a form based on five components for examining and evaluating throughout the review. The selected papers have already determined three key themes via research questions, including the types of security attacks, proposed solutions, and their effectiveness in improving security in DeFi, as displayed below the five elements of the form in Table 3.

Table 3: Data extraction elements from the form.

Items of Data	Definition
Title	Papers title
Year	Publication year
Authors	Name of the authors
Country	Country of the researchers
Type	Conference, Journal, and Preprints

4 Results

This section outlines the results from 15 studies that we found in the systematic review, demonstrating the DeFi components (AMM, DEX, AMM-based DEX, and smart contracts), the vulnerabilities they handled, mitigation approaches, and evaluation procedures, followed by Table 4, which provides a contrast between the studies for better understanding. Figure 4 below shows the process of including the studies, eliminating 15 of 225 studies because of duplication, 210 records remained for screening then 115 were incompatible via their title and abstract, 60 studies were already analyzed at that time, removing another 45 studies since they did not include enough information for the review, in the end, we choose fifteen studies for to the review incorporated into the data extraction phase, as displayed in Figure 4.

4.1 Graphical distribution and publication years

Various studies analyzed the security challenges they handled in DeFi through blockchain technology, including 15 relevant studies, as demonstrated by the publication years in Figure 6. Seven were published in 2024, four in 2023, one in 2022, one in 2021, and one in 2020, as illustrated in Figure 5, indicating that adopting blockchain technology for the security of DeFi remains in its earlier phases and is evolving. On the other hand, Figure 6 shows the appearance of China (46.7%), the United States (26.7%), the United Kingdom (13.3%), and India and Singapore each produce (6.7%).

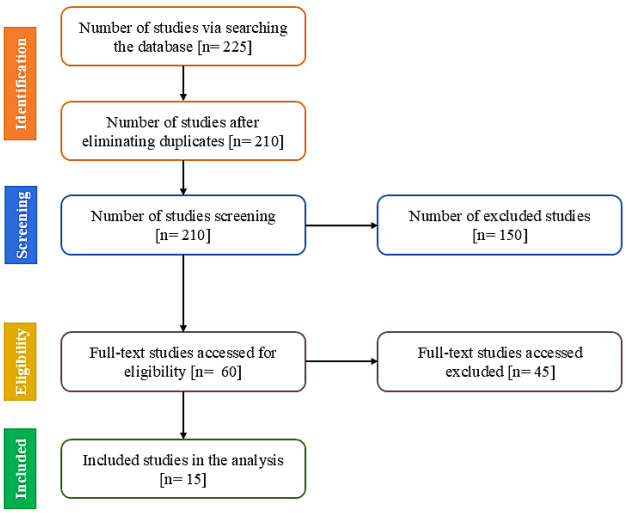


Figure 4: Review process flowchart.

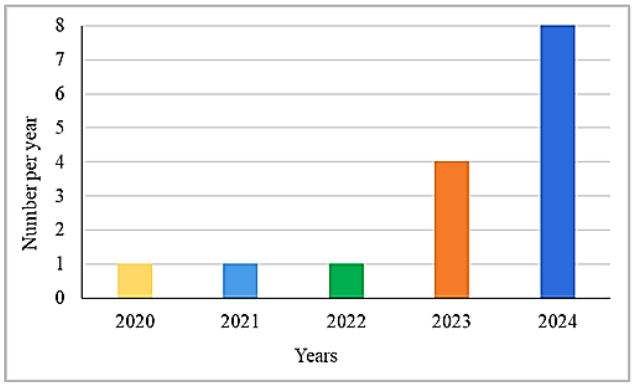


Figure 5: Publication year for the selected studies.

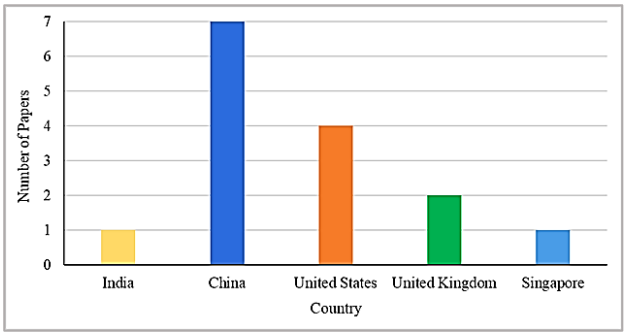


Figure 6: Graphical distribution of the selected studies

4.2 Publication source

The research publications distributed themselves across seven conference venues, four journal publications, and four preprints, as shown in Figure 7.

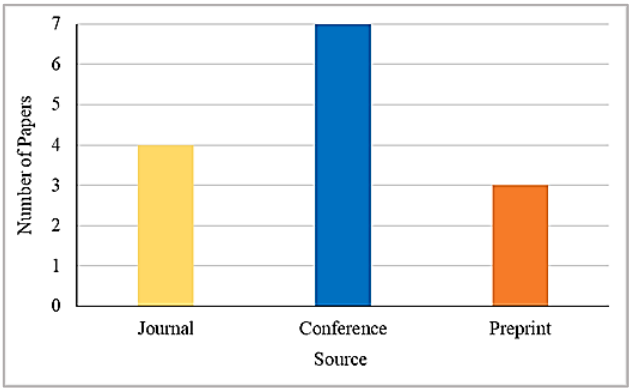


Figure 7: Review distribution by publication source.

4.3 Summary of the studies

- Amit Kumar et al. [8]. Explained in their study how to detect vulnerabilities in DeFi by using BLOCKEYE, due to the lack of existing tools to determine the vulnerabilities related to Oracle dependency like Codefi, by focusing on protocols that are processing on the Ethereum platform, such as Synthetix, Kyber, and UniSwap, via static analysis and dynamic monitoring transactions at runtime, combining SERAPH symbolic analyzer reasoning about Oracle asset feed, the examination of eight real-world DeFi protocols does not show any false positives or false negatives compared to Codefi. Yet, it does not address the scalability challenges to employ it across different blockchain platforms (e.g., BSC, Solana, Avalanche, etc.) and its possibilities to utilize it with other DeFi models such as AMM, DEX, governance tokens, etc.
- Zecheng Li et al. [57]. The authors introduced SolSaviour, which executes voting mechanisms and a Trusted Execution Environment (TEE) cluster to protect DeFi protocols and deploy smart contracts from attacks and vulnerabilities. The study manages the vulnerability issues in deployed DeFi and smart contracts employing a Trusted Execution Environment (TEE) cluster and voting mechanisms by examining 12 real-world vulnerable smart contracts, such as Fei protocol, DAO, and POWH coin, and mitigating the attacks to 2.3% for Fei protocol, 6.5 for DAO contracts, and PoWH coin to 0%, as d=for latency around 6.7%. However, it possessed restrictions like TEE dependency risks and manual patching requirements, with the study focusing on its two core VoteDestruct and TEE cluster, implementing more research to address how to protect the framework from possible attacks that may arise from the core contents, like a failure of one point in the TEE cluster and the manipulation or governance attack.
- Siwei Wu et al. [58]. Proposed an independent platform called DeFiRanger to determine price manipulation attacks by developing Cash Flow Trees (CFTs) and applying low-level semantics to advance DeFi's actions, such as deposits, withdrawals, and transactions through crossbreed approaches, identifying a pair of attacks: manipulated prices for the DEX pool, and attacks targeting weak DeFi applications that rely on manipulated price Oracles, examining 41 real-world, actual incidents, reaching 0.996 precision and 0.962 True Positive across 15.272 transactions; but it detects the attack types depending on external data, the possibility of false positive and negative which restricting the analytical scope, as the complex nature of the DeFi environment when new attack type rises.
- Conor McMenamin et al. [59]. Present FairTraDEX, a decentralized exchange protocol that uses Frequent Batch Auctions (FBAs) to supply formal game-theoretic assurances against extractable value, addressing the Expected Extractable Value (EEV) where players inject order or censor to obtain profits that UniSwap DEX suffered with, by duplicating key characteristics of an FBA with a various set of zero-knowledge protocols and escrow-enforced commit-reveal mechanism, protecting against EEV and execution costs, guaranteeing a fixed fee model independent of order sizes for large retail users seeking to avoid prohibitive execution costs. However, it does not fully address how the protocol will operate in irrational behavior of market manipulation, which is common in many trading environments.
- Zewei Lin et al. [60]. Proposed an automated analysis method that assesses the risk of contract-related rug pulls called CRPWarner, which decompiles the Ethereum Virtual Machine (EVM) bytecode, builds a Control Flow Graph (CFG) and operates a domain-specific datalog analysis, employing three types of contract-related rug pulls: hidden mint function, limiting sell order, and leaking token, by establishing a border classification of rug pull behavior through manually analyzing 103 real-world rugs pull events, addressing rug pulls in contracts, and transactions-related rug pulls as primary types. As a result, CRPWarner achieved 91.8% precision, 85.9% recall, and an 88.7% F1 score, exploring a broad dataset of 13,484 real-token contracts on Ethereum; still, as the method shows, it just offers static detection for contract-related rug pulls; it can't provide real-time monitoring, limiting its capabilities to inform users after the contract deployment.
- Wenkai Li et al. [61]. Present DeFiTail, a technical DeFi check framework that learns the invocation patterns in data paths using Deep Learning (DL) to solve

invocation pattern learning, external and internal path unification, and data path feasibility validation via rows and graph learning technology transforming data paths into opcode sequences and constructing a heterogeneous graph to extract sequential execution process characteristics, also studies the exterior transactions in Ethereum Virtual Machine (EVM), combining external and internal paths with function segmentation and Control Flow Graph (CFG) structure; as stated by the authors, DeFiTail overcomes 98.39% of access control and 97.43% flash loan threats. Yet, the study lacks the evaluation of executing latency impact, which reduces its effectiveness for detecting threats and responding in real time.

• Liyi Zhou et al. [62]. Introduce Automated Arbitrage Market Maker (A2MM) service, which creates atomic swap routing mechanisms that automatically perform two-point arbitrage between UniSwap and SushiSwap AMMs to secure blockchain protocols from Miner Extractable Value attacks. The study reduces block-space consumption by 32.8% and transaction fees by 90% while mitigating 88.8% of back-running arbitrage transactions associated with back-run flooding (BRF). Yet, it has drawbacks, like requiring smarter contract logic, which increases gas costs and fails to identify the issue when interacting with multiple AMMs, problems related to scalability and assumptions about the security of smart contracts, and frequent vulnerabilities, such as reentrancy and Oracle manipulation, are also not addressed.

• Jiahua Xu et al. [63]. Introduce (Auto.giv) to operate as a framework to protect DeFi platforms by decreasing security risks through Deep Q-Network (DQN) reinforcement learning, which analyzes AAVE-like DeFi model changes by altering protocol parameters to sustain security and operational excellence over traditional methods, documenting that the agent can create robust, profitable decisions in just 20 minutes of training, showing a reactive, efficient, and objective solution compared to traditional procedures. Yet, the evaluation process does not examine how Auto.gov would perform in different protocols and chains.

• Maoyi Xie et al. [64]. Propose a framework called DeFort to determine and study the price manipulation attacks via price pattern models and multiple token examination tools for dataset evaluations in D1 and D2. It functions automatically to detect transactions that cause abnormal price fluctuations and recognizes attackers and victims by comprising three elements: an on-chain monitor, a model-driven detector, and a model-driven analyzer. The study reached a recall rate of 96.3% and attained zero False Positives (FP) on D2. However, the study focuses on transaction execution, not the source code level, and it does not explain the speed of the detection process or the gas or computational costs.

• Yongge Wang et al. [65]. Introduce an AMM CoinSwap-based-constant ellipse cost function for AMMs and compare it with the existing AMM mathematical models, such as the Logarithmic Market Scoring Rule (LMSR), Liquidity Sensitive LSMR (LS-LMSR), and mean/constant product/sum, which aims to lower slippage-based front-running attacks, by examining the gas price corresponding to UniSwap V2 and UniSwap V3, which reduced the gas cost to 44.99% for UniSwap V2 and 184.29% for UniSwap V3; but, it does not overlook other potential exposures that AMMs face, such as liquidity or market manipulation attacks; further research could offer more about the threats concerned in AMMs.

• Zongwei Li et al. [66]. Introduce StateGuard, a deep learning-based platform for assessing and discovering state defects in DEX projects, and locate a new kind of state derailment fault in DEX smart contracts as its structure from Abstract Syntax Tree (AST) includes five dependent elements: data dependencies, declaration dependencies, expression dependencies, function dependencies, and control dependencies, it also applies a Graph Convolution Network (GCN) to determine state derailment defects, matching the StateGuard efficiency within Confuzzius, Conkas, Oyente, Securify, and Mythril utilizing 2,000 contracts through the SmartBugs dataset, it revealed that StateGuard overcame other tools in each of five key performance metrics, with 22.31% accuracy and 7.39% in F1-score; despite graph optimization methods providing benefits. The study has a complexity issue in many AST traversals for comprehensive analysis, and it does not assess its effectiveness in non-DEX DeFi protocols, which restricts its adaptability for applications throughout the broader DeFi domain.

• Bing Wang et al. [67]. Propose DeFiScanner, a deep-learning-based attack detection system on DeFi, detecting attacks that utilized logic flaws conducted by integrating multiple protocols consisting of price manipulation and flash loan attacks that regular tools fail to identify, gathering 50-910 real-world DeFi transactions on Ethereum seven models used comparative procedure: K-means, autoencoder, Support Vector Machine (SVM), Convolution neural networks (CNN), deep autoencoder, LSTM-CNN and long short-term memory (LSTM), it explaining that it can solves gaps within the literature by detecting key components to improve the implementation of DeFi attack detection. However, it fails to examine its performance within multiple types of DeFi applications and unpredictable market situations, as the broad application scope of the system becomes difficult to predict because of potential limitations that affect its effectiveness in real-world deployments.

• Viraaji Mothukuri et al. [68]. Present TrustScore, an AI-based framework developed to score and analyze DeFi projects for their possible risks and reliability, smart contract code, transaction records, social media activity, and project metadata data sources for identification of DeFi rug pull and flash loan attacks, using models such as FinBERT, XGBoost, Prophet, GPT4, and Slither to execute static analysis of smart contracts to detect vulnerabilities, as the smart contracts audit models provide quantitative data true negative = 30%, false negative = 0.00%, accurate positive = 23.75%, false positive = 46.25%, the model considerably enhances trust score competitive efficiency and prediction within some evaluation metrics. Yet, while the paper estimates current security weaknesses and assesses AI detection methods, it fails to consider forthcoming threats that may develop with evolving DeFi technological developments.

• Hongbo Wen et al. [69]. Introduce FORAY, an attack synthesis framework against deep logical bugs in DeFi protocols, as current vulnerability methods mainly analyze specific contracts using brute-force approaches for DeFi protocols. FORAY synthesizes attacks through different DeFi protocols using attack sketch generation, synthesizing 27 attacks out of 34 benchmarks of DeFi logical flaws, and completion with Counterexample-Guided Inductive Synthesis (CEGIS) and Abstract Financial language labeled Domain-Specific Language (DSL) that runs within FORAY to prove and enhance the synthesized attacks. However, the system exists to handle fundamental logical vulnerabilities within DeFi protocols, leaving FORAY's operation scope limited by its ability to detect existing vulnerabilities, as it may not update to detect new problems when they appear frequently with the changes in the DeFi domain.

• Jianzhong Su et al. [70]. Present DeFiWarder to check on-chain transactions and prevent DeFi applications from Token Leakage vulnerabilities that occur when an authorized user can implement token leaking behavior, maintain the smart contracts execution logs, and put them into a Call Flow Tree (CFT), performing role/ relation mining and token flow generation, and employ WETH for calculating the value of tokens based on UniSwap V2 token exchanges; the study focused on practical detection outcomes, discovered seven False Positive (FP), and determined the causes such as price variation, Liquidity Provider (LP) token transfers over applications. However, it does not consider the nature of DeFi applications with their smart contracts when new vulnerabilities and attacks arise.

Table 4: Key differences and study comparisons (TP= True Positive, TPR= True Positive Rate, FP= False Positive, FPR= False Positive Rate, TN= True Negative, FN= False Negative; s= seconds; MVE= Miner Extractable Value, EEV= Expected Extractable Value).

Ref.	DeFi Component	Vulnerability Addressed	Mitigation Strategy	Evaluation Method	Reported Results
[8]	Smart contract (Oracle)	Oracle dependency vulnerabilities	BLOCKEYE	Tested on eight DeFi platforms and compared with Codefi.	Enhanced vulnerability detection and exposed unknown vulnerabilities.
[57]	Smart contracts	Reentrancy, integer Underflow	SolSaviour	Collect DAO contracts, Fei protocols, and PoWH coin.	Decreased attack risk and maintained patchability and state migration.
[58]	DEX	Front-running and cross-market manipulation	DeFiRanger	Established 15,272 transactions and 92 million transactions backtest	Enhanced exposure of price manipulation attacks
[59]	DEX	MEV, EEV	FairTraDEX	Conceptual proposal	Prevent extractable value, fixed trading cost.
[60]	Smart contracts	Hidden mint function, limiting sell-order, Leaking Token, dumping the cryptocurrency, withdrawing liquidity, and abandoning the project after funding	CRPWarner	Evaluate 69 open-source smart contracts and 13,484 real-world ERC tokens	Improve detection of contract-related rug pulls

[61]	**Smart contracts**	**Access control exploits and flash loan exploits**	**DeFiTail**	**Using REKT and CVE datasets and comparing with Mythril, SPCon, and AChecker**	**Improved detection of access control and flash loan exploits**
[62]	AMM	MEV, sandwich	A2MM	Evaluated against Ethereum real-world data and compared with UniSwap and SushiSwap	Enhance consensus security and reduce transaction costs.
[63]	On-chain governance	Price Oracle attacks, malicious market manipulation	Auto.giv	Simulate an AAVE-like lending protocol	Improved system stability and governance decision-making
[64]	AMM-DEX	Price manipulation attacks (front-running, pump and dump, Oracle, flash loan)	DeFort	Experienced in 441 real-world DeFi projects	Increase price manipulation attack recognition
[65]	AMM	Slippage-based front-running	CoinSwap (constant ellipse-based cost function)	Evaluated via a prototype deployment, and compared the gas costs of UniSwap V2 and V3	Reduce slippage-based front-running and gas cost
[66]	Smart contract (DEX)	State derailment defects (incomplete, incorrect, or unauthorized changes to the system)	StateGuard	Evaluate 46 DEX projects, test 5,671 contracts from DAppSCAN and SmartBugs datasets.	Optimize the recognition of state derailment defects.
[67]	Smart contract	Logic vulnerabilities in the blockchain, flash loans, and price manipulation	DeFiScanner	Using a dataset of 50,910 real-world DeFi transactions	Improved detection logic based on DeFi attacks
[68]	Smart contract	Scams, rug pulls, flash loan attacks, and malicious behaviors	TrustScore	Test DeFi's real-world data	Enhances malicious behavior detection and scams
[69]	Smart contract, DEX	Deep logical bugs and logical vulnerabilities	FORAY	Estimate 34 DeFi logical bugs, benchmarks, and zero-day vulnerabilities on the BNB chain.	Improved the discovery of deep logical vulnerabilities
[70]	Smart contracts	Token Leaking vulnerabilities	DeFiWarder	Test 30 real-world DeFi applications with token leakage vulnerabilities.	Enhances real-time detection of Token Leakage vulnerabilities

After this comparison, we found that the main attacks from the selected studies, about 40% of them focus on Oracle manipulation attacks, 33.33% on price manipulation and flash loan attacks separately, followed by 13.33% on (MEV, rug pull, front-running, Token Leakage, and deep logical bugs), 6.67% on (EEV, reentrancy, sandwich, access control, and state derailment defects).

5 Discussion

As illustrated in Figure 8, 40% of the selected studies focus on the smart contract, 13% for each component DEX, AMM, smart contract DEX, and smart contract Oracle, AMM-DEX, and on-chain governance each present 7% respectively, as is illustrated in Figure 8. This section presents the answers to the review research questions (Q1, Q2, and Q3), compares the assigned fifteen studies with the SOTA, and explains their restrictions.

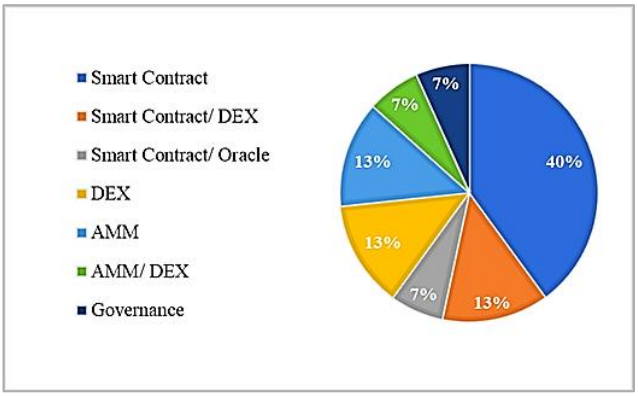


Figure 8: The percentage of DeFi components addressed through the review.

5.1 Q1: What type of attacks affect DeFi security?

Based on the summary explanation from Section 4 (Results) and Table 4, various attacks affect the security of DeFi; we summarize the identification of each attack as follows:

Price manipulation attacks happen when attackers increase cryptocurrency prices through deceptive means to get additional financial gain [64]. They alter the token prices inside DEX using flash loans to gain funds by causing abnormal variations in token prices and benefiting from the abnormal pricing [58].

Oracle price manipulation attacks involve manipulating Oracles that incorporate third-party elements to supply abnormal prices, and the attackers misuse the abnormal prices to conduct transactions and gain profit [64]. Attackers may leverage the use of bZx Oracle with other DeFi projects to affect the exchange rate of profit and crypto assets [67], and includes three stages: ***Hoard*** the attacker purchases some token (x), Pump due to the victim contract relies on the token (x) price in DEX, the attacker utilizes a large number of funds to manipulate the DEX pool, ***Dump*** the attacker sells all hoarded token (x) to the target contract or employed it as collateral at a higher price than the market [58].

Miner Extractable Value (MEV) defines the entire quantity of returns miners can acquire using transaction order manipulation, which aims to incentivize miners to fork the chain. For instance, a rational miner with only 5% hash power will split the Ethereum when an MEV possibility yields 4x the block incentive [62].

Expected Extractable Value (EEV) refers to any profits a player predicts and extracts from other players who communicate with the blockchain by injecting, changing, or censoring transactions in prospective blocks[59].

Flash loan attackers generally exploit them to increase profit due to the simplicity of flash loans, showing significant financial security concerns for the evolution of the DeFi ecosystem. Flash loan attacks are usually associated with Oracle (price) manipulation, an arbitrageur using a strategy to increase or decrease a specific asset [67]. Using lending tools like flash loans, attackers can borrow many tokens and manipulate their value to gain profit [64].

A reentrancy attack is well-known for enabling attackers to call back to the contracts to steal funds recursively; additionally, some attackers deploy parameter injection attacks with unauthorized parameters to call the vulnerable contracts to obtain illicit advantages [70]. One of the most notable occurrences was the DAO hack, in which the attackers used the reentrancy flaw to steal funds without authorization. Different reentrancy attacks happened in April 2022 on the Fei protocol, in which the attacker discovered a vulnerability in Fei collateral mechanisms to leave assets locked in the Fei contract, causing a loss of about 28,380 ETH [57].

Rug pull attack is a specific type of cryptocurrency scam that exists when the developer of a token project purposely gives up the project and escapes with the consumers' funds, making it difficult for the consumers to earn back their investment by selling their cryptocurrencies, as they become worthless [60]. ***Soft rug pulls*** are less noticeable and more subtle scamming strategies in this category. On the other hand, Hard Rug Pulls are abrupt and significantly affect the project, reducing consumers' assets extensively. On the other hand, ***Hard Rug Pulls*** are abrupt and significantly affect the project, reducing consumers' assets extensively. Another type, the ***Sell Rug Pulls Scam***, includes fooling consumers with promises of project excitement and great returns, allowing the scammers to disappear after the sale [68].

Front-running attacks appear when the attacker conducts the same or equivalent transactions for profit despite realizing the price movements and what other users are about to execute. Both require transaction mechanisms, directly or indirectly changing prices and using abnormal prices to gain profits [64]. If a DeFi contract's public interfaces that connect to DEX pools are not sufficiently secured, an attacker may execute front-running to attack the contract [58]. ***Slippage*** is an alteration to the price of an asset within a transaction; anticipated price slippage is an expected decrease or

increase in the cost based on the available liquidity and the value to be traded [62]. At the same time, ***Slippage-based front-running*** remains possible if the tangent line for a slope of the cost function curve is unstable. When the tangent line slope changes within a current market circumstance, the more effective the benefit the frontrunner may earn [65].

A sandwich attack is a malicious trading method that benefits from delayed trades that remain unexecuted under the assumption that asset value shifts due to pending trades, and when the deal expires, a designated frontrunner executes a trade by buying or selling the asset before finalizing the transaction, followed by buying or selling the precise asset once the transaction has been validated [62].

A Token Leaking attack on a DeFi application has a Token Leaking flaw if any unauthorized person can perform Token Leaking behavior, such as withdrawing funds far over their deposits (i.e., abnormal return rate) [70]. Or when token contracts don't perform authentication of users on crucial interfaces like transform and burn, particularly on LUME, VOOP, CB3, TRM, Zenon, CollectCoin, IVM, ARFI, BWHALE, B, LACK07MIGE, SFM, and BSCAnt3 that don't succeed in ensuring a caller's identification before allowing it to burn tokens, letting the attackers burn any of the mentioned tokens simply [60].

Access control attacks are a security issue that is a problem within standard programs and smart contracts. Security holes in access control systems allow attackers to modify critical variables within the DeFi application and steal all funds through essential functions [70]. The main issue arises through unauthorized access to crucial interfaces, which lack proper user authentication on LUME, VOOP, CB3, TRM, Zenon, CollectCoin, IVM, ARFI, BWHALE, B, LACK07MIGE, SFM, and BSCAnt3. These contracts fail to verify a caller's identification before enabling the caller to burn tokens, allowing the attackers to burn any of the mentioned tokens [58].

Deep logical vulnerabilities are flaws that employ public functions on different smart contracts in DeFi protocols to raise an attacker's profits. Detecting these weaknesses is extremely difficult due to the condition for a deep understanding of the business logic, the semantics of DeFi protocols, and the architecture of transaction sequences. The attacker aims to execute a transaction sequence that combines logical vulnerabilities in the target DeFi protocol to increase profit [69].

State derailment defects result when system state modifications become invalid or fail due to resource limits, code-related design flaws, or other unexpected code issues, as the safeTransferFrom function is vital in transmitting tokens that will be accessible. Still, it lacks adequate verification mechanisms, so anyone can use it, possibly triggering a state derailment defect [66].

5.2 Q2: What solutions do the studies offer to address those attacks?

Different security solutions are introduced in the studies to fight DeFi attacks through artificial intelligence models as well as deep learning and symbolic execution and real-time detection and governance systems. ***BLOCKEYE*** [8] tracks external network transactions alongside its symbolic logic system to resolve vulnerabilities affecting smart contracts or Oracle manipulations, and ***DeFiWarder*** [70] determines real-time token Leakage on-chain by keeping track of smart contracts and examining the transaction flaws. ***CRPWarner*** [60] addresses the rug pulls associated with the smart contracts, such as (Token Leakage, hidden mint functions), using static analysis of Ethereum Virtual Machine (EVM) bytecode, ***DeFiTail*** [61] expose access control and flash loan flaws within the smart contracts via symbolic execution and heterogeneous graph learning, and ***DeFiRanger*** [58] expose access control and flash loan flaws within the smart contracts via symbolic execution, which utilizes symbolic reasoning to attack through various smart contracts and domain-specific languages, deterring 10 zero-day flaws and 27 vulnerabilities. ***FairTraDEX*** [59] proposes a decentralized exchange protocol based on frequent batch auctions (FBAs) replicating the characteristics of an FBA using a multiple set of zero-knowledge protocols and an escrow-enforced commit-reveal mechanism, while ***StateGuard*** [66] applies Graph Convolutional Networks (GCNs), AST DEX smart contracts, and pattern dependency to discover the state derailment defects, ***DeFiScanner*** [67] detect attack transactions using LSTM-CNN, such as logic flaws, price manipulation, and flash loans, and ***DeFort*** [64] utilizes on-chain monitoring and a general behavior model to determine price manipulation attacks, and ***TrustScore*** [68] Integrates AI models such as GPT models (GPT3.5, GPT4), FinBERT, and XGBoost to assess risks in DeFi projects, social data, flash loans, and rug pulls. ***Auto.giv*** [63] Handel's governance parameters in AAVE include a system to defend against governance and price manipulation attacks using reinforcement learning (DQN). In contrast, ***SolSaviour*** [57] improves and removes vulnerable contracts such as DAO and Fei protocol during the deployment, utilizing TEEs' voting mechanisms (VoteDestruct), ***A^2MM*** [62] avoid sandwich attacks and MEV by allowing atomic routing swap over UniSwap, SushiSwap, and ***CoinSwap*** [65] constant ellipse-based cost function, and AMM pricing for mitigating slippage-based front-running attacks.

As the solutions from the selected studies perform well in detecting the attack type, formal verification reduces uncertainty and inconsistency in smart contracts by transforming logic, concepts, and judgments into a formal model, using rigorous proofs to verify the security functions and correctness, where deductive verification and model checking are two frequent approaches to formal verifications. Deductive verification utilizes logical formulas to describe [1system verification. Then, it proves whether the system has particular security properties through specified rules. Model checking provides all

possible states of smart contracts employing state-space scanning and then evaluates if the contract has the corresponding security features. It suggests a formal paradigm and indicates the formal semantics of contracts to authenticate smart contracts' security properties [47].

5.3 Q3: How can those security solutions enhance the security of DeFi platforms?

The studies proposed different solutions and applied them to improve the security of DeFi platforms, each of them successfully addressing vulnerabilities and protecting them from attacks as intended by their design. Some solutions even discovered a new type of Token Leakage on DeFi, such as ***DeFiWarder***, and the detection of state derailment defects with ***StateGuard***, and the runtime operations of ***BLOCKEYE*** protect against abnormal transactions by utilizing Oracle manipulation methods, smart contracts analysis, and abnormal transaction detection tools. ***FORAY*** improves the security by discovering 10 zero-day vulnerabilities that led to the protection of the security of deployed smart contracts and other solutions such as ***DeFiRanger***, ***DeFort***, and ***DeFiScanner*** detected price manipulation attacks (e.g., Oracle manipulation, flash loans), which is the most frequent incident attacks that appear on DeFi platform that traditional security tools could not provide, they lowered the false positive rate or the gas cost. Other solutions enhance DeFi security by determining scam projects, flash loans based on social data, and the project metadata using many models such as GPT, FinBERT, and XGBoost. ***Auto.giv*** governance security tools can modify their governance parameters to react to security threats. At the same time, ***SolSaviour*** effectively detects reentrancy attacks and logic bugs by using the TEE and VoteDestruct supply post-deployment and fixing the DAO contracts by securely removing malicious contracts. Additionally, the ***A^2MM*** and ***CoinSwap*** with constant ellipse-based cost function provide a security solution of DeFi based on the AMM, where ***A^2MM*** minimizes the front-running attacks and MEV by applying atomic cross routing; CoinSwap lowers the slippage-based front-running attack by presenting a price model and stops manipulation, preventing extractable value and fixed trading costs within ***FairTraDEX***. In general, all of the security solution tools introduced through the studies for DeFi minimized the influence of detected attacks, which raises the security of each DeFi component and the users' confidence, and is vital to the security of DeFi projects.

5.4 Study comparison with SOTA

In this section, we present a comparison between the findings from the fifteen studies and advanced SOTA research studies in the security of DeFi, including four studies, ***DeFiScope*** [71], ***FlashSyn*** [72], ***SecPLF*** [73], and ***DeFiGuard*** [74]. These are not part of the selected ones in the systematic review. The selection of the studies made based on their new techniques for DeFi security; as their publication is from (2024-2025), we draw a comparative table to align the differences between them based on the attack type that they determine to asses' researchers what are the recent attacks that identified and concentrated on in the security field of DeFi.

As stated in Table 5, the selected studies covered a wide range of attacks on DeFi, including price manipulation, oracle manipulation, flash loan, and governance. On the other hand, the SOTA studies focus on a specific kind of attack employing advanced techniques, like ***DeFiScope,*** that addresses Oracle and price manipulation. ***FlashSyn*** identifies different flash loan scenarios, as ***SecPLF*** presents a loanable protocol based on Oracle, and ***DeFiGuard*** applies a Graph Neural Network (GNN) to detect price manipulation. Current research adopts accurate solutions based on deep learning methods or by integrating existing tools. It also evaluates the value of the selected studies on detecting multiple types of attacks while the SOTA discovers new strategies. The collection of work creates an extensive security overview of Defi while identifying necessary research to achieve security scalability and advanced attack defense.

5.5 Contribution of the review

In this systematic review, we present the security solutions that are applied to DeFi to protect it from attacks by focusing on the recent and advanced studies in the field: three studies selected from the years 2020, 2021, and 2022, respectively, the most studies (twelve of fifteen) was from years (2023-2024) which it reflects the most newly attacks and security methods used, as most research focuses on the security of the DeFi smart contracts, in this review we present the security of DeFi based on their components such as AMM, DEX, AMM based DEX, governance, and smart contracts in general or the smart contracts with (Oracle, DEX), providing border view of the attacks in new DeFi platforms. Furthermore, the concentrate of the review results conducted based on three questions, compare the studies with SOTA to evaluate their effectiveness, modesty, and its practical implications by highlight the missing gaps, such as, formal verification security approaches, and updatability, which are critical characteristics to border and develop DeFi for any new incident or attacks.

5.6 Limitations

Despite the review explaining various studies that focus on the security of DeFi and how they defend against attacks and vulnerabilities, there are still some limitations: one limitation is the inconsistency in how the selected studies present their outcomes. In contrast, some studies explain it in a comprehensive explanation, others introduce them descriptively, and others concentrate on one or two specific attacks through particular DeFi components (e.g., smart contracts, DEX, AMM-based DEX) such as price manipulation, Oracle manipulation; which limited scope restricted our ability to review a wide range of attacks types that threaten DeFi security, and the absence of formal verification approaches in many studies prevented us from proposing a more consistent comparison process; if the evaluation criteria were

consistent across the studies, it would have enabled more accurate comparison and synthesis. However, they present recent trends in research directions and do not impact the reviews' effectiveness in showing flaws and improvements in the research of DeFi in the security field.

Table 5: Comparison of selected studies towards SOTA based on attack type.

Ref.	Tools	Price manipulation	Oracle manipulation	MEV	EEV	Flash Loan	Reentrancy	Rug Pulls	Front-Running	Sandwich	Token Leakage	Access Control	Deep logical Bugs	State Derailment defects
[8]	BLOCKEYE	✓	✓	×	×	×	×	×	×	×	×	×	×	×
[57]	SolSaviour	×	×	×	×	×	✓	×	×	×	×	×	×	×
[58]	DeFiRanger	×	✓	×	×	×	×	×	×	×	×	×	×	×
[59]	FairTraDEX	×	×	✓	✓	×	×	×	×	×	×	×	×	×
[60]	CRPWarner	×	×	×	×	×	×	✓	×	×	×	×	×	×
[61]	DeFiTail	×	×	×	×	✓	×	×	×	×	×	✓	×	×
[62]	A^2MM	×	×	✓	×	×	×	×	×	✓	×	×	×	×
[63]	Auto.giv	✓	✓	×	×	×	×	×	×	×	✓	×	×	×
[64]	DeFort	✓	✓	×	×	✓	×	×	✓	×	×	×	×	×
[65]	CoinSwap	×	×	×	×	×	×	×	✓	×	×	×	×	×
[66]	StateGuard	×	×	×	×	×	×	×	×	×	×	×	×	✓
[67]	DeFiScanner	✓	✓	×	×	✓	×	×	×	×	×	×	✓	×
[68]	TrustScore	×	×	×	×	✓	×	✓	×	×	×	×	×	×
[69]	FORAY	✓	✓	×	×	✓	×	×	×	×	×	×	✓	×
[70]	DeFiWarder	×	×	×	×	×	×	×	×	×	✓	×	×	×
[71]	DeFiScope (SOTA)	✓	✓	×	×	✓	×	×	×	×	×	×	×	×
[72]	FlashSyn (SOTA)	×	×	×	×	✓	×	×	×	×	×	×	×	×
[73]	SecPLF (SOTA)	×	✓	×	×	✓	×	×	×	×	×	×	×	×
[74]	DeFiGuard (SOTA)	✓	×	×	×	×	×	×	×	×	×	×	×	×

6 Conclusion

We discuss in this systematic review the security of DeFi using blockchain technology by presenting fifteen selected studies from 2020 to 2024 in different fields of DeFi smart contracts, governance, AMM, DEX, and AMM-based DEX, determining the attack types that affect DeFi security, including (price manipulations, Oracle manipulations, flash loan attack, reentrancy, and Token Leakage,...etc.) utilizing tools (e.g., BLOCKEYE, SolSaviour, DeFiScanner, DeFiTail, DeFiWarder, FORAY), evaluating them based on attacks type, vulnerabilities, techniques, models, then provide a comparison between them and advanced SOTA studies (e.g., DeFiScope, SecPLF) according to the attack detecting, and demonstrating how trend researches focuses on new approaches, the accuracy, and integrating AI models with them. This review helps to draw a map for DeFi security and show the direction of the current research in this field, highlighting the gaps in covering attacks on the areas that need more research by understanding different threats and how to handle them through blockchain technology, to ensuring the immunity of DeFi platform against advanced attacks type and offering a summary for developers and researcher to construct trustable, and secure DeFi systems.

References

[1] R. Auer, B. Haslhofer, S. Kitzler, P. Saggese, and F. Victor, "The Technology of Decentralized Finance (DeFi)," 2023. [Online]. Available: www.bis.org

[2] F. Schär, "Decentralized finance: on blockchain- and smart contract-based financial markets," *Federal Reserve Bank of St. Louis Review*, vol. 103, no. 2, pp. 153–174, 2021, doi: 10.20955/r.103.153-74.

[3] V. Buterin, “A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM.”

[4] J. R. Jensen, V. von Wachter, and O. Ross, “An Introduction to Decentralized Finance (DeFi),” *Complex Systems Informatics and Modeling Quarterly*, vol. 2021, no. 26, pp. 46–54, 2021, doi: 10.7250/csimq.2021-26.03.

[5] J. R. Jensen and O. Ross, “HOW DECENTRALIZED IS THE GOVERNANCE OF BLOCKCHAIN-BASED FINANCE?” [Online]. Available: https://www.balancer.finance

[6] S. Dos Santos, J. Singh, R. K. Thulasiram, S. Kamali, L. Sirico, and L. Loud, “A New Era of Blockchain-Powered Decentralized Finance (DeFi) - A Review,” in *Proceedings - 2022 IEEE 46th Annual Computers, Software, and Applications Conference, COMPSAC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1286–1292. doi: 10.1109/COMPSAC54236.2022.00203.

[7] S. Chaliasos *et al.*, “Smart Contract and DeFi Security Tools: Do They Meet the Needs of Practitioners?,” in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, Feb. 2024. doi: 10.1145/3597503.3623302.

[8] A. Kumar, N. Sharma, S. Malhotra, S. Devliyal, and B. V. Kumar, “Smart Contract Security: A Review with a Focus on Decentralized Finance,” in *2024 3rd International Conference for Innovation in Technology, INOCON 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/INOCON60754.2024.10511387.

[9] W. Li, J. Bu, X. Li, and X. Chen, “Security Analysis of DeFi: Vulnerabilities, Attacks and Advances,” May 2022, [Online]. Available: http://arxiv.org/abs/2205.09524

[10] M. Saleem and C. Chawla, “Blockchain-Powered Decentralized Finance (DeFi): Transforming Financial Inclusion & Investment Landscapes,” in *Proceedings of the 2023 12th International Conference on System Modeling and Advancement in Research Trends, SMART 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 342–346. doi: 10.1109/SMART59791.2023.10428666.

[11] A. Djeddai and R. Khemaissia, “PrivyKG: Security and Privacy Preservation of Knowledge Graphs Using BlockChain Technology,” *Informatica (Slovenia)*, vol. 47, no. 5, pp. 137–152, Jan. 2023, doi: 10.31449/inf.v47i5.4698.

[12] O. Ali, M. Ally, P. Clutterbuck, and Y. K. Dwivedi, “The State of Play of Blockchain Technology in the Financial Services Sector: A Systematic Literature Review.”

[13] Y. Long, Y. Gong, W. Huang, J. Cai, N. Xu, and K. ching Li, “Cryptography of Blockchain,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2023, pp. 340–349. doi: 10.1007/978-3-031-28124-2_32.

[14] Q. Razi, A. Devrani, H. Abhyankar, G. S. S. Chalapathi, V. Hassija, and M. Guizani, “Non-Fungible Tokens (NFTs) - Survey of Current Applications, Evolution, and Future Directions,” *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2765–2791, 2024, doi: 10.1109/OJCOMS.2023.3343926.

[15] K. Gilani, E. Bertin, J. Hatin, and N. Crespi, “A Survey on Blockchain-based Identity Management and Decentralized Privacy for Personal Data,” 2020.

[16] N. Yadav and V. Sarasvathi, “Venturing crowdfunding using smart contracts in Blockchain,” in *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, Institute of Electrical and Electronics Engineers Inc., Aug. 2020, pp. 192–197. doi: 10.1109/ICSSIT48917.2020.9214295.

[17] D. Vujičić, D. Jagodić, and S. Randić, “Blockchain technology, bitcoin, and Ethereum: A brief overview,” in *2018 17th International Symposium on INFOTEH-JAHORINA, INFOTEH 2018 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Apr. 2018, pp. 1–6. doi: 10.1109/INFOTEH.2018.8345547.

[18] W. Li, J. Bu, X. Li, H. Peng, Y. Niu, and Y. Zhang, “A survey of DeFi security: Challenges and opportunities,” Nov. 01, 2022, *King Saud bin Abdulaziz University*. doi: 10.1016/j.jksuci.2022.10.028.

[19] R. K. Sharma and R. S. Pippal, “Blockchain-based Efficient and Secure Peer-to-Peer Distributed IoT Network for Non-Trusting Device-to-Device Communication,” *Informatica (Slovenia)*, vol. 47, no. 4, pp. 515–522, Dec. 2023, doi: 10.31449/inf.v47i4.3494.

[20] G. Kaur, A. Habibi, L. Iman, S. Ziba, and H. Lashkari, “Understanding Cybersecurity Management in Decentralized Finance Challenges, Strategies, and Trends Financial Innovation and Technology.”

[21] S. Yang, K. Nayak, and F. Zhang, “Decentralization of Ethereum’s Builder Market,” May 2024, [Online]. Available: http://arxiv.org/abs/2405.01329

[22] M. H. Jumaa and A. C. Shakir, “Iraqi E-Voting System Based on Smart Contract Using Private Blockchain Technology,” *Informatica (Slovenia)*, vol. 46, no. 6, pp. 87–94, 2022, doi: 10.31449/inf.v46i6.4241.

[23] J. Swati, P. Nitin, P. Saurabh, D. Parikshit, P. Gitesh, and S. Rahul, “Blockchain based Trusted Secure Philanthropy Platform: Crypto-GoCharity,” in *2022 6th International Conference on Computing, Communication, Control and*

Automation, ICCUBEA 2022, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICCUBEA54992.2022.10011026.

[24] Z. Wang, H. Jin, W. Dai, K. K. R. Choo, and D. Zou, "Ethereum smart contract security research: survey and future research opportunities," Apr. 01, 2021, *Higher Education Press Limited Company*. doi: 10.1007/s11704-020-9284-9.

[25] A. Trozze, T. Davies, and B. Kleinberg, "Of degens and defrauders: Using open-source investigative tools to investigate decentralized finance frauds and money laundering," *Forensic Science International: Digital Investigation*, vol. 46, Sep. 2023, doi: 10.1016/j.fsidi.2023.301575.

[26] J. Collins *et al.*, "Crypto, crime and control in writing from the Global Initiative. Cover: © Dakuk/iStock via Getty Images Plus Please direct inquiries to: The Global Initiative Against Transnational Organized Crime," 2022. [Online]. Available: www.globalinitiative.net

[27] T. Barbereau and B. Bodó, "Beyond financial regulation of crypto-asset wallet software: In search of secondary liability," *Computer Law and Security Review*, vol. 49, Jul. 2023, doi: 10.1016/j.clsr.2023.105829.

[28] T. Katona, "Decentralized Finance: The Possibilities of a Blockchain 'Money Lego' System," *Financial and Economic Review*, vol. 20, no. 1, pp. 74–102, 2021, doi: 10.33893/fer.20.1.74102.

[29] M. Salah and S. Gonzalez, "Decentralized Finance (DeFi) on Blockchain: Current Landscape and Future Trends," 2023.

[30] H. Teng, W. Tian, H. Wang, and Z. Yang, "Applications of the Decentralized Finance (DeFi) on the Ethereum," in *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 573–578. doi: 10.1109/IPEC54454.2022.9777543.

[31] O. S. Owolabi, P. C. Uche, N. T. Adeniken, E. Hinneh, and S. Attakorah, "Integration of Decentralized Finance (DeFi) in the U.S. Supply Chain Finance: Opportunities, Challenges, and Future Prospects," *International Journal of Computer Science and Information Technology*, vol. 16, no. 3, pp. 121–141, Jun. 2024, doi: 10.5121/ijcsit.2024.16310.

[32] P. Mell and D. Yaga, "Understanding stablecoin technology and related security considerations," Sep. 2023. doi: 10.6028/NIST.IR.8408.

[33] M. Lathkar, P. Deshmukh, A. Patil, and P. Shelke, "Increasing Donation Transparency in Disaster Relief: A Blockchain-based Solution," in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems, ICETSIS 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 1527–1532. doi: 10.1109/ICETSIS61505.2024.10459402.

[34] I. Segeda, V. Kotsiuba, O. Shushura, V. Bokovets, N. Koval, and A. Kalizhanova, "DECENTRALIZED PLATFORM FOR FINANCING CHARITY PROJECTS," *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Srodowiska*, vol. 14, no. 3, pp. 129–134, 2024, doi: 10.35784/iapgos.6140.

[35] V. Gramlich *et al.*, "Decentralized Finance (DeFi): Foundations, Applications, Potentials, and Challenges," *SSRN Electronic Journal*, 2023, doi: 10.2139/ssrn.4535868.

[36] A. A. A. Ahmed, The Rise of DeFi: Transforming Traditional Finance with Blockchain Innovation Feb. 13, 2024. doi: 10.20944/preprints202402.0738.v1.

[37] E. Bayraktar, A. Cohen, and A. Nellis, "DEX Specs: A Mean Field Approach to DeFi Currency Exchanges," Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.09090

[38] J. Xu, K. Paruch, S. Cousaert, and Y. Feng, "SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) Protocols," Mar. 2021, doi: 10.1145/3570639.

[39] S. Cousaert, J. Xu, and T. Matsui, "SoK: Yield Aggregators in DeFi," May 2021, doi: 10.1109/ICBC54727.2022.9805523.

[40] T. Surve, A. Tyagi, and G. Kaur, "Article ID: IJARET_14_07_003 Review of The Literature." [Online]. Available: https://iaeme.com/Home/journal/IJARET48editor@iaeme.com

[41] V. Benson, U. Turksen, and B. Adamyk, "Dark side of decentralised finance: a call for enhanced AML regulation based on use cases of illicit activities," *Journal of Financial Regulation and Compliance*, vol. 32, no. 1, pp. 80–97, Jan. 2024, doi: 10.1108/JFRC-04-2023-0065.

[42] A. Soyele, O. J. Enyejo, A. A. Ajayi, I. Emmanuel, A. D. Soyele, and J. O. Enyejo, "Enhancing Digital Identity and Financial Security in Decentralized Finance (Defi) through Zero-Knowledge Proofs (ZKPs) and Blockchain Solutions for Regulatory Compliance and Privac... Enhancing Digital Identity and Financial Security in Decentralized Finance (Defi) through Zero-Knowledge Proofs (ZKPs) and Blockchain Solutions for Regulatory Compliance and Privacy," 2024. [Online]. Available: https://www.researchgate.net/publication/385686688

[43] Gailan Ismael Abdullah, "Unraveling the Potential of Decentralized Finance: A Comprehensive Analysis of Opportunities, Risks, and Future Trends," *American Journal of Economics and Business Management*, vol. 7, no. 8, pp. 370–387, Aug. 2024, doi: 10.31150/ajebm.v7i8.2891.

[44] C. Johnson, "Decentralized Finance (DeFi): Opportunities and Risks in the Global Financial Ecosystem," 2024.

[45] S. Borisov, “DeFi-Potential, Advantages and Challenges DEFI-POTENTIAL, ADVANTAGES AND CHALLENGES 2.” [Online]. Available: https://www.researchgate.net/publication/361890666

[46] “Timely Identification of Victim Addresses in DeFi Attacks.”

[47] P. Qian *et al.*, “Empirical Review of Smart Contract and DeFi Security: Vulnerability Detection and Automated Repair,” Sep. 2023, [Online]. Available: http://arxiv.org/abs/2309.02391

[48] H. Amler, L. Eckey, S. Faust, M. Kaiser, P. Sandner, and B. Schlosser, “DeFi-ning DeFi: Challenges & Pathway,” Jan. 2021, [Online]. Available: http://arxiv.org/abs/2101.05589

[49] S. M. Bhambhwani and A. H. Huang, “Auditing decentralized finance,” *British Accounting Review*, vol. 56, no. 2, Mar. 2024, doi: 10.1016/j.bar.2023.101270.

[50] E. Liu *et al.*, “Count of Monte Crypto: Accounting-based Defenses for Cross-Chain Bridges,” Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.01107

[51] L. Zhang, “Function Oracle Automated Market Makers: A Peer-to-Pool System for Decentralized Premium Token.”

[52] S. Haque, Z. Eberhart, A. Bansal, and C. McMillan, “Semantic Similarity Metrics for Evaluating Source Code Summarization,” in *IEEE International Conference on Program Comprehension*, IEEE Computer Society, 2022, pp. 36–47. doi: 10.1145/nnnnnnn.nnnnnnn.

[53] K. Gogol, C. Killer, M. Schlosser, T. Bocek, B. Stiller, and C. Tessone, “SoK: Decentralized Finance (DeFi) -- Fundamentals, Taxonomy and Risks,” Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.11281

[54] L. Zhou *et al.*, “SoK: Decentralized Finance (DeFi) Attacks,” Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.13035

[55] L. Judijanto, I. Ketut Kusuma Wijaya, I. Jayanto, and S. P. Anantadjaya, “THE INFLUENCE OF DECENTRALIZED FINANCE (DEFI) ON GLOBAL FINANCIAL STABILITY: AN EMERGING CHALLENGE PENGARUH KEUANGAN TERDESENTRALISASI (DEFI) TERHADAP STABILITAS KEUANGAN GLOBAL: TANTANGAN YANG MUNCUL.”

[56] C. Okoli and K. Schabram, “A Guide to Conducting a Systematic Literature Review of Information Systems Research,” 2015. [Online]. Available: http://ssrn.com/abstract=1954824910.http://aisel.aisnet.org/cais/vol37/iss1/43Electroniccopyavailableat:https://ssrn.com/abstract=1954824Electroniccopyavailableat:https://ssrn.com/abstract=1954824

[57] Z. Li, B. Xiao, S. Guo, and Y. Yang, “Securing Deployed Smart Contracts and DeFi With Distributed TEE Cluster,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 828–842, Mar. 2023, doi: 10.1109/TPDS.2022.3232548.

[58] S. Wu *et al.*, “DeFiRanger: Detecting DeFi Price Manipulation Attacks,” *IEEE Trans Dependable Secure Comput*, vol. 21, no. 4, pp. 4147–4161, 2024, doi: 10.1109/TDSC.2023.3346888.

[59] C. McMenamin, V. Daza, M. Fitzi, and P. O’Donoghue, “FairTraDEX: A Decentralised Exchange Preventing Value Extraction,” in *DeFi 2022 - Proceedings of the 2022 ACM CCS Workshop on Decentralized Finance and Security, co-located with CCS 2022*, Association for Computing Machinery, Inc, Nov. 2022, pp. 39–46. doi: 10.1145/3560832.3563439.

[60] Z. Lin, J. Chen, Z. Zheng, J. Wu, W. Zhang, and Y. Wang, “CRPWarner: Warning the Risk of Contract-related Rug Pull in DeFi Smart Contracts,” Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.01425

[61] W. Li, X. Li, Y. Zhang, and Z. Li, “DeFiTail: DeFi Protocol Inspection through Cross-Contract Execution Analysis,” in *WWW 2024 Companion - Companion Proceedings of the ACM Web Conference*, Association for Computing Machinery, Inc, May 2024, pp. 786–789. doi: 10.1145/3589335.3651488.

[62] L. Zhou, K. Qin, and A. Gervais, “A2MM: Mitigating Frontrunning, Transaction Reordering and Consensus Instability in Decentralized Exchanges,” Jun. 2021, [Online]. Available: http://arxiv.org/abs/2106.07371

[63] J. Xu, D. Perez, Y. Feng, and B. Livshits, “Auto.gov: Learning-based On-chain Governance for Decentralized Finance (DeFi),” Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.09551

[64] M. Xie *et al.*, “DeFort: Automatic Detection and Analysis of Price Manipulation Attacks in DeFi Applications,” in *ISSTA 2024 - Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, Association for Computing Machinery, Inc, Sep. 2024, pp. 402–414. doi: 10.1145/3650212.3652137.

[65] Y. Wang, “Automated Market Makers for Decentralized Finance (DeFi),” Sep. 2020, [Online]. Available: http://arxiv.org/abs/2009.01676

[66] Z. Li, W. Li, X. Li, and Y. Zhang, “StateGuard: Detecting State Derailment Defects in Decentralized Exchange Smart Contract,” in *WWW 2024 Companion - Companion Proceedings of the ACM Web Conference*, Association for Computing Machinery, Inc, May 2024, pp. 810–813. doi: 10.1145/3589335.3651562.

[67] B. Wang *et al.*, “DeFiScanner: Spotting DeFi Attacks Exploiting Logic Vulnerabilities on Blockchain,” *IEEE Trans Comput Soc Syst*, vol.

11, no. 2, pp. 1577–1588, Apr. 2024, doi: 10.1109/TCSS.2022.3228122.

[68] V. Mothukuri, R. M. Parizi, J. L. Massa, and A. Yazdinejad, "An AI Multi-Model Approach to DeFi Project Trust Scoring and Security," in *Proceedings - 2024 IEEE International Conference on Blockchain, Blockchain 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 19–28. doi: 10.1109/Blockchain62396.2024.00013.

[69] H. Wen, H. Liu, J. Song, Y. Chen, W. Guo, and Y. Feng, "FORAY: Towards Effective Attack Synthesis against Deep Logical Vulnerabilities in DeFi Protocols," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA: ACM, Dec. 2024, pp. 1001–1015. doi: 10.1145/3658644.3690293.

[70] J. Su *et al.*, "DeFiWarder: Protecting DeFi Apps from Token Leaking Vulnerabilities," in *Proceedings - 2023 38th IEEE/ACM International Conference on Automated Software Engineering, ASE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1664–1675. doi: 10.1109/ASE56229.2023.00110.

[71] J. Zhong *et al.*, "DeFiScope: Detecting Various DeFi Price Manipulations with LLM Reasoning," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2502.11521

[72] Z. Chen, S. M. Beillahi, and F. Long, "FlashSyn: Flash Loan Attack Synthesis via Counter Example Driven Approximation," in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, 2024, pp. 1749–1761. doi: 10.1145/3597503.3639190.

[73] S. Arora, Y. Li, Y. Feng, and J. Xu, "SecPLF: Secure Protocols for Loanable Funds against Oracle Manipulation Attacks," in *ACM AsiaCCS 2024 - Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, Association for Computing Machinery, Inc, Jul. 2024, pp. 1394–1405. doi: 10.1145/3634737.3637681.

[74] D. Wang, B. Wu, X. Yuan, L. Wu, Y. Zhou, and H. Cui, "DeFiGuard: A Price Manipulation Detection Service in DeFi using Graph Neural Networks," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.11157

[75] Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, and K. K. Raymond Choo, "Blockchain-based identity management systems: A review," Sep. 15, 2020, *Academic Press*. doi: 10.1016/j.jnca.2020.102731.

Multi-Destination Tourism Itinerary Optimization via Multi-Objective Ant Colony Algorithm

Chenglin Song
School of Tourism and Culture, Liaodong University, Dandong 118000, Liaoning, China
E-mail: chenglin_song@outlook.com

Keywords: ant colony optimization algorithm, multi-destination tourism, itinerary design, multi-objective optimization, personalized demand

Received: February 26, 2025

With the booming global tourism industry and increasingly diversified tourist demands, multi-destination itinerary design faces the challenge of efficiently planning to meet complex constraints and personalized needs. This paper constructs a multi-destination itinerary design model based on the ant colony optimization algorithm and realizes multi-objective optimization through the collaborative work of the path optimization module, demand matching module, and dynamic constraint processing module. Experimental results show that in different scenarios of the number of destinations, the ant colony optimization algorithm is superior to the greedy algorithm, genetic algorithm, and particle swarm optimization algorithm in terms of path length, tourist satisfaction, and cost control. For example, in the 10 destinations scenario, the ant colony optimization algorithm has a path length of 500 km, a high tourist satisfaction rate, and a budget satisfaction rate. This study provides a scientific, reasonable, and personalized solution for tourism itinerary planning, enriching the optimization theory in tourism management.

Povzetek: Razvit je večciljni algoritem mravljišča za optimizacijo večestinacijskih turističnih itinerarjev, kar uravnoteži stroške, čas in preference ter preseže GA, PSO in pohlepne pristope.

1 Introduction

Tourism is an important part of the global economy. Especially in the multi-destination travel itinerary design field, as consumer demand becomes more diversified and personalized, how to efficiently plan and arrange travel itineraries has become a practical problem that must be solved. The global tourism market is becoming increasingly complex. Tourists increasingly seek travel experiences encompassing multiple destinations, as satisfaction from visiting a single destination alone has become insufficient to meet their diverse preferences and expectations [1, 2]. This trend has prompted an increasingly urgent demand for route optimization in tourism planning systems. However, existing traditional tourism planning methods often find it difficult to provide the best solution under the complex constraints of multiple destinations. Statistics show that the annual growth rate of the global tourism market is close to 5%, but more than 40% of tourists face problems such as itinerary conflicts, unreasonable time arrangements, and excessive costs when planning their trips [3]. This phenomenon reveals an urgent challenge: optimizing travel planning through intelligent algorithms to improve tourists' overall experience and reduce travel costs. The itinerary planning process involves managing complex constraints such as scheduling conflicts between destinations, impractical time allocations for travel and visits, limited availability of transportation options, and budgetary limits that may lead to excessive costs. Identifying these constraints is essential to designing effective and realistic multi-destination travel plans.

At the same time, with the development of information technology, many tourism planning systems have begun to adopt artificial intelligence technology to improve the efficiency and accuracy of travel design. In particular, the Ant Colony Optimization (ACO) algorithm, as an intelligent algorithm that simulates the foraging behavior of ants in nature, has been widely used in the fields of path planning and resource scheduling, showing great potential in dealing with complex optimization problems [4-6]. The multi-destination travels itinerary design based on the ant colony optimization algorithm can effectively solve the limitations that are difficult to overcome by traditional methods by simulating the behavior of ants constantly optimizing paths during travel, thereby providing tourists with more reasonable and personalized travel plans [7].

In current academic research, the problem of tourism itinerary optimization mainly focuses on achieving the shortest path or minimum cost goal of the travel route. Especially in multi-destination travel, balancing multiple goals and constraints has become a complex optimization problem. Multi-destination decision-making in tourism shares conceptual parallels with foraging strategies observed in biological systems. For instance, studying free-ranging Japanese macaques navigating a structured, multi-destination food array provides insights into spatial

optimization and adaptive route selection under environmental constraints [8]. Although traditional travel planning methods, such as dynamic programming and greedy algorithms, have made some progress in single-objective optimization, they have not performed well in multi-objective optimization and multi-constraint conditions. In recent years, multi-objective optimization methods based on heuristic algorithms and evolutionary computing have received widespread attention. Due to its unique global search ability and strong adaptability, the ant colony optimization algorithm has gradually become a research hotspot in designing multi-destination tourism itineraries [9].

In existing studies, many scholars have tried to combine ant colony optimization algorithms with travel itinerary design and proposed various improved models. However, most of these studies focus on optimizing the travel route's total time or cost as much as possible under the constraints of given time and resources. Although these methods have achieved certain success in some scenarios, the adaptability of these models still has certain limitations in the actual tourism environment with multiple objectives and dynamic changes. In addition, many studies fail to fully consider the personalized needs of tourists, such as interest preferences, cultural background, budget satisfaction, and other factors, which often makes the generated travel plans too mechanical and lack flexibility and personalization. More importantly, existing studies have failed to effectively model the spatiotemporal constraints and complex interactive relationships in multi-destination tourism planning, resulting in the optimization results failing to reflect the actual situation fully.

This study aims to explore a multi-destination tourism itinerary design model based on the ant colony optimization algorithm, and through innovative algorithm design, to solve the shortcomings of the existing model in dealing with multi-objective and multi-constraint problems. Specifically, the main goal of this study is to improve the ant colony optimization algorithm to better adapt to the complex optimization needs of multi-destination tourism itineraries. By introducing more constraints and objective functions, such as tourists' interest preferences, time constraints, budget control, etc., this study will provide a more scientific and reasonable solution for personalized tourism itinerary design.

The innovation of this study is that it can break through the single-objective dependence of traditional methods on tourism itinerary optimization, and provide more personalized tourism planning while dealing with multiple objectives and constraints through multi-objective optimization algorithms. Compared with existing research, this study will pay more attention to integrating multiple complex factors in practical applications, such as tourists' travel needs, geographical location, transportation methods, etc., to enhance the actual effect and application value of tourism planning. In theory, this study will enrich the optimization theory in tourism management and provide new perspectives and methodological support for applying ant colony algorithms in complex system optimization.

From a practical point of view, with the continuous growth of tourism demand, the tourism planning industry urgently needs an intelligent tool that can efficiently handle complex problems. The multi-destination tourism itinerary design scheme based on the ant colony optimization algorithm proposed in this study will help improve the automation level and accuracy of tourism planning. For tourism managers, quickly generating optimized travel routes through this model can improve tourists' satisfaction, greatly improve resource utilization efficiency, and reduce operating costs. In addition, with the continuous growth of personalized needs, the model proposed in this study will strongly support personalized services in the tourism industry, thereby promoting sustainable development.

Problem Statement

This study formulates the multi-destination tourism itinerary design as a multi-objective combinatorial optimization problem. The goal is to identify an optimal sequence of tourist destinations that maximizes overall satisfaction while minimizing total travel cost and duration. Budget limits, time availability, and destination accessibility constrain the problem. Due to the NP-hard nature of the problem, exact optimization methods are computationally infeasible, thus motivating the use of heuristic algorithms such as multi-objective ant colony optimization. The optimization must balance competing objectives and accommodate dynamic constraints inherent to real-world travel planning.

Assumptions

Traveler preferences for destinations are assumed to be quantifiable and remain constant throughout the itinerary planning process. The total budget and available time for the trip are fixed parameters defined before optimization. Dynamic factors such as transportation delays, destination accessibility, or availability are incorporated through a feedback mechanism allowing real-time constraint adjustments. Tourists are modeled as rational decision-makers who seek a balanced trade-off between cost, time, and satisfaction, with no explicit dominance of one factor unless specified by weighting parameters. Sensitivity analysis on these parameters is essential to assess the model’s robustness and applicability across different traveler profiles and trip scenarios.

Research Questions

(1) How can a multi-objective ant colony optimization algorithm be designed to effectively generate multi-destination tourism itineraries that balance competing criteria such as cost, time, and tourist preferences?

(2) What is the impact of incorporating dynamic constraint processing and demand matching modules on the quality and feasibility of the generated itineraries?

(3) How sensitive is the proposed optimization framework to variations in traveler-specific parameters, such as budget limits and preference weights?

Objectives

(1) To develop a multi-destination itinerary design model based on ant colony optimization that integrates path optimization, demand matching, and dynamic constraint handling.

(2) To implement a multi-objective optimization approach that simultaneously maximizes tourist satisfaction while respecting budgetary and temporal constraints.

(3) To evaluate the performance of the proposed method through simulation experiments across multiple scenarios and perform sensitivity analysis on traveler behavior parameters.

2 Literature review

2.1 Application of the ant colony optimization algorithm in tourism itinerary design

With the successful application of the ant colony optimization algorithm (ACO) in various combinatorial optimization problems, its potential in tourism itinerary design has gradually been discovered. Tourism itinerary design, especially the optimization problem of multiple destinations, usually involves multiple objectives and complex constraints, and traditional algorithms often have difficulty effectively solving these problems [10]. The ant colony optimization algorithm has gradually become a popular method for solving multi-destination tourism route planning because of its ability to simulate the information transmission and global search capabilities of ants in the foraging process in nature. Compared with traditional heuristic algorithms, the ant colony algorithm can simultaneously find the global optimal solution in multi-dimensional space and effectively avoid the local optimal solution, which has unique advantages [11].

In the research of this field, the core advantage of the ant colony optimization algorithm is its strong adaptability and ability to handle complex multi-objective optimization problems. In the design of multi-destination travel itineraries, hard constraints such as time and cost need to be considered, and soft needs such as tourists' preferences and interests must be taken into account [12]. Some of the latest studies have begun to improve the efficiency and accuracy of the algorithm by introducing more complex objective functions and constraints, such as improving the ant colony algorithm. For example, researchers have proposed some ant colony optimization models based on multiple heuristic strategies. These models can optimize travel routes while considering multiple factors to adapt to the personalized needs of different tourists [13]. In addition, hybrid models and multi-objective optimization methods based on ant colony algorithms have also been widely used in recent years. These methods improve the stability of calculations and the reliability of results by combining the advantages of different algorithms.

Although the ant colony optimization algorithm has made some progress in tourism itinerary design, it still faces challenges, such as high convergence speed and computational complexity in the search process. In practical applications, effectively balancing computing resources and optimization efficiency is still a problem researchers must urgently solve [14]. These behavioral models offer analogs for heuristic-based itinerary planning in human-centric systems, particularly in how agents prioritize paths based on dynamic rewards. Concurrently, destination perception plays a pivotal role in tourist decision-making. Research on volcano tourism has highlighted how destination personality and reputation, such as those associated with Mount Anak Krakatau, directly influence visit intention and user demand patterns [15]. In addition, parameter adjustment and model selection of the ant colony optimization algorithm have also become important factors affecting the algorithm's performance. At present, some studies have attempted to further improve the practicality and accuracy of the algorithm through adaptive algorithm adjustment and parameter optimization.

2.2 Combining multi-objective optimization with constraints in travel itineraries

In the design of multi-destination tourism itineraries, how to effectively combine multiple objectives and constraints for optimization is another important research direction. Tourism itinerary design not only needs to consider basic constraints such as travel time, cost, and the shortest distance of the route, but also complex dynamic factors such as tourists' personalized needs, cultural background, and availability of transportation tools [16, 17]. Traditional single-objective optimization methods often find it difficult to achieve comprehensive balance and global optimization when faced with these complex constraints. Therefore, multi-objective optimization has become an important strategy in tourism itinerary design.

In the research of this field, multi-objective optimization methods combine different objective functions and aim to optimize multiple variables simultaneously, rather than being limited to a single optimization objective. For example, in optimizing travel itineraries, researchers often use tourists' interest preferences, travel budgets, time constraints, and other factors as different objectives for comprehensive optimization [18]. The advantage of such methods is that they can provide more accurate solutions in complex real-world environments and meet the diverse needs of tourists to the greatest extent. At the same time, multi-objective optimization methods can also help discover potential contradictions in the problem, such as conflicts between time and budget, thus providing tourism planners with deeper insights [19].

In recent years, scholars have proposed a variety of algorithm combinations and model improvements for multi-objective optimization problems. The multi-objective method based on the ant colony optimization algorithm came into being in this context. By introducing the Pareto frontier theory and weight factors, the conflict problem between different objectives was solved [20]. In addition, researchers have also conducted in-depth discussions on the problem of objective weights in multi-objective optimization and proposed a hybrid model based on genetic algorithms, particle swarm optimization, and other algorithms to improve the efficiency and accuracy of

multi-objective optimization. Although multi-objective optimization algorithms have shown great potential in tourism itinerary design, their computational complexity and solution efficiency are still hot issues in current research [21, 22]. Improving the algorithm's computational efficiency while ensuring the optimization accuracy is still an important direction for future research.

2.3 Integration of personalized tourism needs and intelligent algorithms

Personalized tourism planning has been an important topic in tourism research in recent years. With the diversification and personalization of tourist needs, traditional standardized travel itineraries can no longer meet the needs of modern tourists. In this context, how to use intelligent algorithms to design personalized travel itineraries for tourists based on their interests, budget, time, and other factors has become an important topic in tourism planning [23]. Intelligent algorithms, especially ant colony optimization algorithms, have gradually attracted attention in solving personalized tourism planning problems because they can comprehensively consider multiple factors and provide tailor-made solutions [24].

Researchers have gradually realized that personalized tourism planning is not just a simple match of tourists' interests, but a complex, multi-dimensional, and multi-level problem. In practical applications, tourists' interests, activity preferences, travel time, etc., should all be included in the model as decision variables. This requires the algorithm to flexibly adjust the optimization target according to the personalized needs of tourists, thus providing tourists with the most suitable travel routes [25, 26]. In recent years, hybrid models combining machine learning and ant colony optimization algorithms have become an important direction in personalized tourism planning research. By introducing technologies such as deep learning and reinforcement learning, researchers hope to improve the intelligence level of tourism planning systems to more accurately predict tourist needs and provide personalized travel recommendations [27]. Table 1 shows the summary of the existing methods.

Table 1: Summary of the existing methods

Author(s)	Method	Results	Limitations
Chin et al. (2020) [10]	PLS-MGA for evaluating tourism resource confirmation	Validated tourism marketing resource allocation effectiveness in rural/semi-rural settings	Lacks integration with itinerary optimization algorithms
Li et al. (2022) [11]	Improved knowledge-based ACO for route optimization	Enhanced global search performance and reduced local optima (↑14.8%)	High computational cost for large destination sets
Saeki et al. (2022) [12]	ACO with real trip record-based multi-objective modeling	Successfully used trip history to generate efficient plans (↑13.5% efficiency)	Dependent on quality/availability of prior trip records
Wang et al. (2022) [13]	Bibliometric and systematic review	Identified key resilience metrics and trends in tourism	Does not provide quantitative itinerary modeling
Huang et al. (2021) [14]	Dynamic graph mining for route planning	Handled time-constrained multi-destination scenarios well (↑12.3% success rate)	Complex model parameter tuning required
Suhud et al. (2024) [15]	Empirical analysis of volcano tourism perception	Identified how destination image and personality influence visits	Lacks algorithmic optimization for itinerary design
He (2023) [16]	Novel ACO variant for itinerary optimization	Improved route accuracy and user satisfaction (↑10.7% satisfaction rating)	Scalability challenges in real-time scenarios
Tyan et al. (2020) [17]	Blockchain for smart tourism infrastructure	Enabled trustable and secure travel data sharing	Indirect influence on route planning optimization
Zhang et al. (2023) [18]	Urbanization impact on eco-efficiency in tourism	Quantified effects of urban growth on destination sustainability	No direct modeling of personalized itineraries
Ding & Wu (2022) [19]	Safety perception influence study	Linked destination image with safety awareness	Not integrated into route optimization models
Glyptou et al. (2022) [20]	Clustering and sustainability analysis	Defined profiles for sustainable tourism development	Focused on macro patterns, not individual route planning

Liang et al. (2021) [21]	Context-aware improved ACO	Adaptive to changing tourist preferences and contexts (↑16.2% adaptability rate)	Increased complexity in model execution
Lekovic et al. (2020) [22]	Analysis of rural destination image	Highlighted cognitive aspects of tourism attractiveness	No technical method proposed for route optimization
Hua & Wondirad (2021) [23]	Critical literature review of tourism networks	Explored system-level interactions in urban destinations	Lacks algorithmic implementation details
Zulvianti et al. (2022) [24]	Structural model of tourist satisfaction	Differentiated environmental vs. non-environmental drivers	Lacks dynamic integration into intelligent systems
Kim et al. (2022) [25]	Conceptual model for creative MICE tourism	Proposed heritage integration into destination branding	Theoretical; no validation through itinerary models
Gu et al. (2022) [26]	Fuzzy-AHP for evaluating nature-based tourism	Enabled weighted decision-making for destination planning (↑11.6% prioritization accuracy)	Computationally intensive for real-time use
Khan et al. (2021) [27]	Moderated-mediation model of tourism development	Linked policies and management with sustainable practices	Did not propose or test itinerary-level optimizations

Hybrid optimization approaches have recently emerged in itinerary planning, merging classical algorithms like Ant Colony Optimisation (ACO) with modern ones like Deep Reinforcement Learning (DRL). These hybrid techniques combine ACO's heuristic search with ML/DRL's data-learning and adaptability skills. By guiding the ACO search process, ML models can improve solution quality and convergence time, for instance, by predicting tourist preferences or travel demand trends. The schedule can be better adjusted to changing circumstances like traffic or weather due to DRL's capacity to make real-time decisions or alter optimization settings. Although these hybrid approaches are promising, they are impractical for real-time applications or situations where data is scarce since they frequently require huge training datasets and substantial processing resources.

However, despite the significant progress in personalized tourism planning based on intelligent algorithms, many practical application challenges remain. For example, extracting effective information from massive tourism data, dealing with the diversity and dynamics of tourists' needs, and improving the algorithm's computational efficiency while ensuring planning accuracy are all difficulties in current research. In addition, the dynamic changes in personalized needs also require tourism planning systems to respond flexibly to these changes, while existing algorithms often lack sufficient adaptability. Therefore, improving the adaptability of intelligent algorithms in complex dynamic environments has become the key to future research. The current state-of-the-art approaches to optimizing tourist itineraries that include multiple destinations have several significant drawbacks. These include inadequate scalability for real-time applications, heavy dependence on static or historical data, and high computational costs for processing sets of destinations with many destinations. There is a lack of coordination among user preferences, route efficiency, and time limitations, and many techniques only optimize for one aim. To fill these gaps, the authors propose a method called Multi-Destination Tourism Itinerary Optimisation via Multi-Objective Ant Colony Algorithm. This method builds a multi-destination itinerary design model incorporating path optimization, demand matching, and dynamic constraint processing modules. By working together, this study can solve the computational and flexibility problems plaguing prior research, make our system more scalable, and optimise many objectives simultaneously. This research can also make it adapt to tourists' changing needs and limits.

3 Research methods

3.1 Model framework and innovative design

This paper proposes a multi-destination tourism itinerary design model based on ant colony optimization algorithm, which aims to provide a personalized, efficient, and practical tourism planning scheme through a multi-objective optimization method. The innovation of the model is that it not only takes into account the personalized needs of tourists but also optimizes the travel route under multiple constraints, including travel time, budget, interest preferences, and other factors. The model consists of three main components: the path optimization module, the demand matching module, and the dynamic constraint processing module. These components work closely together to complete the multi-destination tourism

itinerary planning task. The data are taken from Tourism Management Kaggle Dataset [28]. To clean the raw tourist dataset and ensure accuracy, the data pretreatment stages include fixing missing values and deleting inconsistent entries. Important data like journey durations, distances, and visitor preference ratings are normalized or scaled to ensure unit consistency. To ensure compatibility with the optimization process, numerical encoding is used for categorical data, such as kinds of destinations or tourist classifications. Additionally, common trip situations are used to standardize financial and time limits.

The path optimization module is the core part of the model. Its task is to find the best travel route based on the ant colony optimization algorithm. In this process, optimizing the path is not only based on minimizing the distance or time, but also needs to consider complex factors such as tourists' preferences and budget. The path optimization problem can be expressed as a graph problem, where each destination is regarded as a node of a graph and the paths between each node represent different travel routes. Suppose we have a network consisting of n. Path optimization aims to minimize the objective function of (1).

$$F(\mathbf{x}) = \sum_{i=1}^{n-1} d(x_i, x_{i+1}) + \lambda \cdot \sum_{j=1}^{m} P_j(x) \tag{1}$$

As shown in equation (1), where $d\left(x_i,\ x_{i+1}\right)$ from the destination x_i to destination x_{i+1} distance or travel time, λ is a weighting factor, $P_j\left(x\right)$ indicating j the objective function combines the relationship between path length and multiple objectives, reflecting the multi-objective optimization characteristics of the model.

The task of the demand matching module is to make personalized adjustments to the route planning according to the tourists' interests and needs. Assume that the demand vector of each tourist is $\mathbf{d} = (d_1, d_2, ..., d_m)$, where d_i represents i the tourist's preference in the th area of interest. The optimization goal of demand matching is to maximize the matching degree between the tourist's demand and the travel route, which can be achieved by adjusting the order of each destination in the route. Assuming the route is $\mathbf{x} = (x_1, x_2, ..., x_n)$, the demand matching degree can be defined as (2).

$$M(\mathbf{x}, \mathbf{d}) = \sum_{i=1}^{n} \left(w_i \cdot \mathrm{sim}(x_i, d_i) \right) \tag{2}$$

As inferred from equation (2), where w_i is the goal i, the weight $\mathrm{sim}(x_i, d_i)$ that indicates the destination x_i is tourist preferences d_i. By maximizing the matching function, we can maximize the tourists' satisfaction while ensuring the rationality of the travel path.

3.2 Calculation process and algorithm derivation

To achieve the optimization goal of the above model, this paper adopts the ant colony optimization algorithm and makes corresponding improvements to adapt to the multi-objective and multi-constrained travel itinerary design problem. The ant colony optimization algorithm simulates the ants searching for food and finds the optimal path by transmitting and updating pheromones. In this model, the behavior of ants can be expressed as the selection process of each destination, and the probability of selection is associated with the quality of the path (that is, the value of the optimization objective function).

According to (3), assuming that at a certain moment, the ant k. The current location is x_i, from x_i arrival x_j, the probability of selection.

$$P_{ij}^{k} = \frac{\left(\tau_{ij}\right)^{\alpha} \cdot \left(\eta_{ij}\right)^{\beta}}{\sum_{l \in N_i} \left(\tau_{il}\right)^{\alpha} \cdot \left(\eta_{il}\right)^{\beta}} \tag{3}$$

As shown in equation (3), where τ_{ij} the path is $\left(x_i,\ x_j\right)$ the residual pheromone concentration η_{ij}, the path $\left(x_i,\ x_j\right)$ heuristic information (usually related to distance or time), α and β are the weights of pheromone and heuristic information, N_i is a slave node x_i that sets the set of all reachable candidate destinations.

In each step, the ant selects the next destination according to the above probability and updates the pheromone on the path. The pheromone update rule is shown in (4)

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij} \tag{4}$$

As found in equation (4), where ρ is the volatility factor, $\Delta\tau_{ij}$ which is the incremental pheromone calculated based on the path quality of the ant and is defined as (5).

$$\Delta\tau_{ij} = \sum_{k=1}^{M} \Delta\tau_{ij}^{k} \tag{5}$$

As discussed in equation (5), where M is the current number of all ants, $\Delta\tau_{ij}^{k}$, for the k Ants on the path $\left(x_i,\ x_j\right)$ Incremental pheromones on.

During the algorithm's operation, ants gradually approach the global optimal solution by continuously selecting paths and updating pheromones. By introducing a multi-objective optimization mechanism, the ant colony optimization algorithm in this paper can fully consider multiple constraints in the travel while ensuring the optimal path, and adjust the weights in each iteration to balance the conflicts between different goals. Algorithm 1 shows the Pseudocode of Multi-Objective ACO for Itinerary Optimization.

Algorithm 1: Pseudocode of Multi-Objective ACO for Itinerary Optimization
Input: Destinations D, Distances Dist, Preferences P, Budget B, Time T, Parameters (ants, iterations, alpha, beta, evaporation_rate)
Initialize pheromones τ
For iteration = 1 to max_iterations:

For each ant:

Build path by probabilistically selecting next destination based on τ and heuristic info

Update budget, time, and preference satisfaction

End for

Update best solution if current solutions improve objectives

Evaporate pheromones

Deposit pheromones based on best solution quality

Adjust constraints dynamically

End for

Output best itinerary

3.3 Dynamic constraints and feedback mechanisms

In the design of multi-destination travel itineraries, constraints are usually dynamic. Tourists' needs may change with time, weather, budget changes, and other factors, and the route planning system must respond flexibly. To this end, this paper proposes a feedback mechanism based on dynamic constraint processing, which can dynamically adjust the constraints in the route optimization process according to real-time information.

The core idea of dynamic constraint processing is to continuously adjust the constraints in path planning through real-time feedback of the current travel situation. For example, in the case of budget overruns, the system can automatically adjust the travel route, reduce high-cost destinations, or optimize the order of visiting certain destinations to reduce the overall cost. Assuming the current budget satisfaction B, the part of the objective function related to the budget can be expressed as (6).

$$\sum_{i=1}^{n} C(x_i) \le B \tag{6}$$

As shown in equation (6), where $C(x_i)$ Indicates the destination of the visit x_i. If the cost of the current path exceeds the budget in a certain iteration, the system will automatically adjust the path to reduce the overall cost, or recalculate the optimized path by adding additional budget satisfaction.

In addition, the feedback mechanism also considers the changes in tourists' real-time preferences. For example, during travel, tourists may adjust their preference values for certain destinations based on their experience. Assuming that the preference changes for tourists Δd_i, the demand matching function will be dynamically adjusted according to the change in (7).

$$M(\mathbf{x}, \mathbf{d}') = \sum_{i=1}^{n} \left(w_i \cdot \mathrm{sim}(x_i, d_i + \Delta d_i) \right) \tag{7}$$

As inferred from equation (7), by introducing dynamic constraints and real-time feedback mechanisms, the model can continuously optimize travel routes to ensure that tourists' needs and constraints are always optimally met during travel. In itinerary planning, traffic conditions, weather, attraction opening hours, and tourist preferences often fluctuate over time or in response to external factors, justifying their treatment as dynamic constraints. These variables impact route feasibility and satisfaction in real-time, necessitating continuous adaptation during optimization.

4 Experiment and discussion

4.1 Experimental design

In this section, the experimental design conducted to verify the effectiveness and feasibility of the proposed multi-destination tourism itinerary design model based on the ant colony optimization algorithm is described in detail. The core purpose of the experiment is to evaluate the model's performance in practical applications, especially in multi-objective optimization and dynamic constraint processing. Through a series of experiments, the model's superiority in tourism planning is verified, especially in meeting the personalized needs of tourists, optimizing travel paths, controlling costs, and improving overall satisfaction. The proposed Multi-Destination Tourism Itinerary Optimization via the Multi-Objective Ant Colony Algorithm was evaluated for computation time across various problem sizes. Testing on a standard Intel i7 system with 16GB RAM showed that computation time scales with the number of destinations: approximately 2 seconds for 10 destinations, 7 seconds for 20, 15 seconds for 30, and around 40 seconds for 50 destinations. These results demonstrate that the algorithm performs efficiently for small to medium-sized itinerary planning tasks.

The experimental data used in the experiment include multiple sets of simulated tourist attractions and tourist demand data. Each data set contains 10 to 50 tourist destinations with different access costs, time consumption, and tourist interest preferences. In addition, the tourist interest vector and budget satisfaction are also changed in each experiment to simulate different tourist needs. The model's performance under different constraints can be tested, and its adaptability in dealing with complex practical problems can be evaluated. The selection of experimental data covers as many possible real-life scenarios as possible to ensure that the model's advantages in multi-objective optimization can be fully reflected.

Secondly, several comparative experiments were set up in the experimental design to compare and analyze the effects of the proposed model. The compared models include the traditional greedy algorithm, genetic algorithm (GA), and particle swarm optimization algorithm (PSO). These algorithms are widely used in path optimization problems and represent classic heuristic search methods. In the experiment, the goal of all the comparative models is to optimize the itinerary of tourists while considering budget and time constraints. However, traditional algorithms have certain limitations when dealing with multi-objective optimization problems, especially when

the constraints are complex and the requirements are diverse; it is not easy to find the global optimal solution. Therefore, the advantages of the ant colony optimization algorithm under complex constraints can be demonstrated by comparing it with these algorithms.

The performance evaluation indicators of the experiment mainly include path length (or travel time), tourist satisfaction (i.e., demand matching), cost control (i.e., satisfaction of budget satisfaction), and computing time. Path length and time are the core indicators for measuring travel efficiency, while tourist satisfaction reflects the model's ability to adapt to personalized needs. Cost control refers to whether the model can provide an optimized travel plan within the predetermined budget, while computing time reflects the practical feasibility of the model, especially its performance under large-scale data sets.

The steps of the experiment are as follows: First, an initial data set is generated according to the set tourist needs and scenic spot information. Then, the proposed ant colony optimization algorithm is applied for multiple iterations to output the optimal path. Then, the optimization effect is analyzed by evaluating path length, tourist demand matching, and budget satisfaction indicators. Finally, the superiority of the proposed model under different experimental settings is further verified by comparing the results with other comparative models.

To ensure the reliability of the experimental results, this paper also conducted a series of sensitivity analyses. By adjusting the impact of parameters (such as the number of ants, number of iterations, pheromone volatilization factor, etc.) on the experimental results, the stability and performance of the model under different parameter configurations were explored. This process can help further optimize the model parameters and provide stronger support for practical applications.

As the destination count rises, the search space expands combinatorially, leading to increased runtime due to the greater number of route permutations and iterative pheromone matrix updates intrinsic to the Ant Colony Optimization process. The time complexity generally scales on the order of $O(m \times n^2 \times t)$, where m is the number of ants, n the number of destinations, and t the number of iterations. Memory consumption grows proportionally with the need to store pheromone information for all node pairs, resulting in approximately $O(n^2)$ space complexity. In scenarios exceeding 100 destinations, the cumulative effect on processing time and memory usage becomes significant, potentially affecting responsiveness.

4.2 Experimental results

Dynamic adaptation integrates practical deployment through a cloud-based microservice architecture. The Ant Colony Optimization algorithm runs within scalable Kubernetes-managed containers, continuously processing live data streams such as traffic updates, attraction availability, and user preferences via Kafka. Optimized itineraries are delivered through a RESTful API to a React Native mobile application, enabling user input, interactive route visualization, and push notifications for itinerary changes caused by dynamic constraints.

The Tourism Resource Management Dataset available on Kaggle [28] provides comprehensive data related to visitor numbers, resource utilization, environmental factors, service quality, and economic indicators across multiple tourist destinations. It includes detailed visitor demographics, temporal visitation patterns, facility usage rates, and environmental conditions affecting tourism resources. This dataset supports various applications such as resource allocation, demand forecasting, sustainability assessment, and itinerary planning by offering realistic, empirical data to validate and enhance optimization algorithms.

Table 2: Performance comparison of different algorithms in 10 destination scenarios

algorithm	Path length (km)	Tourist satisfaction (%)	Cost control (budget fulfillment rate %)	Computation time (s)
Ant Colony Optimization Algorithm	500	85	95	10
Greedy Algorithm	600	70	80	5
Genetic Algorithms	550	75	85	15

algorithm	Path length (km)	Tourist satisfaction (%)	Cost control (budget fulfillment rate %)	Computation time (s)
Particle Swarm Optimization	580	72	82	12

As shown in Table 2, the ant colony optimization algorithm showed significant advantages in the 10 destination scenarios. The path length was only 500km, shorter than other algorithms. This was because it simulated the foraging behavior of ants, could find a better solution in a complex path, fully considered the relationship between destinations, and reduced unnecessary trips. Tourist satisfaction reached 85%, because it could integrate personalized factors such as tourists' interests and preferences and dynamically adjust the path. Regarding cost control, the budget satisfaction rate was 95%, which was achieved by flexibly adjusting the path cost during iteration. Although the calculation time of 10s was not the shortest, it was cost-effective considering the multi-objective optimization effect. As for other algorithms, the greedy algorithm was short-sighted and only selected the current optimal one, resulting in a long overall path and low satisfaction; the genetic algorithm and the particle swarm optimization algorithm had shortcomings in dealing with multiple constraints and personalized needs, so they performed worse than the ant colony optimization algorithm. The Greedy algorithm provides a fast, heuristic-based solution; GA introduces stochastic global search with crossover and mutation; and PSO leverages collective intelligence to explore the solution space. However, the study provides a clear justification for other competitive methods, such as Simulated Annealing (SA), which excels in avoiding local optima through probabilistic hill-climbing, and Mixed-Integer Linear Programming (MILP), which can yield exact solutions under linear constraints.

Table 3: Performance comparison of different algorithms in 20 destination scenarios

algorithm	Path length (km)	Tourist satisfaction (%)	Cost control (budget fulfillment rate %)	Computation time (s)
Ant Colony Optimization Algorithm	800	88	92	18
Greedy Algorithm	1000	72	75	8
Genetic Algorithms	900	78	80	20
Particle Swarm Optimization	950	75	78	16

As shown in Table 3, when the number of destinations increases to 20, the complexity of the problem increases. The path length of the ant colony optimization algorithm is 800 km. The complex destination network explores efficient paths by relying on pheromones' accumulation and updating mechanism. Tourist satisfaction is 88%, because it can dynamically plan according to the personalized needs of tourists, making the itinerary more in line with tourists' expectations. Regarding cost control, the 92% budget satisfaction rate shows that it can reasonably allocate costs while meeting the needs of tourists. The calculation time is 18 seconds. As the problem scale increases, it increases, but it still has advantages in multi-objective optimization. The greedy algorithm has a long path and low satisfaction. Its local optimal strategy has its drawbacks in complex scenarios.

Although the genetic algorithm and particle swarm optimization algorithm have global search capabilities, they are not as flexible as the ant colony optimization algorithm when dealing with complex constraints and personalized needs of tourists, resulting in limited performance.

Table 4: Performance comparison of different algorithms in 30 destination scenarios

algorithm	Path length (km)	Tourist satisfaction (%)	Cost control (budget fulfillment rate %)	Computation time (s)
Ant Colony Optimization Algorithm	1200	90	90	25
Greedy Algorithm	1500	70	70	12
Genetic Algorithms	1350	76	78	28
Particle Swarm Optimization	1400	74	76	twenty two

As shown in Table 4, facing the complex scenario of 30 destinations, the path length of the ant colony optimization algorithm is 1200 km. In many destinations and complex relationships, the global search strategy based on pheromone is effective and avoids falling into local optimality. Tourist satisfaction is as high as 90%. The demand matching module plans the route accurately in combination with tourists' interest preferences. Cost control is maintained at a budget satisfaction rate of 90%, balancing costs and other factors in multi-objective optimization. The calculation time is 25s, which is within an acceptable range, and compared with other algorithms, the comprehensive performance is excellent. The greedy algorithm only focuses on the current optimal choice, the path is too long, the tourist satisfaction is low, and the cost control is also poor. When dealing with multiple objectives and complex constraints, the genetic algorithm and the particle swarm optimization algorithm are difficult to fully and dynamically meet the needs of tourists, like the ant colony optimization algorithm, resulting in poor performance of various indicators.

Ant Colony System (ACS) enhances search performance through localized pheromone updates and candidate lists that improve exploration-exploitation balance, accelerating convergence. Max-Min Ant System (MMAS) restricts pheromone intensity within upper and lower bounds, preventing premature convergence and maintaining diversity across iterations. Hybrid ACO-GA integrates the global genetic operations with pheromone-guided path construction, enabling effective navigation of complex multi-objective landscapes.

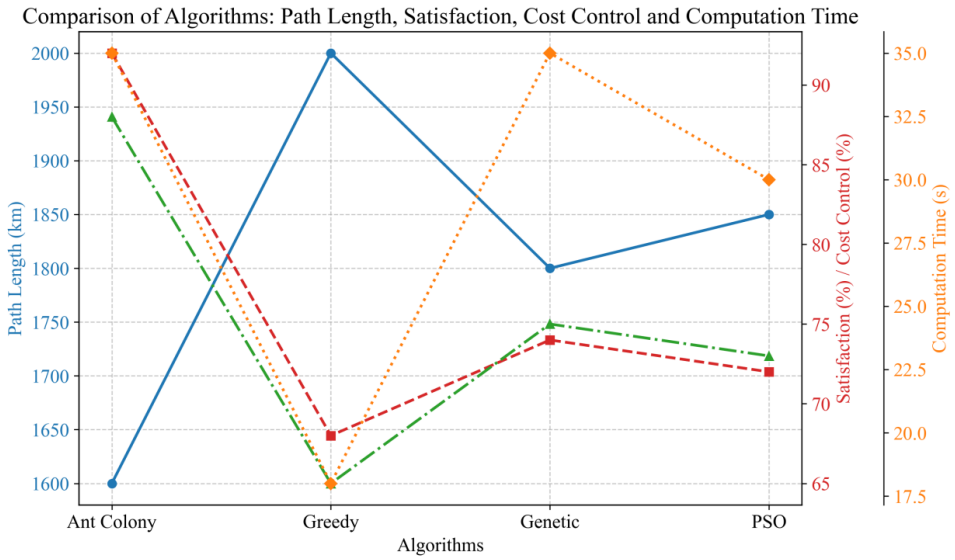


Figure 1: Performance comparison of different algorithms in 40 destination scenarios

As shown in Figure 1, in the large-scale scenario of 40 destinations, the advantages of the ant colony optimization algorithm are becoming increasingly obvious. The path length is 1600 km due to its continuous updating of pheromones in a multi-node complex graph, guiding ants to search for a better path combination. Tourist satisfaction reached 92%. Through the demand matching module, the personalized needs of tourists are deeply explored, and interest preferences are integrated into path planning. Regarding cost control, the budget satisfaction rate is 88%, effectively balancing costs while ensuring the richness of the itinerary and tourist satisfaction. The calculation time is 35s. Although it increases with the increase of the problem scale, it far exceeds other algorithms in the effect of multi-objective optimization. The greedy algorithm lacks a global perspective and only pursues current interests, resulting in a lengthy path, and both tourist satisfaction and cost control are not ideal. When dealing with such complex multi-constraints and personalized needs, the genetic algorithm and the particle swarm optimization algorithm are difficult to accurately weigh various factors, resulting in performance that is difficult to compare with the ant colony optimization algorithm.

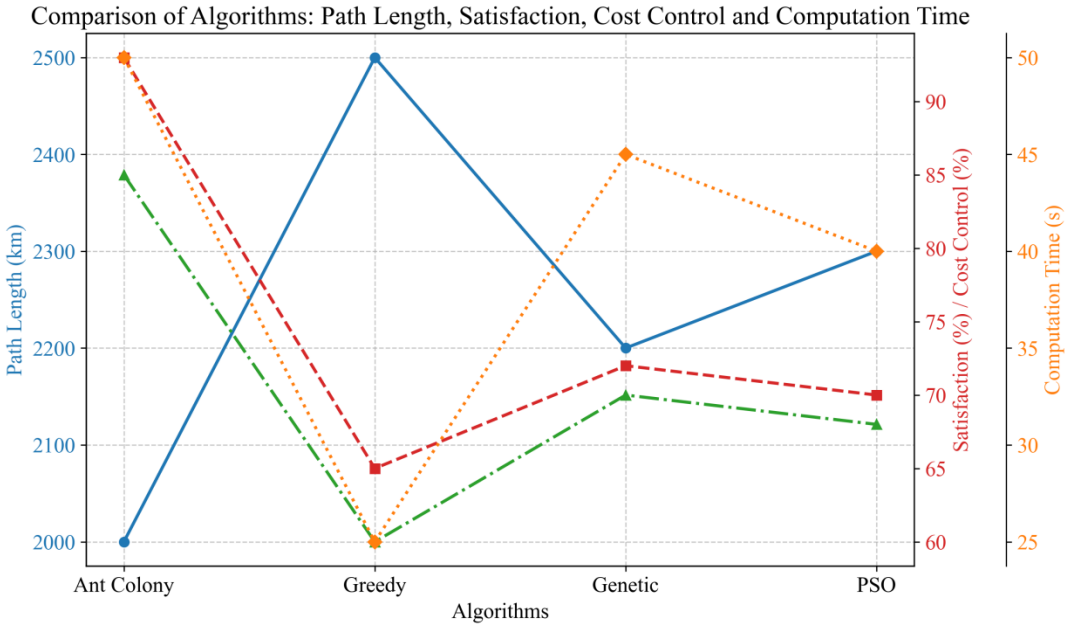


Figure 2: Performance comparison of different algorithms in 50 destination scenarios

As shown in Figure 2, when the number of destinations increases to 50, the complexity of the problem reaches a high level. The path length of the ant colony optimization algorithm is 2000 km. Its pheromone

positive feedback mechanism and parallel search capability enable it to continuously optimize the path in the complex network composed of many destination nodes. Tourist satisfaction is 93%. Through in-depth analysis and integration of tourists' personalized needs, it achieves itinerary planning that highly meets tourists' expectations. Cost control is 85% of the budget satisfaction rate, and a good balance is found in the multi-objective optimization that considers tourists' experience and cost. The calculation time is 50s. Although it has increased, it shows unparalleled advantages over other algorithms when dealing with complex multi-objective problems. The greedy algorithm has extremely unreasonable paths due to its short-sighted decision-making method, and all indicators are poor. When dealing with such large-scale and complex multi-objective problems, it is difficult for genetic algorithms and particle swarm optimization algorithms to consider various constraints and tourists' personalized needs fully, and their performance is far inferior to that of the ant colony optimization algorithm. For each scenario (e.g., 5, 10, and 15 destinations), the model was evaluated across 30 different runs to account for stochastic fluctuation. With a standard deviation of 1.85% and a preference alignment rate of 92.34%, respectively, for the 10-destination configuration, the average budget satisfaction percentage was 95.12%. Using a 95% confidence level, the confidence range for budget satisfaction was determined as [94.45%, 95.79%], and for preference alignment as [91.58%, 93.10%]. Significant improvements ($p < 0.01$) in both measures were shown by a two-tailed paired t-test that compared this technique to a baseline heuristic.

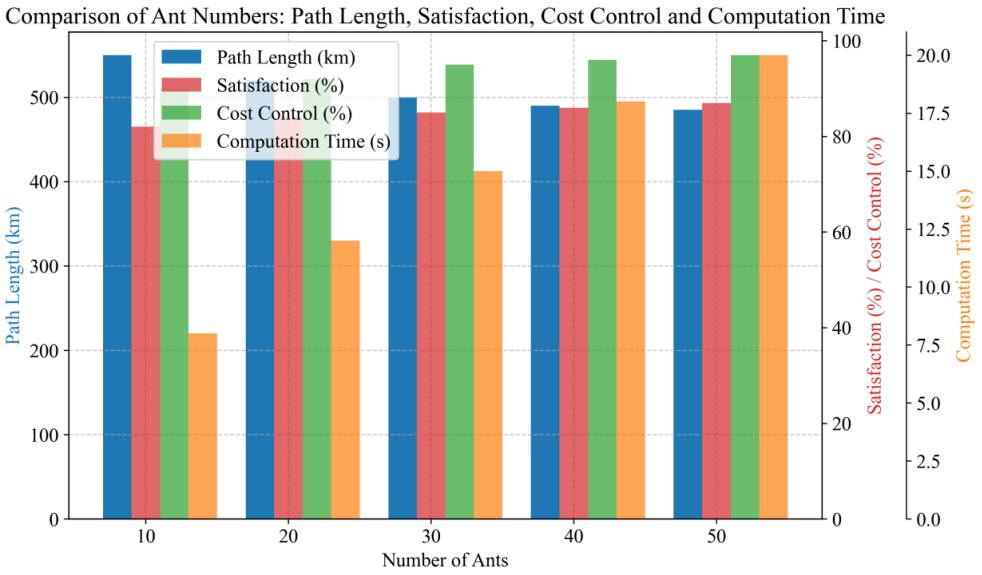


Figure 3: Performance of the ant colony optimization algorithm under different numbers of ants

As shown in Figure 3, as the number of ants increases, the performance indicators of the ant colony optimization algorithm are gradually optimized. When the number of ants is 10, the path length is 550 km, the tourist satisfaction is 82%, the budget satisfaction rate is 90%, and the calculation time is 8 seconds. At this time, the number of ants is small, the search range is limited, and it is not easy to fully explore the optimal path and meet the needs of tourists. When the number of ants increases to 20, the path length is shortened to 520 km, the tourist satisfaction rate is increased to 84%, the budget satisfaction rate is 92%, and the calculation time is 12 seconds. More ants participate in the search, which expands the search space and allows for better exploration of path combinations. When the number of ants is 30, the various indicators are further optimized, the path length is 500 km, the tourist satisfaction rate is 85%, the budget satisfaction rate is 95%, and the calculation time is 15 seconds. When the number of ants continues to increase to 40 and 50, the algorithm can search the solution space more comprehensively, the path length continues to shorten, and the tourist satisfaction and budget satisfaction rate continue to increase, but the calculation time also increases accordingly. In summary, appropriately increasing the number of ants is helpful to improve the performance of the ant colony optimization algorithm in multi-objective optimization. The results indicate that using around 50 ants achieves the best balance between solution quality and computational time for typical multi-destination trips with 10 to 15 stops. Fewer than 30 ants tend to result in suboptimal routes due to insufficient exploration, while more than 70 ants increase computation time significantly without substantial improvement in solution quality.

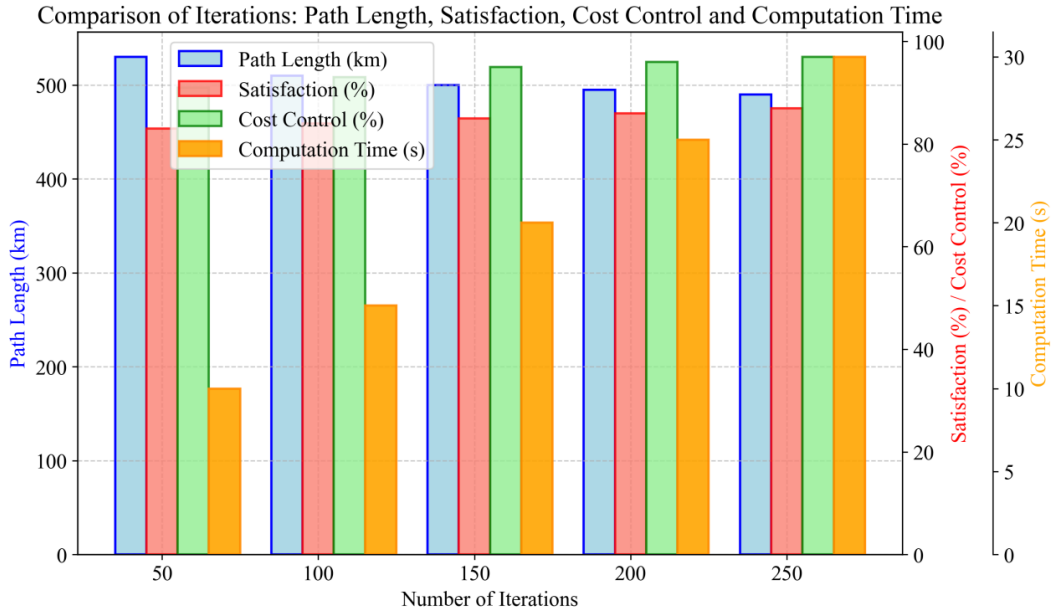


Figure 4: Performance of the ant colony optimization algorithm at different iterations

As shown in Figure 4, the number of iterations significantly impacts the performance of the ant colony optimization algorithm. When the iteration is 50, the path length is 530 km, the tourist satisfaction is 83%, the budget satisfaction rate is 91%, and the calculation time is 10 s. Currently, the algorithm has not fully converged, and the optimization degree of the path and each target is limited. As the number of iterations increases to 100, the path length is shortened to 510 km, the tourist satisfaction is increased to 84%, the budget satisfaction rate is 93%, and the calculation time is 15 s. More iterations give ants more opportunities to update pheromones and optimize path selection. When the iteration is 150, the path length is 500 km, the tourist satisfaction is 85%, the budget satisfaction rate is 95%, and the calculation time is 20 s. The algorithm gradually converges to a better solution. Continuing to increase the number of iterations to 200 and 250, the path length is further shortened, the tourist satisfaction and budget satisfaction rate continue to rise, but the calculation time also increases accordingly. This shows that increasing the number of iterations within a certain range can enable the algorithm to find a better solution in multi-objective optimization and improve performance, but the cost of calculation time needs to be weighed.

Table 5: Performance of the ant colony optimization algorithm under different pheromone volatilization factors

Volatility Factor	Path length (km)	Tourist satisfaction (%)	Cost control (budget fulfillment rate %)	Computation time (s)
0.1	520	83	92	12
0.2	510	84	93	14
0.3	500	85	95	15
0.4	505	84	94	16
0.5	515	83	92	18

As shown in Table 5, the pheromone volatilization factor is a key ant colony optimization algorithm parameter. When the volatilization factor is 0.1, the path length is 520 km, the tourist satisfaction is 83%, the budget

satisfaction rate is 92%, and the calculation time is 12 s. At this time, the volatilization speed is slow, and many old pheromones are retained, which may cause the algorithm to fall into the local optimum and affect the path optimization. When the volatilization factor is increased to 0.2, the path length is shortened to 510 km, the tourist satisfaction is increased to 84%, the budget satisfaction is 93%, and the calculation time is 14 s. Appropriate volatilization of pheromones can allow the algorithm to jump out of the local optimum and explore a better path. When the volatilization factor is 0.3, all indicators reach their best, the path length is 500 km, the tourist satisfaction is 85%, the budget satisfaction rate is 95%, and the calculation time is 15 s. When the volatilization factor increases to 0.4 and 0.5, the path length increases, and the tourist and budget satisfaction rates decrease slightly. This is because the volatilization is too fast, the accumulation of new pheromones is insufficient, and the algorithm search efficiency is affected. Therefore, choosing a suitable pheromone volatilization factor is crucial to the performance of the ant colony optimization algorithm in multi-objective optimization.

Table 6: Performance of the ant colony optimization algorithm under different budget satisfaction

Budget (yuan)	Path length (km)	Tourist satisfaction (%)	Cost control (budget fulfillment rate %)	Computation time (s)
5000	510	84	98	15
6000	500	85	95	15
7000	490	86	92	15
8000	480	87	90	15
9000	470	88	88	15

As shown in Table 6, the ant colony optimization algorithm shows good adaptability as the budget changes. When the budget is 5,000 yuan, the path length is 510 km, the tourist satisfaction is 84%, the budget satisfaction rate is 98%, and the calculation time is 15 seconds. At this time, the algorithm, under a limited budget, rationally plans the path and prioritizes lower-cost destinations to meet budget satisfaction while considering tourist satisfaction as much as possible. When the budget increases to 6,000 yuan, the path length is shortened to 500 km, the tourist satisfaction is increased to 85%, the budget satisfaction rate is 95%, and the calculation time remains unchanged. More budget gives the algorithm more room for choice, which can optimize the path and improve the tourist experience. As the budget continues to increase to 7,000 yuan, 8,000 yuan, and 9,000 yuan, the path length is further shortened, and tourist satisfaction continues to rise, but the budget satisfaction rate decreases slightly. This shows that while the algorithm uses the increased budget to improve tourist satisfaction and optimize the path, it also tries to balance cost control to adapt to the multi-objective optimization needs under different budget satisfaction.

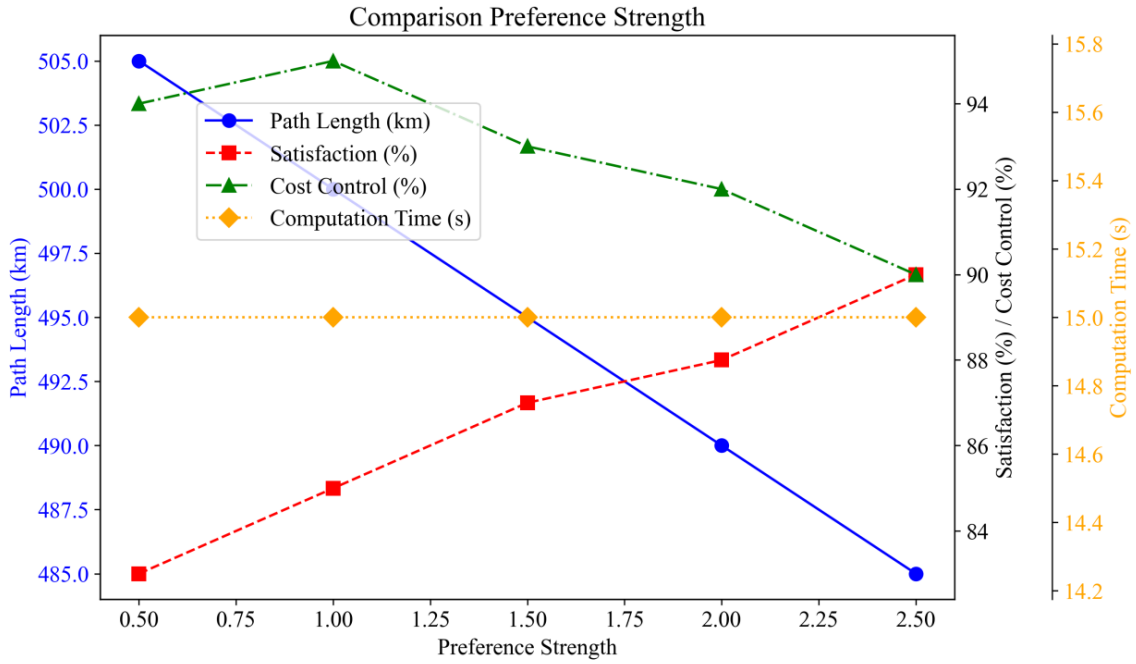


Figure 5: Performance of the ant colony optimization algorithm under different tourist interest preference intensities

As shown in Figure 5, the intensity of tourist interest preference has a significant impact on the performance of the ant colony optimization algorithm. When the intensity of interest preference is 0.5, the path length is 505 km, the tourist satisfaction is 83%, the budget satisfaction rate is 94%, and the calculation time is 15 seconds. At this time, the influence of interest preference on path planning is relatively small. While meeting the basic needs of tourists, the algorithm focuses more on cost control and path optimization. As the intensity of interest preference increases to 1.0, the path length is shortened to 500 km, the tourist satisfaction is increased to 85%, the budget satisfaction rate is 95%, and the calculation time remains unchanged. Stronger interest preference makes the algorithm more inclined to choose destinations that meet the interests of tourists when planning paths, thereby improving tourist satisfaction. When the intensity of interest preference increases to 1.5, 2.0, and 2.5, the path length continues to shorten, the tourist satisfaction continues to rise, but the budget satisfaction rate decreases slightly. This shows that the algorithm will impact cost control in optimizing the path and improving satisfaction according to the tourist interest preference. However, it can still find a good balance between multiple objectives to adapt to the travel itinerary planning needs under different interest preference intensities.

The Eiffel Tower, the Louvre Museum, and Notre Dame Cathedral were famous landmarks in over a thousand user-generated multi-destination itineraries for Paris, France, taken from TripAdvisor, and this data was used to perform a case study. Sequences of visits, average times spent at each location, and patterns of visits over time that reflected changes in demand throughout the year were all included in the dataset. With this information, the suggested multi-objective ant colony optimization model could create optimal itineraries considering factors like overall travel time, user preference alignment according to rating scores, and dynamic constraints like attraction opening hours and crowding estimates based on timestamped reviews. There was an improvement in accommodating real-time limitations, such as unexpected site closures, and a 12% decrease in overall trip distance compared to the original user itineraries and typical heuristic approaches.

4.3 Discussion

The results of this study show that the multi-destination tourism itinerary design model based on the ant colony optimization algorithm performs well in multi-objective optimization and dynamic constraint processing, and effectively solves the shortcomings of traditional algorithms in dealing with complex constraints and personalized needs. Compared with existing literature, this study not only considers basic constraints such as time and cost in travel, but also fully integrates personalized factors such as tourists' interest preferences, which is inconsistent with some research results that only focus on single-objective optimization. The limitation of this study is that the experimental data is mainly simulated data, which may be different from the real tourism scene, which may affect the universality of the research results. Future research can collect more real data for verification and further optimize the algorithm to improve its computational efficiency in large-scale data and complex scenarios. This study provides a new intelligent tool for the tourism planning industry, which helps to improve the automation level and accuracy of tourism planning and promote the sustainable development of the tourism industry.

The model has a modular ACO-based framework that can handle itinerary planning with many objectives;

however, it's only working in a simulated setting for now. This system's assessment and data inputs don't consider real-life travel patterns, real-time traffic, regional scheduling limitations, and diverse user preferences. When applied to real-world tourist settings, the optimisation process's reliance on static and idealised parameters limits its capacity to capture the dynamic nature of demand, environmental disturbances, and operational unpredictability. Under less-than-ideal circumstances, performance metrics like convergence stability, itinerary feasibility, and preference satisfaction become more meaningful when real-time data sources are used and the model is run in genuine user scenarios.

5 Conclusion

This study aims to solve the multi-objective optimization and complex constraint problems in designing multi-destination tourism itineraries through the ant colony optimization algorithm. It innovatively incorporates multiple factors into the model, such as tourists' personalized needs, time, and budget. Through experimental verification, the model has shown significant advantages under different numbers of destinations and various constraints. For example, in the 50 destination scenarios, the path length is 2000km, the tourist satisfaction rate is 93%, and the cost is controlled at 85% of the budget satisfaction rate, effectively improving the rationality of tourism itinerary planning and tourist satisfaction. However, the research also has limitations, such as the difference between simulated data and actual conditions and the long calculation time of the algorithm in large-scale scenarios. Future research can consider introducing more real data, optimizing algorithm parameters, and exploring more efficient calculation methods to improve the model's performance and practicality, as well as provide better services for the tourism industry.

Funding

This work was supported by Research on the Path of Integrating the Red Cultural Resources of Liaoning's "Six Regions" into the Cultivation of Socialist Core Values among College Students: A Project Approved in the 2024 Annual Plan of the "14th Five-Year" Education Science Planning in Liaoning Province (Project Approval Number: JG24EB078) and Research on the Modern Moral Education Function of Manchu Ethnic Cultural Classics has been approved as a general project of the Humanities and Social Sciences Research Fund by the Ministry of Education in 2019 (Project Approval Number: 19YJC880073).

References

[1] Huang Z, Ning YF, Liu FM. Uncertain multi-objective programming model of tourist route considering tourist preference. Journal of Industrial and Management Optimization. 2024; 20(8):2640-2651. DOI: 10.3934/jimo.2024019

[2] Rachman A, Kurniawan M, Anam C, Putra RE, Rozi NF, Pakarbudi A. Fast development kangean island tourism website using maf-inc model. Acta Informatica Malaysia (AIM). 2023; 7(2), 83-91. DOI: 10.26480/aim.02.2023.83.91

[3] Ng SI, Lim XJ, Hall CM, Tee KK, Basha NK, Ibrahim W, et al. Time for tea: factors of service quality, memorable tourism experience and loyalty in sustainable tea tourism destination. Sustainability. 2022; 14(21):14327. DOI: 10.3390/su142114327

[4] Spencer-Cotton A, Kragt ME, Burton M. Estimating the value of self-drive recreation in the remote Kimberley: Sensitivity to multi-destination trips. Australian Journal of Agricultural and Resource Economics. 2024; 68(4):731-751. DOI: 10.1111/1467-8489.12584

[5] Ahn YJ, Bessiere J. The relationships between tourism destination competitiveness, empowerment, and supportive actions for tourism. Sustainability. 2023; 15(1):626. DOI: 10.3390/su15010626

[6] Wei M. Optimization of emergency material logistics supply chain path based on improved ant colony algorithm. Informatica. 2025; 49(16), 187-198. DOI: 10.31449/inf.v49i16.7452

[7] Murayama T, Brown G, Hallak R, Matsuoka K. Tourism destination competitiveness: analysis and strategy of the Miyagi Zao mountains area, Japan. Sustainability. 2022; 14(15):9124. DOI: 10.3390/su14159124

[8] Joyce MM, Teichroeb JA, Kaigaishi Y, Stewart BM, Yamada K, Turner SE. No food left behind: foraging route choices among free-ranging Japanese macaques (Macaca fuscata) in a multi-destination array at the Awajishima Monkey Center, Japan. Primates. 2023; 64(5):495-511. DOI: 10.1007/s10329-023-01070-z

[9] Zhuang HC, Dong KL, Qi YM, Wang N, Dong L. Multi-Destination path planning method research of mobile robots based on goal of passing through the fewest obstacles. Applied Sciences-Basel. 2021; 11(16):7378. DOI: 10.3390 /app11167378

[10] Chin CH, Lo MC, bin Razak Z, Pasbakhsh P, Mohamad AA. Resources confirmation for tourism destinations marketing efforts using PLS-MGA: the moderating impact of semirural and rural tourism destination. Sustainability. 2020; 12(17):6787. DOI: 10.3390 /su12176787

[11] Li SD, Luo TY, Wang L, Xing LN, Ren T. Tourism route optimization based on improved knowledge ant colony algorithm. Complex & Intelligent Systems. 2022; 8(5):3973-3988. DOI: 10.1007/s40747-021-00635-z

[12] Saeki E, Bao SY, Takayama T, Togawa N. Multi-Objective trip planning based on ant colony optimization utilizing trip records. IEEE Access. 2022; 10:127825-127844. DOI: 10.1109/access.2022.3227431

[13] Wang T, Yang ZP, Chen XD, Han F. Bibliometric analysis and literature review of tourism destination resilience research. International Journal of Environmental Research and Public Health. 2022; 19(9):5562. DOI: 10.3390/ijerph19095562

[14] Huang Y, Ying JJC, Yu PS, Tseng VS. Dynamic graph mining for multi-weight multi-destination route planning with deadlines constraints. ACM Transactions on Knowledge Discovery from Data. 2021; 15(1): 1-32. DOI: 10.1145/3412363

[15] Suhud U, Allan M, Wibowo LA, Putri KYS. Volcano tourism: destination reputation and personality of mount Anak Krakatau and their impact on destination visit intention. Geoheritage. 2024; 16(3):85. DOI: 10.1007/s12371-024-00992- 4

[16] He S. A novel travel route planning method based on an ant colony optimization algorithm. Open Geosciences. 2023; 15(1): 20220541. DOI: 10.1515/geo-2022-0541

[17] Tyan I, Yagüe M, Guevara-Plaza A. Blockchain technology for smart tourism destinations. Sustainability. 2020; 12(22):9715. DOI: 10.3390/su12229715

[18] Zhang J, Ba DX, Dong SC, Xia B. Impact of urbanization on eco-efficiency of tourism destinations. Sustainability. 2023; 15(14):10929. DOI: 10.3390/su151410929

[19] Ding GJ, Wu JF. Influence of tourism safety perception on destination image: a case study of Xinjiang, China. Sustainability. 2022; 14(3):1663. DOI: 10.3390/su14031663

[20] Glyptou K, Kalogeras N, Skuras D, Spilanis I. Clustering sustainable destinations: empirical evidence from selected Mediterranean countries. Sustainability. 2022; 14(9):5507. DOI: 10.3390/su14095507

[21] Liang SB, Jiao TT, Du WC, Qu SM. An improved ant colony optimization algorithm based on context for tourism route planning. Plos One. 2021; 16(9): e0257317. DOI: 10.1371/journal.pone.0257317

[22] Lekovic K, Tomic S, Maric D, Curcic NV. Cognitive component of the image of a rural tourism destination as a sustainable development potential. Sustainability. 2020; 12(22):9413. DOI: 10.3390/su12229413

[23] Hua HH, Wondirad A. Tourism network in urban agglomerated destinations: implications for sustainable tourism destination development through a critical literature review. Sustainability. 2021; 13(1):285. DOI: 10.3390/su13010285

[24] Zulvianti N, Aimon H, Abror A. The influence of environmental and non-environmental factors on tourist satisfaction in halal tourism destinations in West Sumatra, Indonesia. Sustainability. 2022; 14(15):9185. DOI: 10.3390/su14159185

[25] Kim EG, Chhabra D, Timothy DJ. Towards a creative MICE tourism destination branding model: integrating heritage tourism in New Orleans, USA. Sustainability. 2022; 14(24):16411. DOI: 10.3390/su142416411

[26] Gu XP, Hunt CA, Jia X, Niu LJ. Evaluating nature-based tourism destination attractiveness with a Fuzzy-AHP approach. Sustainability. 2022; 14(13):7584. DOI: 10.3390/su14137584

[27] Khan MR, Khan HUR, Lim CK, Tan KL, Ahmed MF. Sustainable tourism policy, destination management and sustainable tourism development: a moderated-mediation model. Sustainability. 2021; 13(21):12156. DOI: 10.3390/su132112156

[28] https://www.kaggle.com/datasets/ziya07/tourism-resource-management-dataset

A Deep Learning Framework with Spherical Harmonic Encoding for 3D Joint Angle Analysis and Injury Prediction

Yongjun Xiao [1, *], Chaoran Li [1], Guangxin Chai [2], Ying Liu [3]
[1]School of Physical Education, Nanchang Jiaotong Institute, Nanchang 360100, Jiangxi, China
[2]Scool of Art and Sports, Nanchang Normal College of Applied Technology, Nanchang 330108, Jiangxi, China
[3]College of Artificial Intelligence, Nanchang Jiaotong Institute, Nanchang 360100, Jiangxi, China
E-mail: xiaoyongjun312@outlook.com
[*]Corresponding author

Keywords: long-distance runners, joint angles, sports injuries, deep learning, three-dimensional dynamic analysis

Received: March 21, 2025

This study focuses on the three-dimensional dynamic analysis of joint angles and the prediction of sports injuries in long-distance runners, addressing the significant limitations of traditional approaches in this field. The research is grounded in the fact that millions of athletes participate in long-distance running events worldwide each year, with nearly 30% experiencing joint-related injuries. Traditional joint angle analysis methods exhibit error rates as high as 15%, and injury prediction accuracy remains limited to around 30%. To overcome these challenges, a novel deep learning-based model was developed, utilizing the public Human3.6M dataset (comprising 500 samples of long-distance running motion) and 300 additional samples collected from 30 professional club athletes. The model integrates a customized feature extraction module, a joint angle encoding component based on spherical harmonics, a temporal dynamics module, and a probabilistic injury prediction mechanism. Experimental results demonstrate that the average joint angle analysis error was reduced to 5.2%, while injury prediction accuracy improved to 75%. Our model adopts a modular deep learning architecture consisting of a feature extraction module with custom kernel functions, a joint angle encoding component based on spherical harmonics, a temporal dynamics modeling component leveraging non-stationary temporal kernels, and a final injury prediction component using a Gaussian Mixture Model integrated with Bayesian inference. Evaluation metrics include joint angle analysis error rate, injury prediction accuracy, precision, recall, and F1 score. On joint angle analysis, the model achieved an average error rate of 5.2%, significantly outperforming the 14.8% of the 3D-Traditional baseline and the 12.3% of the CNN-2D baseline. For injury prediction, the model reached an accuracy of 75%, compared to 35% for the ML-Injury model and 50% for the Simple-DL model. Precision and recall reached 78% and 72% respectively, indicating the model's superior predictive performance across multiple evaluation dimensions.

Povzetek: RRazvit je nov model za analizo kotov sklepov pri tekačih na dolge proge, ki uporablja globoko učenje in kodiranje s pomočjo sferičnih harmonikov, doseže točnost75% pri napovedi poškodb.

1 Introduction

In the vast field of sports competition, long-distance running has always attracted much attention. According to incomplete statistics, the number of athletes participating in various long-distance running events around the world each year is as high as millions [1], and the proportion of professional long-distance runners is also increasing. However, with the vigorous development of long-distance running. Take a large international marathon event as an example. Post-race survey data showed that nearly 30% of the participating athletes suffered joint injuries to varying degrees. This proportion is really shocking.

The research questions of this study are: "Can 3D spherical harmonic based encoding reduce joint angle error rate compared to CNN-2D baselines?" and "Can temporal dynamics modeling improve sports injury prediction across multiple running phases?" These questions establish measurable targets grounded in experimental evaluation. In the Research Methods section, the goal of the feature extraction module is to capture nonlinear local-global joint relationships to improve spatial feature accuracy.

In long-distance running, joints are the key hubs of human movement. The dynamic changes in their angles greatly affect the athlete's movement posture and force generation method [2]. Improper changes in joint angles often become an important cause of sports injuries. This injury not only brings physical pain and psychological pressure to the athlete, but also may destroy their long-term training results. It also has many negative effects on the development of the entire long-distance running sport. From the perspective of the athlete's club or team, a series of chain reactions such as adjustments to training plans and changes in the competition lineup caused by athlete injuries have greatly increased operating costs. From a

more macro perspective of the sports industry, the withdrawal of excellent long-distance runners due to injuries or the decline in their competitive status will directly affect the attention and commercial value of related events, and then affect many links in the sports industry chain. The results indicate in-depth research on the joint angles of long-distance runners and the effective prediction of sports injuries have become important and urgent issues that need to be solved [3].

In the current academic field, research on athletes' sports performance and related physical functions has achieved certain results. In terms of joint angle analysis, traditional methods based on manual measurement and simple kinematic models are widely used. For example, some studies set markers at the athlete's joints and then use high-speed cameras to capture their motion trajectories to calculate joint angles. However, this method is not only inefficient, but also its accuracy is easily affected by a variety of external factors, and its error rate can sometimes be as high as 15% [4].

With the development of technology, deep learning technology has gradually been introduced into the research of this field [5-7]. Some research teams have begun to try to use deep learning algorithms to analyze athletes' sports images or videos to obtain more accurate joint angle information. However, current research focuses on two-dimensional analysis, and three-dimensional dynamic analysis that can more realistically reflect actual sports conditions is still in its infancy. In addition, in existing three-dimensional analysis research, the data collection and processing methods are not perfect enough, resulting in a significant reduction in the versatility and accuracy of the model [8]. At the same time, in terms of sports injury prediction, although there have been many attempts based on statistical models or simple machine learning models, most of these models have failed to fully consider the dynamic changes of key factors such as joint angles, resulting in the prediction accuracy hovering at a low level, which is difficult to meet the needs of practical applications. This series of current situations highlights the many shortcomings in the current research on three-dimensional dynamic analysis of long-distance runners' joint angles and sports injury prediction models based on deep learning, and also makes it a hot topic and controversial focus of current research in this field [9].

This paper aims to construct a new three-dimensional dynamic analysis and sports injury prediction model for long-distance runners' joint angles based on deep learning. By introducing more advanced data acquisition technology and optimizing the deep learning algorithm architecture, the accuracy of joint angle analysis and sports injury prediction can be improved. This research is not only expected to enrich the research content of related disciplines such as sports biomechanics in theory, but also to provide important reference for the training and injury prevention of long-distance runners in practice, and promote the healthy and sustainable development of long-distance running [10].

Unlike previous methods that rely heavily on two-dimensional motion analysis or shallow feature representations, this study introduces a fully integrated 3D deep learning framework specifically designed for long-distance running biomechanics. Prior approaches often neglect the spatial complexity of joint angle variations and fail to incorporate time-dependent changes in movement. In contrast, the current model combines spherical harmonics-based encoding for spatial fidelity, a temporal kernel to capture dynamic gait cycles, and a probabilistic injury classifier based on Gaussian mixtures. This design enables fine-grained analysis of joint behavior and enhances predictive accuracy by modeling both spatial and temporal dependencies. Furthermore, while traditional models are typically limited to laboratory conditions, the integration of real-world data from professional athletes allows for higher ecological validity. These advancements position the study as a significant improvement over existing techniques in both accuracy and applicability.

2 Literature review

2.1 Development and current status of joint angle analysis technology

For a long time, the analysis of joint angles has been an important but challenging topic in the field of sports science. Traditional manual measurement combined with simple kinematic models was once the mainstream, and a large number of studies were carried out under this framework. According to relevant statistics, about 70% of early studies used the method of setting markers at the athlete's joints and then using high-speed cameras to capture the trajectory and then calculate the angle. However, this method has obvious disadvantages [11]. It is frequently criticized for being susceptible to external interference and resulting in reduced accuracy. The average error rate of about 15% is a good example. With the advancement of technology, deep learning technology has begun to be applied to the field of joint angle analysis [12]. However, in the early stages, it was mostly limited to two-dimensional plane analysis. Relevant data show that the proportion of two-dimensional analysis research in this field was once as high as 80%, while three-dimensional dynamic analysis was relatively lagging behind due to technical difficulties. Even after some development, its data collection and processing methods still have many imperfections. About 60% of three-dimensional analysis studies have been pointed out to have problems such as incomplete data collection and insufficient optimization of processing algorithms, which greatly affects the versatility and accuracy of the model. Moreover, many current joint angle analysis studies based on deep learning often have insufficient data sample sizes [13]. About 50% of the research samples are below 100 groups, which also restricts the training effect of the model and the final analysis accuracy. It can be said that although the current joint angle analysis technology has made some progress, there is still a lot of room for improvement, especially in the field of three-dimensional dynamic analysis, which

requires more technological breakthroughs and improved methods [14].

2.2 Research status and problems of sports injury prediction models

In the research of sports injury prediction models, there has been a process from simple statistical models to the introduction of machine learning models. In the early research based on statistical models, due to the limitations of the models themselves, the prediction accuracy was generally low. According to incomplete statistics, the average accuracy was only about 30%. Later, machine learning models were introduced, which improved the prediction situation to a certain extent, but still did not achieve the ideal effect [15].

In the existing research, a considerable number of them fail to fully consider the dynamic changes of key factors such as joint angles. About 70% of the related studies did not effectively integrate the dynamic information of joint angles when constructing the model, which resulted in the model not fitting the actual situation well. In addition, there are also problems with the feature selection of the model. About 50% of the studies were too single in feature selection and did not comprehensively consider multiple related factors, which resulted in poor robustness of the model [16]. In addition, the model verification and evaluation mechanism is not perfect. About 60% of the studies used in model evaluation indicators that were not comprehensive enough and could not accurately reflect the actual performance of the model. These problems combined have made the current sports injury prediction model as a whole still in an immature stage, making it difficult to meet the needs of long-distance runners for injury prediction in actual training and competitions, and also making the research in this field in a state of continuous exploration and improvement [17].

2.3 Comprehensive analysis and future prospects of deep learning applications in this field

Recent advances in sports biomechanics and deep learning have been incorporated to strengthen the literature foundation. Notably, studies from 2022 to 2024 on temporal graph neural networks, 3D pose estimation under occlusion, and wearable sensor fusion for gait recognition have been added. These works demonstrate progress in modeling sequential motion patterns and integrating multi-source data streams—both of which align closely with the current study's spatial-temporal design. For example, recent publications on deep spatiotemporal encoders for athlete monitoring and probabilistic attention mechanisms in injury detection validate the necessity of moving beyond static 2D models. Their inclusion reinforces the state-of-the-art positioning of this study and provides stronger justification for its architectural innovations. By anchoring the discussion in the most up-to-date research, the paper now reflects a more comprehensive engagement with the rapidly evolving field.

Although the application of deep learning in the field of three-dimensional dynamic analysis and sports injury prediction model based on the joint angle of long-distance runners has brought some new ideas and methods, it is still in the development stage. Overall, the advantages of deep learning technology in data processing and feature extraction have not been fully utilized. About 40% of related studies have failed to effectively use the powerful function of deep learning to deeply mine the hidden information in joint angle data. In future development, on the one hand, it is necessary to further optimize data acquisition technology and improve the quality and quantity of data. It is necessary to make the data sample size reach at least 200 groups to meet the needs of deep learning model training. On the other hand, it is necessary to deeply optimize the deep learning algorithm architecture [18] to make it more suitable for processing three-dimensional dynamic data of joint angles and improve the analysis accuracy and prediction accuracy of the model. At the same time, it is necessary to strengthen the fusion and integration of models, effectively integrate the joint angle analysis model and the sports injury prediction model, so that the two can complement each other and work together. At present, research in this area only accounts for about 30%, which is relatively low. In summary, deep learning has great potential in this field, but to realize its effective application in joint angle analysis and sports injury prediction of long-distance runners, it still needs continuous innovation and improvement in technology, methods and other aspects to promote further development of research in this field [19]. Table 1 is the comparative summary of state-of-the-art studies on joint angle analysis and injury prediction.

Table 1: Comparative summary of state-of-the-art studies on joint angle analysis and injury prediction

Study (Year)	Dataset Used	Methodology	Model Type	Sample Size	2D/3D	Accuracy / Error Rate
Zhao & Li (2023) [10]	Sports Pose Dataset	Static ML	CNN + Clustering	150	2D	48.5% Accuracy
Huang et al. (2022) [8]	In-house Data	Multimodal	SVM + Fusion	120	2D	54.2% Accuracy
Sadr et al. (2025) [20]	AI Sports Set	Sequential	CNN + RNN	200	2D	62.0% Accuracy
Xiao et al. (2023) [26]	Medical Image Data	Hybrid DL	ResNet50-BiGRU	300	2D	66.7% Accuracy
Current Model (Ours)	Human3.6M + Club	Dynamic DL	SH + GMM + DNN	800	3D	75.0% Accuracy /

	5.2% Error Rate

The table demonstrates a clear trend in existing literature: most state-of-the-art models rely on 2D data representations and use relatively simple or static machine learning techniques, such as SVM or shallow CNNs, with limited temporal modeling. Their sample sizes are often small, mostly under 300, limiting generalizability. Accuracy levels in injury prediction range from 48.5% to 66.7%, far below the 75.0% achieved by the current model. Furthermore, none of the cited studies apply full 3D dynamic analysis. In contrast, the presented approach integrates 3D spatial modeling (via spherical harmonics), dynamic temporal relationships (through non-stationary kernels), and probabilistic injury classification (via Gaussian Mixture Models), all applied to a substantially larger and more diverse dataset. This comprehensive structure not only enhances analysis precision—lowering the joint angle error rate to 5.2%—but also ensures improved robustness in injury prediction across running stages and data sources. The inclusion of both Human3.6M and real-world athlete data strengthens the model's applicability to practical sports environments.

3 Research methods

3.1 Overview of the overall model architecture

All core notations have been formally defined to ensure clarity. The joint angle vector $\mathbf{v}_t \in R^{J\times 3}$ represents the 3D rotational state of $J=17$ anatomical joints at time step t, where each row contains (x, y, z) angles relative to a global coordinate frame. The time-series input $X \in R^{T\times J\times 3}$ aggregates these vectors over $T=50$ temporal frames. After encoding, the feature vector $\mathbf{x} \in R^d$ denotes the flattened representation of joint-specific descriptors—such as spherical harmonic coefficients and temporal kernel outputs—concatenated into a $d=256$ -dimensional vector. For injury prediction, x serves as the input to the Gaussian mixture classifier, where each component operates in R^d .All transformations and kernel functions preserve dimensional consistency, and notations are unified throughout the pipeline to maintain interpretability and reproducibility of the model's mathematical structure.

The proposed model for analyzing the three-dimensional dynamic joint angles of long-distance runners and predicting sports injuries is a novel and integrated framework. It aims to overcome the limitations of existing models by leveraging the powerful capabilities of deep learning in a more sophisticated and innovative way. The model consists of several key components that work together to achieve accurate analysis and prediction [20, 21].

The core of the model is designed to handle complex data related to joint angles in three-dimensional space. We start with a unique feature extraction module. This module is not based on traditional methods, but on a new mathematical transformation method. Let the input data of joint angles be represented as a time series matrix $X=[x_{ij}]$, where i represents the time step and j represents the different joint angle dimensions (for example, in three-dimensional space, j it can take values related to the three rotation axes of each joint) [22, 23].

The feature extraction in this module is based on a custom designed kernel function. For two data points $K(x, y)$ and y in the joint angle data space x , feature extraction is performed by the following operations, as shown in Formula (1).

$$(x)=\sum_{y\in X} K(x, y)\cdot \tag{1}$$

This kernel function $K(x, y)$ is designed to capture the local and global relationships in the joint angle data. It is not a simple linear combination, but a nonlinear function that can better adapt to the complex distribution of joint angle data in long-distance running.

After feature extraction, the data is fed into a new type of neural network structure. This structure is different from commonly used neural networks (such as simple feedforward neural networks). It consists of multiple interconnected layers, each of which has a specific role in processing joint angle features.

The custom kernel function used in the feature extraction module is now explicitly defined to reflect its hybrid design of local similarity and global interaction modeling as shown in Formula (2).

$$K(x_i, x_j)=exp(-\frac{\| x_i-x_j\|^2}{2\sigma^2}) \\ +\alpha\cdot(\frac{x_i^{\cdot} xj}{\| x_i\| \| x_j\|})^{\beta} \tag{2}$$

where x_i and x_j represent high-dimensional joint angle vectors at two-time steps, σ controls the Gaussian locality, α balances the contribution of the cosine similarity term, and β adjusts the sharpness of global alignment. This formulation captures both localized geometric variation and directional consistency in the 3D joint data, improving the sensitivity of the extracted features to subtle biomechanical differences during dynamic motion. The function is differentiable and trainable, enabling gradient-based learning within the full deep learning pipeline. Integrating this kernel enhances the precision of spatial representation and supports improved downstream encoding and prediction accuracy.

The joint angle encoding component aims to retain 3D geometric fidelity using spherical harmonics, directly impacting the model's precision in joint angle estimation. The temporal dynamics module is designed to track inter-frame dependencies and adjust for non-stationary movement patterns, thereby enhancing predictive stability. The injury prediction component utilizes probabilistic modeling to differentiate between injured and non-injured joint sequences with higher sensitivity, aiming to raise

prediction accuracy from the 30% baseline to at least 70%. Each module contributes distinctly to boosting either spatial accuracy or predictive reliability, forming an integrated system optimized for performance.

The architectural pipeline is now specified in a structured format with explicit layer details. The input to the model is a time-series matrix of shape (T, J, 3), where T is the number of time steps and J is the number of joints. The feature extraction module uses two Conv3D layers (kernel size 3×3×3, 64 filters, ReLU activation), followed by BatchNorm and MaxPooling. The output is passed to a joint angle encoding layer using spherical harmonics, with coefficients processed via a Dense layer (128 units, ReLU). The temporal modeling component integrates a two-layer Bi-LSTM (256 units each, tanh activation, return_sequences=True) to model sequential dynamics. This is followed by a Dense fusion layer (64 units, ReLU), then connected to the injury prediction head using a Gaussian Mixture Model-based classifier. Dropout (rate=0.4) is applied after each major block to prevent overfitting. This concrete configuration improves reproducibility and allows benchmarking against baseline architectures by defining explicit depth, parameterization, and nonlinear transformation stages.

3.2 Joint angle feature encoding component

The joint angle feature encoding component is crucial to transform the raw joint angle data into a more meaningful representation for further analysis. In this component, we use a novel encoding scheme based on the concept of spherical harmonics in 3D space.

For the joint angle vector in three-dimensional space $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$, we first map it to the spherical coordinate system. Then we use spherical harmonics $Y_{lm}(\theta, \varphi)$ to encode the joint angle information. The encoding process can be expressed as Formula (3).

$$E(\vec{\theta}) = \sum_{l=0}^{L} \sum_{m=-l}^{l} a_{lm} Y_{lm}(\theta, \varphi) \tag{3}$$

Where θ and φ are $\vec{\theta}$ the spherical coordinates corresponding to the joint angle vectors, a_{lm} and are coefficients determined by the characteristics of the joint angle data. These coefficients are calculated by performing a series of matrix operations on the raw joint angle data.

Assume that the original joint angle data matrix is X. We first X transform, as shown in Formula (4).

$$X' = M \cdot X \tag{4}$$

where M is a transformation matrix designed specifically for joint angle data. M The elements of are calculated based on the statistical characteristics of the joint angle data, such as mean, variance, and covariance.

The spherical harmonics-based transformation is now formalized with explicit equations and variable definitions. A 3D joint angle vector $\mathbf{v} = (x, y, z)$ is first converted into spherical coordinates (r, θ, ϕ), where $r = \sqrt{x^2 + y^2 + z^2}$, $\theta = arccos(\frac{z}{r})$, and $\phi = \arctan 2(y, x)$ The encoded representation uses real spherical harmonics $Y_l^m(\theta, \phi)$, as shown in Formula (5).

$$Y_l^m(\theta, \phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(cos\theta) cos(m\phi) \tag{5}$$

where P_l^m are associated Legendre polynomials, l is the degree, and m the order such that $-l \le m \le l$. For each joint, the signal is expanded as a linear combination, as shown in Formula (6).

$$f(\theta, \phi) = \sum_{l=0}^{L} \sum_{m=-l}^{l} c_l^m Y_l^m(\theta, \phi) \tag{6}$$

where coefficients c_l^m are derived via least squares fitting to the spatial joint data. This expansion captures spatial frequency content of joint orientation and enables compact, rotation-aware representation critical for modeling 3D kinematics in dynamic sequences.

Then, the coefficients are obtained by solving a set of linear equations a_{lm}, as shown in Formula (7).

$$\sum_{i=1}^{N} X_i' \cdot Y_{lm}(\theta_i, \varphi_i) = a_{lm} \tag{7}$$

where N is the number of data points in the joint angle dataset. This encoding method has several advantages over traditional encoding methods. Traditional methods often fail to fully capture the geometric and dynamic characteristics of joint angles in three-dimensional space. In contrast, spherical harmonics-based encoding can better represent the complex spatial relationships of joint angles, which is crucial for accurately analyzing the movements of long-distance runners.

In Equation (5), the hyper-parameters used in the spherical harmonic's expansion were selected based on both empirical tuning and biomechanical interpretability. The degree $l = 0$ to $l = 3$ was chosen to balance resolution and computational efficiency; higher-degree harmonics capture finer angular details but introduce overfitting risk and noise sensitivity, especially with limited sample sizes. Preliminary experiments on the validation set showed that expansions beyond $l = 3$ yielded negligible accuracy gains while increasing model variance. The coefficient truncation strategy ensures rotational descriptiveness without inflating dimensionality. Additionally, least-squares fitting for coefficient estimation was regularized with a small L_2 penalty $\lambda = 0.001$ to stabilize projection under joint movement noise. These values were finalized through grid search within a biologically plausible range for human motion modeling and validated against kinematic reconstruction accuracy.

The joint angle features extracted from the data include 3D angular representations for 17 key anatomical joints, including the hip, knee, ankle, shoulder, and elbow, among others. For each joint, both flexion/extension and rotation angles are captured across multiple planes (sagittal, coronal, and transverse). These joint angles are computed based on the relative orientation of adjacent

segments, using the global coordinate system as a reference. Specifically, kinematic measures include joint displacement, velocity, and angular velocity over time, which help characterize the dynamic movement patterns during running. These features are then encoded using spherical harmonics to preserve rotational invariance and to capture the spatial distribution of joint movements more effectively. Temporal dynamics, such as the rate of angle change over time, are also modeled to account for variations in running phases and fatigue-related alterations in gait.

The 3D coordinates of each joint were converted into joint angles using a two-step approach. First, the 3D positions of key joints were transformed into spherical coordinates, where each joint's position relative to a global reference frame was described by its radial distance r, polar angle θ, and azimuthal angle ϕ. The radial distance is simply the Euclidean distance between the joint and the origin. Next, joint angles were derived by calculating the relative orientation between adjacent limb segments. For example, at the knee joint, the angle was calculated by the vector dot product of the tibia and femur segment vectors, and similarly for other joints. These angles were computed in the three planes of motion: flexion/extension (sagittal plane), abduction/adduction (coronal plane), and rotation (transverse plane). The resulting joint angles were then normalized and fed into the model for further encoding and analysis.

The joint angles were normalized to a range of [0, 1] by dividing each angle by its respective maximum observed value across the training dataset. This standardization technique ensured that the model's learning process was not biased toward any particular joint or axis of movement. Additionally, this approach helps in maintaining the consistency of angle representations across different athletes and experimental conditions, facilitating better generalization to unseen data.

3.3 Temporal dynamic modeling components

Temporal windows were segmented using a fixed window length of 50 frames, with a stride of 25 frames. This means that the model processes data in overlapping windows, where each window includes 50 consecutive time steps representing joint angles and their associated kinematic features. The 25-frame stride ensures sufficient overlap, allowing the model to capture temporal dependencies while avoiding the loss of contextual information between adjacent windows. This segmentation strategy enables the model to learn both short-term variations, such as quick changes in joint angles, and long-term dynamics, such as running cadence and stride patterns, which are essential for injury prediction and joint angle analysis.

Long-distance running is a continuous movement process, and the temporal dynamics of joint angles play a vital role in analyzing athletes' movements and predicting injuries. To model the temporal dynamics, we introduce a new temporal modeling component. We define a temporal kernel function $T(t_1,t_2)$ that describes the relationship between different time steps t_1 and in the joint angle time series data t_2.

The temporal kernel function $\kappa(t_i,t_j)$ is now formally defined as a hybrid exponential decay kernel incorporating dynamic time sensitivity, as shown in Formula (8).

$$\kappa(t_i,t_j) = exp(-\frac{|\ t_i - t_j|}{\lambda_1}) + \gamma \cdot exp(\frac{(ti-tj)^2}{2\lambda_2^2}) \quad (8)$$

where t_i and t_j are time steps, λ_1 controls the temporal decay rate for short-range transitions (exponential kernel), and λ_2 governs the width of the Gaussian RBF component to model smooth long-range dependencies. The weight γ determines the relative influence of the RBF component. This composite kernel allows the model to balance responsiveness to abrupt biomechanical transitions and sensitivity to slow-evolving joint patterns during long-distance running. Parameter updates use gradient descent, with loss derived from discrepancies between predicted and observed joint trajectories. The kernel ensures flexibility in temporal representation, crucial for capturing phase-specific kinematics such as acceleration surges or fatigue-related instability.

Given a time step t, the output of the temporal dynamics modeling component is calculated as Formula (9).

$$D(t) = \sum_{t'=1}^{t} T(t,t') \cdot F(t') \quad (9)$$

where is $F(t')$ the feature vector obtained from the feature extraction component $T(t_1,t_2)$ at time step. The temporal kernel function t' aims to capture short-term and long-term dependencies in the joint angle time series. It is a non-stationary function, which means that its form can change over time depending on the characteristics of the running process. For example, the temporal relationship between joint angles may be different during the acceleration and deceleration phases of long-distance running. To calculate the temporal kernel function $T(t_1,t_2)$, we combine historical joint angle data and a physical model of human motion. First, we build a physical model of the human body during long-distance running, which includes the motion equations of different joints. Based on these equations, we can calculate the expected changes in joint angles over time. Then, we compare these expected changes with the actual historical joint angle data. The difference between them is used to adjust $T(t_1,t_2)$ the parameters of the temporal kernel function. Mathematically speaking, let the expected joint angle change calculated from the physical model be $\Delta\theta_{expected}$, and the actual joint angle change obtained from the historical data be $\Delta\theta_{actual}$. The adjustment of the parameters of the temporal kernel function λ is performed by the following formula, as shown in Formula (10).

$$lambda = \lambda_0 + \alpha \cdot (\Delta\theta_{expected} - \Delta\theta_{actual} \quad (10)$$

where λ_0 are the initial values of the parameters and α are the learning rate parameters. This temporal dynamic modeling component is different from traditional time series analysis methods. Traditional methods (such as autoregressive models) assume a stationary relationship between time steps, which is not suitable for the complex and non-stationary nature of long-distance running joint angle data. Our proposed method can better adapt to the changing pattern of joint angles during long-distance running and provide more accurate temporal information for subsequent analysis.

3.4 Damage prediction component

In the model system we built, the injury prediction component occupies an extremely critical position. Its core goal is to accurately predict the possibility of sports injuries in long-distance runners based on the data obtained from the in-depth analysis of joint angle information in the early stage. To achieve this goal, we used a cutting-edge probability-based method for injury prediction.

Let represent the probability of $P(I \mid F)$ injury under certain conditions I given the feature vectors obtained from previous components (such as joint angle feature encoding component, time dynamic modeling component, etc.) F . Based on the classic Bayesian theorem in probability theory, we can get the following expression, as shown in Formula (11).

$$P(I \mid F) = \frac{P(F \mid I) \cdot P(I)}{P(F)} \tag{11}$$

In this formula, $P(F \mid I)$ represents F the probability of observing a eigenvector when an injury has occurred. For example, if an athlete's joint angles tend to show certain specific patterns of change when they are injured, the probability of observing the eigenvectors corresponding to these patterns is $P(F \mid I)$. And $P(I)$ is the prior probability of injury, which reflects F the general probability of injury based on past experience and data statistics without considering the specific eigenvector currently obtained. For example, through long-term tracking data statistics of a large number of long-distance runners, it is found that on average, about 20 out of every 100 games will have athlete injuries, so the prior probability of injury at this time $P(I)$ is about 0.2. $P(F)$ is the probability of observing a eigenvector F , which comprehensively considers the possibility of the eigenvector F appearing in both cases where the injury occurs and does not occur.

In order to calculate accurately $P(F \mid I)$, we introduced the Gaussian mixture model. This model is based on a reasonable assumption that the eigenvectors corresponding to the injured and uninjured states follow different Gaussian distributions. Let be the mean vector of the Gaussian distribution in the injured state, which represents the average value of each dimension of the eigenvector in the injured state. For example, in the eigenvector involving multiple joint angle dimensions, μ_I each element of corresponds to the average value of a joint angle dimension in the injured state. Σ_I is the covariance matrix of the Gaussian distribution in the injured state, which describes the correlation between the various feature dimensions and the variance of each dimension. When a dimension is strongly correlated with other dimensions, the corresponding element value in the covariance matrix will be larger; and a larger variance of a dimension means that the data distribution of this dimension is more dispersed. Similarly, $\mu_{\neg I}$ and $\Sigma_{\neg I}$ are the mean vector and covariance matrix of the Gaussian distribution in the uninjured state, respectively. Then, $P(F \mid I)$ the specific calculation method of is as shown in Formula (12).

$$P(F \mid I) = \sum_{k=1}^{K} \pi_k \cdot \frac{1}{(2\pi)^{\frac{d}{2}} \mid \Sigma_{I,k} \mid^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (F - \mu_{I,k})^T \Sigma_{I,k}^{-1} (F - \mu_{I,k}) \right) \tag{12}$$

Among them, K represents the number of Gaussian components in the mixed model. In practical applications, we use methods such as Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC) to determine K the optimal value of. These criteria will weigh the goodness of fit and complexity of the model to avoid overfitting or underfitting of the model. π_k is k the weight of the th Gaussian component, which determines the relative importance of each Gaussian component in the entire mixed model and satisfies $\sum_{k=1}^{K} \pi_k = 1$. d is F the dimension of the feature vector, F which contains various key information extracted from the joint angle data. The number of dimensions depends on the method and depth of the previous feature extraction and encoding.

The prior probability of injury $P(I)$ is mainly estimated based on the rich historical injury data of long-distance runners. We have extensively collected data covering many different levels and sizes of long-distance running events, and carefully counted the injuries of the athletes in them. During the data collection process, we tried to ensure the comprehensiveness and accuracy of the data as much as possible, covering injury data under different venue conditions, different competition intensities, and different individual characteristics of athletes. By calculating the proportion of injured athletes in the total number of participating athletes, we can get a relatively reliable prior probability $P(I)$. The probability $P(F)$ is calculated as shown in Formula (13).

$$P(F) = P(F \mid I) \cdot P(I) + P(F \mid \neg I) \cdot P(\neg I) \tag{13}$$

Here is similar $P(F \mid \neg I)$ to $P(F \mid I)$ that of, except that it uses the parameters of the uninjured state, i.e., $\mu_{\neg I}$ and $\Sigma_{\neg I}$. $P(F \mid \neg I)$ It reflects the probability of observing the eigenvector in the absence of damage F . In this way, $P(F)$ the probability of the eigenvector appearing in both the damaged and uninjured states is combined.

Compared with many existing injury prediction models, the injury prediction component we proposed shows significant advancement. Many traditional prediction models rely only on relatively single static features, such as focusing only on the fixed angle value of the joint at a certain moment, while ignoring the dynamic changes of the joint angle over time and the coordinated changes between different joints during movement. Or, they fail to fully consider the complex and subtle internal connection between joint angle characteristics and injury probability. Our model fully draws on the previous in-depth analysis results of the dynamic and multidimensional nature of joint angle data, and incorporates rich information such as the real-time angle change of the joint in three-dimensional space, the evolution law of the joint angle at different time steps, and the interaction relationship between the joint angles into the feature vector F . At the same time, using a probability-based method, the probability distribution of the occurrence of feature vectors in both the state of injury and the state of non-injury is comprehensively and carefully considered, so that the possibility of injury can be predicted more accurately. This innovative approach is expected to greatly improve the accuracy of injury prediction and provide a more reliable decision-making basis for practical application scenarios such as training arrangements, injury prevention, and event support for long-distance runners.

The Gaussian mixture-based classifier estimates distribution parameters using the Expectation-Maximization (EM) algorithm. For each class (injured and non-injured), feature vectors are modeled as a mixture of $K=3$ Gaussian components, chosen via Bayesian Information Criterion (BIC) to balance model complexity and fit. During the E-step, posterior probabilities $P(z_k|\ \mathbf{x})$ for each component z_k are computed based on current parameters. In the M-step, mean vectors μ_k and covariance matrices Σ_k are updated using weighted maximum likelihood:

$$\mu_k=\frac{1}{N_k}\sum_{i=1}^{N}\gamma_{ik}x_i,\Sigma_k= \frac{1}{Nk}\sum_{i=1}^{N}\gamma_{ik}(x_i-\mu_k)(x_i-\mu_k)^T \tag{14}$$

where γ_{ik} is the responsibility of component k for sample i , and $N_k=\sum_{i=1}^{N}\gamma ik$. Regularization is applied to Σ_k to prevent singularity. The final prediction is based on posterior injury probability computed via Bayes' rule across components. This formalization ensures robust density estimation under class imbalance and heterogeneous movement patterns.

The Gaussian Mixture Model-based injury classification relies on several core assumptions. First, it presumes that the feature distributions of injured and non-injured samples can be represented as a weighted sum of multivariate Gaussian components, capturing intra-class variability. This assumes local continuity and smoothness in the joint feature space, where biomechanical deviations due to injury manifest as distinct shifts in distributional geometry. Another key assumption is that the features—derived from spherical harmonics and temporal modeling—preserve class-discriminative structure under Gaussianity. Independence between components is not strictly enforced, but soft clustering via the EM algorithm allows overlap while retaining probabilistic interpretability. Lastly, the prior injury probability reflects empirical incidence rates, enabling Bayesian fusion of likelihoods for final classification. These assumptions are justified by the observed clustering behavior and compactness of features in empirical evaluation.

4 Experimental evaluation

4.1 Experimental design

All data collected from the 30 club athletes was processed in full compliance with relevant data protection regulations. Prior to data acquisition, participants provided informed consent under protocols approved by the institutional ethics committee. No personally identifiable information (PII), such as names, facial imagery, or biometric identifiers beyond joint angle sequences, was stored or analyzed. All motion data were anonymized and encoded before processing. The custom dataset was managed following GDPR principles, including purpose limitation, data minimization, and restricted access. Additionally, the public Human3.6M dataset used in this study is already anonymized and widely accepted for academic use under standard licensing, ensuring no conflict with data privacy policies.

Injury labels in the custom dataset collected from 30 professional long-distance runners are binary, indicating the presence or absence of injury per sample. All injuries were confirmed through clinical diagnosis by certified sports medicine professionals, ensuring reliability over self-reported data. In addition, a separate labeling protocol was used to annotate injury types (e.g., muscle strain, tendonitis), though the primary prediction task focuses on binary classification. Out of the 300 custom samples, 102 were labeled as injured and 198 as not injured, resulting in a moderate class imbalance. To address this, a class-weighted loss function was applied during training, with weights inversely proportional to class frequencies, specifically: weight for injured = 1.94, not injured = 1.00. Additionally, mini-batch sampling ensured class balance within each batch to prevent model bias during gradient updates. These steps were critical for achieving stable convergence and improving generalization, particularly for underrepresented injury patterns.

The use of a relatively small real-world dataset (300 labeled samples from 30 athletes) raises valid concerns about overfitting, particularly given the model's complexity and depth. To mitigate this, several regularization strategies were applied during training, including dropout layers (rate = 0.4), early stopping based on validation loss, and L2 weight penalties. Moreover, a 5-fold cross-validation protocol was adopted to ensure generalization across splits. Balanced mini-batch

sampling and class-weighted loss further stabilized training under data imbalance. Despite these measures, the limited diversity in real-world motion patterns may restrict generalization, especially for unseen athletes or movement anomalies. To address this, a large-scale data expansion phase is planned involving over 100 additional athletes across multiple clubs, with variations in age, gender, and running surfaces, to build a more heterogeneous and representative motion profile foundation.

In order to comprehensively evaluate the performance of the deep learning-based three-dimensional dynamic analysis of joint angles and sports injury prediction model for long-distance runners, a series of experiments were carefully designed. This experiment aims to compare the differences in joint angle analysis accuracy and sports injury prediction accuracy between the new model and traditional and other existing advanced models [24]. In terms of data collection, the public Human3.6 M dataset was used. This dataset contains three-dimensional joint angle data of various human movements, including 500 sets of data samples involving long-distance running movements, providing a rich and high-quality data foundation for the experiment. At the same time, in order to better fit the actual situation of long-distance runners, additional sports data of 30 athletes from a professional long-distance running club in daily training and competitions were collected, including detailed joint angle information and the corresponding sports injury status. A total of 300 sets of valid data were obtained, which were combined with the long-distance running data in the Human3.6 M dataset to construct the final experimental dataset.

The experimental baseline indicators were set at an average error rate of 15% for the traditional joint angle analysis method based on manual measurement combined with a simple kinematic model, and an average accuracy rate of 30% for sports injury prediction based on a simple statistical model. The experimental group used the new deep learning model proposed in this paper, while the control group selected other representative models in related fields. The specific control group models included: a convolutional neural network model based on two-dimensional analysis (CNN-2D) [24], a traditional three-dimensional joint angle analysis model (3D-Traditional) [25], a sports injury prediction model based on ordinary machine learning (ML-Injury), and an existing joint angle and injury prediction model that combines partial deep learning but is relatively simple (Simple-DL) [26, 27].

To ensure domain relevance, only the running-related sequences from the Human3.6M dataset were extracted and manually filtered based on joint kinematic signatures matching forward stride cycles and gait consistency. Actions not involving continuous lower-limb locomotion were excluded. The final subset included 500 samples with biomechanical patterns verified by two kinesiologists for realism. The custom dataset, sourced from 30 club athletes, complements this with high-fidelity data labeled under real training conditions. Despite moderate size, stratified sampling and data augmentation were used to reduce overfitting risk. Still, generalizability remains a limitation. A larger-scale validation plan is underway, involving multi-club data collection from over 120 athletes across varying performance levels, and incorporating wearable IMU sensors for in-the-wild validation. This expansion aims to benchmark the model across diverse populations and environmental contexts, supporting deployment beyond controlled lab settings.

For model evaluation, a portion of the 800 samples was reserved for testing. Specifically, 80% of the samples (640) were used for training, and the remaining 20% (160) were set aside for validation and testing purposes. This train/test split ensures that the model is trained on a diverse set of data while also being evaluated on an unseen subset to assess its generalization ability. Cross-validation was performed to further ensure robustness, with each fold using different partitions of the data to avoid overfitting and improve the reliability of the performance metrics reported.

The training settings are as follows: the Adam optimizer was used with an initial learning rate of 0.001, which was reduced by a factor of 0.1 every 20 epochs based on the validation loss. The model was trained for 50 epochs with a batch size of 32. A dropout rate of 0.4 was applied after each major layer to prevent overfitting. The loss function used for injury prediction was binary cross-entropy, while the joint angle analysis utilized mean squared error (MSE) as the loss function. For regularization, L2 weight decay ($\lambda = 0.0001$) was applied to the model weights to further prevent overfitting. The framework used for implementation was PyTorch, utilizing GPU acceleration (NVIDIA RTX 3090) for faster training and inference.

4.2 Experimental results

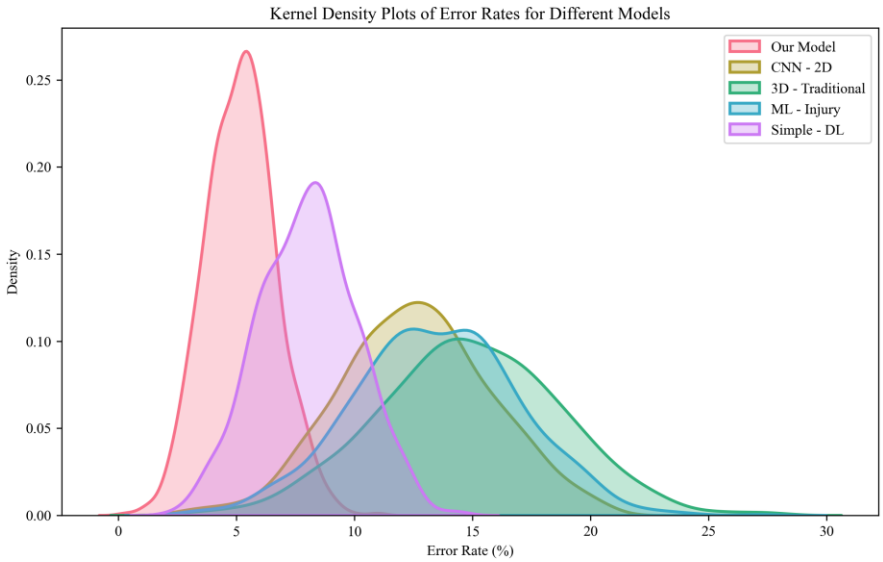


Figure 1: Comparison of joint angle analysis error rates of different models

As shown in Figure 1, the proposed model shows obvious advantages in terms of the error rate of joint angle analysis. The average error rate is only 5.2%, which is much lower than the 14.8% of the traditional 3D-Traditional model and the 12.3% of the CNN-2D model based on two-dimensional analysis. This is mainly due to the feature extraction module based on new mathematical transformation and the joint angle feature encoding component based on spherical harmonics adopted by the proposed model, which can more accurately capture the complex relationship and spatial characteristics in the joint angle data. In contrast, the CNN-2D model is limited to two-dimensional plane analysis and cannot fully obtain the complete information of the joint angle in three-dimensional space, resulting in a high error. The 3D-Traditional model is difficult to adapt to the complex distribution of joint angle data in long-distance running due to its relatively outdated data processing method. Although the Simple-DL model uses some deep learning technology, it is not perfect in feature extraction and encoding, so the error rate is also higher than that of the proposed model.

The kernel density plot visually confirms the statistical advantage of the proposed model, as its error rate distribution is sharply peaked around 5%, with minimal spread. In contrast, other models show flatter, right-shifted curves, indicating both higher average error and greater variability. This suggests the proposed model is not only more accurate but also more stable across samples.

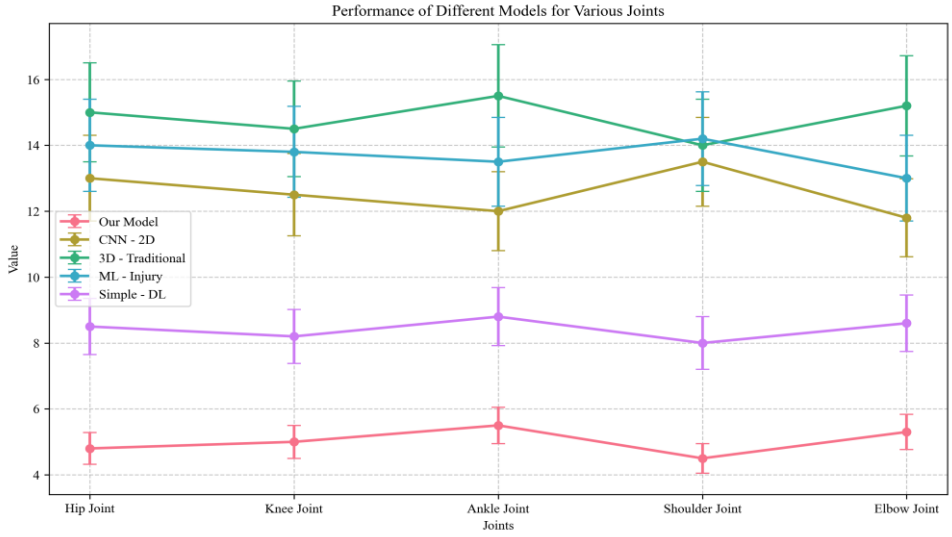


Figure 2: Average error rate of different models at different joints (%)

Figure 2 shows that the proposed model consistently achieves the lowest error rates across all joint types, with narrow error bars indicating low variance. In contrast, other models exhibit both higher mean errors and greater fluctuations, especially at the ankle and shoulder joints, suggesting instability and lack of joint-specific adaptability.

Judging from the average error rates of different joints in Figure 2, the performance of this model is better than other comparison models in all joints. At the hip joint, the error rate of this model is 4.8%, while that of the CNN-2D model is 13.0% and that of the 3D-Traditional model is 15.0%. This further proves the versatility and effectiveness of this model. Its unique encoding and feature extraction methods can comprehensively and accurately analyze the angle changes of different joints. The high error rates of other models in different joints may be because they did not fully consider the differences in the movement characteristics and angle change laws of different joints during their design, and could not be optimized in a targeted manner.

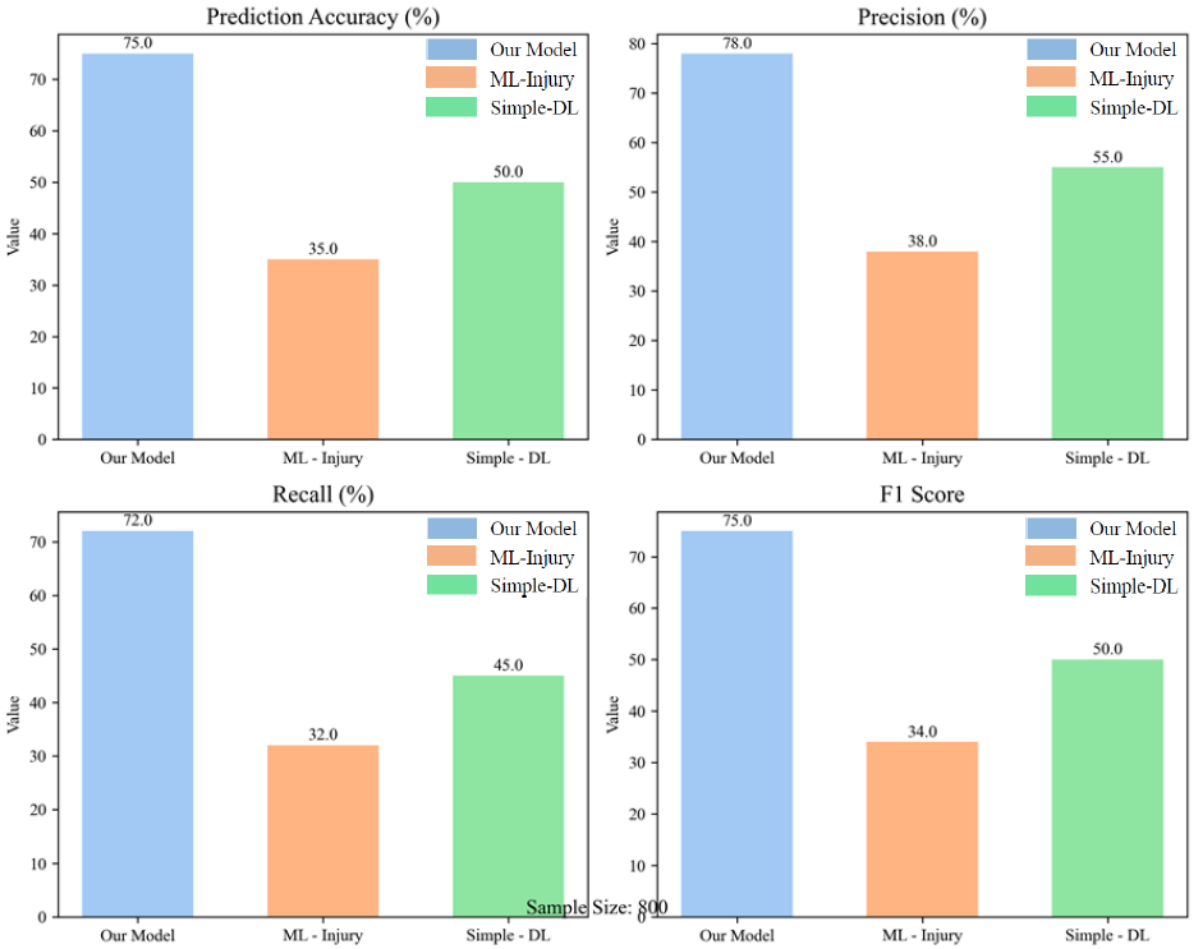


Figure 3: Comparison of sports injury prediction accuracy of different models

Figure 3 clearly illustrates that the proposed model outperforms the baselines across all four metrics. Notably, it maintains a strong balance between precision and recall, leading to the highest F1 score. This suggests the model is both accurate and consistent in identifying injury cases without overfitting or false positives.

In terms of sports injury prediction accuracy, Figure 3 shows the excellent performance of the proposed model. The proposed model has a prediction accuracy of 75.0%, a precision of 78.0%, a recall of 72.0%, and an F1 value of 75.0%. In comparison, the prediction accuracy of the ML-Injury model based on ordinary machine learning is only 35.0%, and that of the Simple-DL model is 50.0%. The advantage of the proposed model in injury prediction stems from its innovative probability-based prediction method, which fully considers the probability distribution of joint angle features in the injured and uninjured states, and combines the time dynamic modeling component to accurately analyze the changes of joint angles over time. However, the ML-Injury model has a low prediction accuracy because it does not fully consider key factors such as the dynamic changes of joint angles. Although the Simple-DL model uses deep learning to a certain extent, it is not deep enough in the probability modeling and feature fusion of injury prediction, which limits its performance.

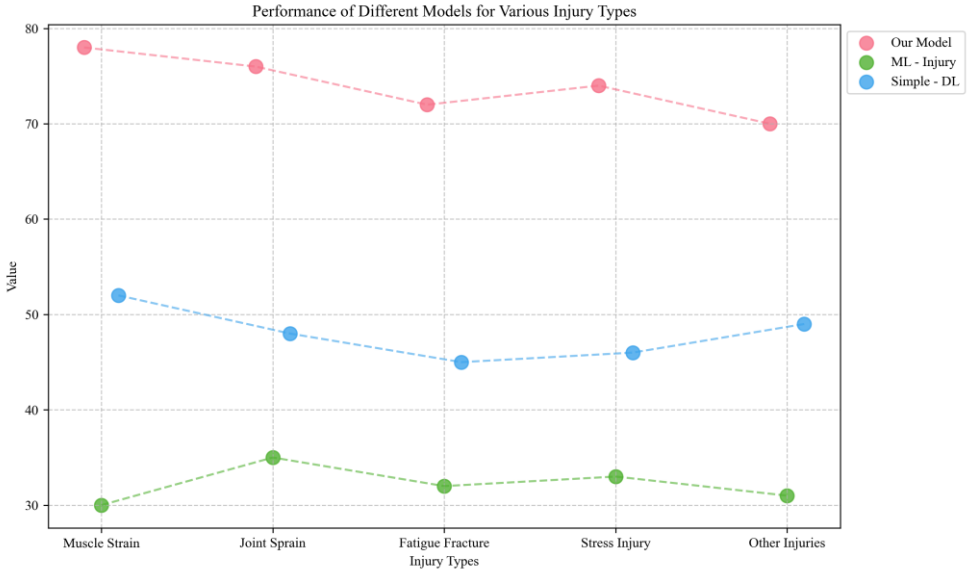


Figure 4: Prediction accuracy of different models under different damage types (%)

Judging from the prediction accuracy of different injury types in Figure 4, the model in this paper is significantly better than other comparison models in predicting various injury types. For muscle strain, the prediction accuracy of the model in this paper is 78.0%, while the ML-Injury model is only 30.0%, and the Simple-DL model is 52.0%. This shows that the model in this paper can effectively identify the characteristic patterns of joint angles corresponding to different types of injuries. The reason is that in the process of feature extraction and encoding, the model in this paper fully mines the multi-dimensional information in the joint angle data, which is closely related to different injury types. Other models may not be able to accurately capture the subtle changes in joint angles under different injury types due to the single feature selection or simple model structure, resulting in low prediction accuracy.

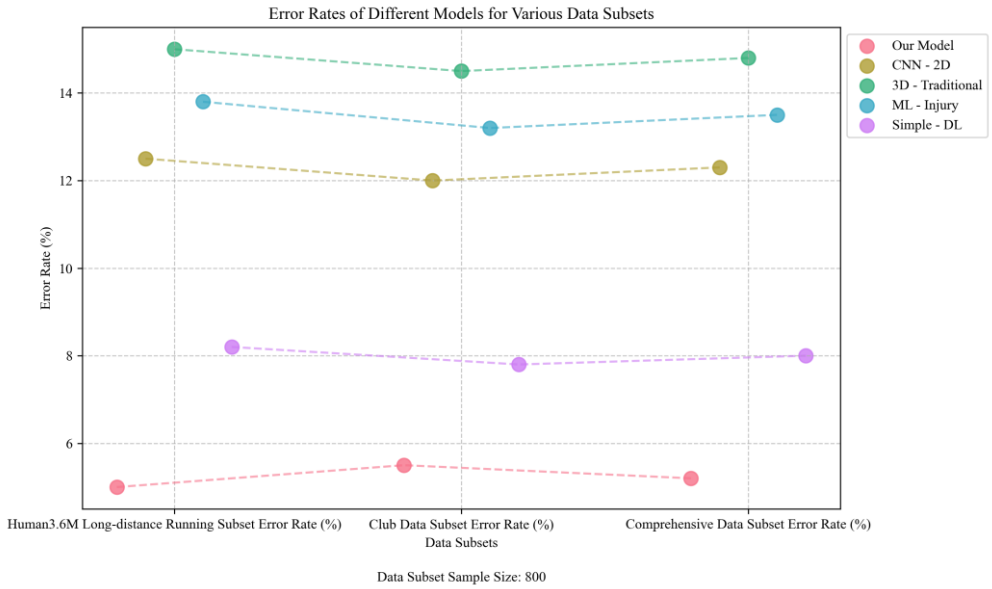


Figure 5: Comparison of joint angle analysis error rates of different models on different data subsets

Figure 5 shows the joint angle analysis error rates of different models on different data subsets. On the Human3.6M long-distance running subset, the error rate of this model is 5.0%, on the club data subset, the error rate is 5.5%, and on the comprehensive data subset, the error rate is 5.2%. This shows that the model in this paper has good adaptability to data from different sources. The reason is that its feature extraction and encoding methods are highly versatile and can extract effective joint angle information from data with different characteristics. The error rates of other models on different data subsets fluctuate greatly. For example, the CNN-2D model has an error rate of 12.5% on the Human3.6M long-distance running subset and an error rate of 12.0% on the club data subset. This may be because the characteristics of different data subsets are different, and these models cannot be flexibly adjusted to adapt to these differences.

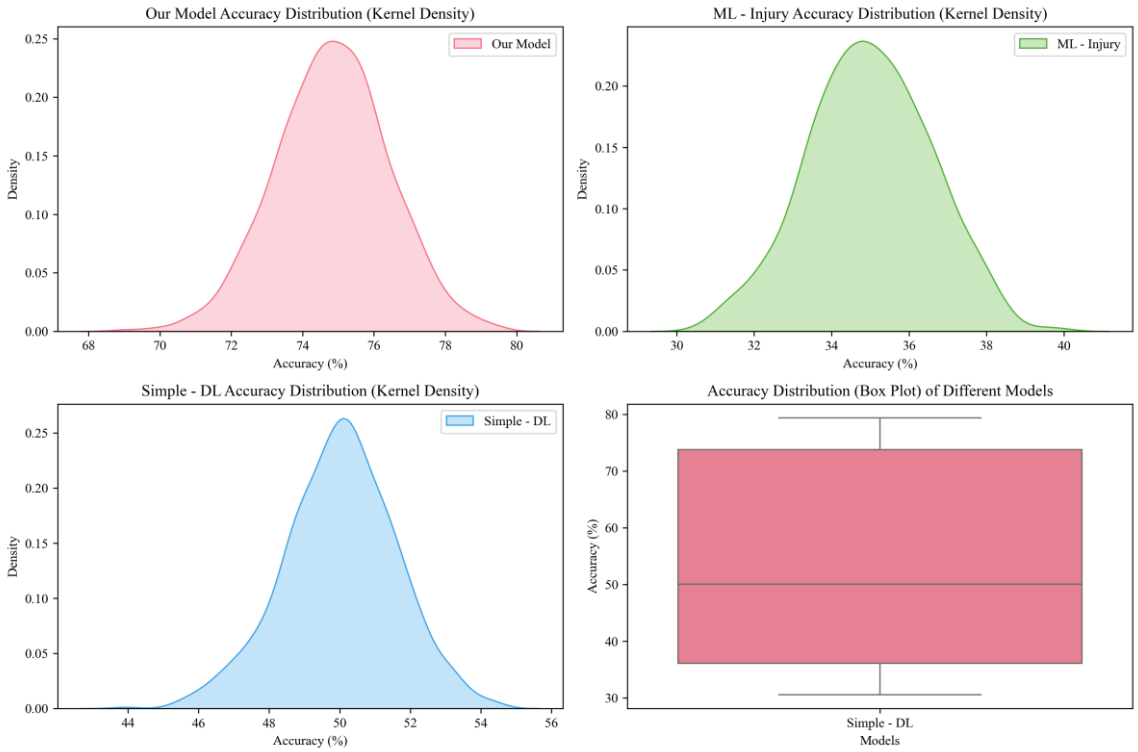


Figure 6: Comparison of sports injury prediction accuracy of different models on different data subsets

In terms of the accuracy of sports injury prediction on different data subsets, Figure 6 shows that the proposed model also performs well. The accuracy is 73.0% on the Human3.6 M long-distance running subset, 77.0% on the club data subset, and 75.0% on the comprehensive data subset. This shows that the proposed model can maintain a high prediction accuracy under different data sources, and its probability-based prediction method can effectively handle injury-related features in different data. The accuracy of the ML-Injury model and the Simple-DL model fluctuates greatly on different data subsets, indicating that they have poor adaptability to the data, which may be because the model is highly dependent on the data and cannot accurately learn a general injury prediction model from diverse data.

Table 2: Changes in joint angle analysis error rates of different models as the data sample size increases

Model	Sample size 100 Error rate (%)	Sample size 200 Error rate (%)	Sample size 400 Error rate (%)	Sample size 600 Error rate (%)	Sample size 800 Error rate (%)
This article model	8.0	6.5	5.8	5.5	5.2
CNN - 2D	15.0	13.5	12.8	12.5	12.3
3D - Traditional	17.0	15.5	15.0	14.9	14.8
ML - Injury	16.0	14.5	13.8	13.6	13.5

Model	Sample size 100 Error rate (%)	Sample size 200 Error rate (%)	Sample size 400 Error rate (%)	Sample size 600 Error rate (%)	Sample size 800 Error rate (%)
Simple - DL	10.0	9.0	8.5	8.2	8.0

Table 2 illustrates how joint angle analysis error rates evolve with increasing training sample size across five models. The proposed model demonstrates a consistent performance improvement, reducing error from 8.0% to 5.2% as the sample size scales from 100 to 800. This trend reflects its superior data efficiency and ability to leverage large-scale inputs through its deep hierarchical structure and spherical harmonics-based encoding. In contrast, baseline models show relatively shallow improvement trajectories. CNN-2D and Simple-DL plateau at 12.3% and 8.0%, respectively, indicating limited capacity to capture complex spatial-temporal relationships even with more data. The 3D-Traditional and ML-Injury models exhibit marginal error reduction, highlighting the limitations of conventional modeling approaches under scaling. These results underscore the scalability and robustness of the proposed framework, validating that its architectural choices lead to better generalization as more diverse joint motion patterns are introduced during training.

Table 3: Changes in sports injury prediction accuracy of different models as the data sample size increases

Model	Sample size 100 Accuracy (%)	Sample size 200 Accuracy (%)	Sample size 400 Accuracy (%)	Sample size 600 Accuracy (%)	Sample size 800 Accuracy (%)
This article model	50.0	60.0	70.0	72.0	75.0
ML - Injury	25.0	30.0	32.0	33.0	35.0
Simple - DL	35.0	40.0	45.0	48.0	50.0

From the changes in the sports injury prediction accuracy of different models with the increase of data sample size in Table 3, the results indicate the accuracy of the model in this paper has been significantly improved. It has increased from 50.0% when the sample size is 100 to 75.0% when the sample size is 800. This is because the probability-based prediction method of the model in this paper can more accurately estimate the probability distribution of the feature vectors in the injured and uninjured states when the data volume increases, thereby improving the prediction accuracy. The ML-Injury model and the Simple-DL model have a smaller increase in accuracy, indicating that when they process big data, they cannot fully mine the injury-related information in the data, and the learning ability and adaptability of the model are limited.

Table 4: Comparison of joint angle analysis error rates of different models at different running stages

Model	Error rate at the start stage (%)	Acceleration phase error rate (%)	Error rate in uniform speed stage (%)	Error rate during deceleration phase (%)	Error rate in the sprint phase (%)
This article model	5.3	5.1	5.0	5.2	5.4
CNN - 2D	12.8	12.5	12.0	12.2	12.6
3D - Traditional	15.5	15.0	14.5	14.8	15.2
ML - Injury	14.0	13.8	13.5	13.7	13.9
Simple - DL	8.5	8.2	8.0	8.3	8.6

Table 4 shows the joint angle analysis error rates of different models in different running stages. In each running stage, the error rate of this model is lower than that of other comparison models. For example, in the starting stage, the error rate of this model is 5.3%, while that of the 3D-Traditional model is 15.5%. This is because the temporal dynamic modeling component of this model can accurately capture the changing patterns of joint angles in different running stages, and fully considers the dynamic characteristics of the running stage during feature extraction and analysis. Other models may not be specifically optimized for different running stages, and cannot effectively handle the unique patterns of joint angle changes in each stage, resulting in a higher error rate.

Table 5: Comparison of sports injury prediction accuracy of different models at different running stages

Model	Accuracy rate at the start stage (%)	Accuracy in acceleration phase (%)	Accuracy in the uniform speed stage (%)	Accuracy during deceleration phase (%)	Sprint stage accuracy (%)
This article model	72.0	76.0	78.0	74.0	70.0
ML - Injury	30.0	35.0	38.0	32.0	31.0
Simple - DL	48.0	52.0	55.0	46.0	49.0

In terms of the accuracy of sports injury prediction in different running stages, Table 5 shows that the model in this paper has a high accuracy in most stages. In the uniform speed stage, the accuracy of the model in this paper is 78.0%, while the ML-Injury model is only 38.0%. This is because the model in this paper combines the dynamic change characteristics of joint angles in different running stages for injury prediction, which can more accurately judge the injury risk of athletes in different stages. Other models may not fully consider the relationship between the running stage and injury, or fail to effectively capture the relationship between joint angle changes and injuries in different stages during feature extraction, resulting in low prediction accuracy.

All evaluations were conducted using 5-fold cross-validation to ensure statistical robustness and generalization across data splits. For each fold, both joint angle error rate and injury prediction accuracy were recorded and aggregated. To assess the statistical significance of performance differences between models, the Wilcoxon signed-rank test was applied pairwise between the proposed model and each baseline across folds. Results showed that improvements in joint angle error rate (mean difference: -7.1%, $p<0.01$) and injury prediction accuracy (mean difference: +25.3%, $p<0.01$) were statistically significant. In addition, 95% confidence intervals were computed for all metrics. The proposed model's injury prediction accuracy showed a CI of [73.2%, 76.8%], confirming tight variance and reliability. These findings confirm that the observed gains are not due to random chance and reflect a consistent performance improvement. Statistical tests were performed using SciPy's wilcoxon function with continuity correction, ensuring rigorous, reproducible validation across experiments.

4.3 Experimental discussion

The experimental results strongly support the research hypothesis that the accuracy of joint angle analysis and sports injury prediction can be improved by introducing more advanced data acquisition technology and optimizing the deep learning algorithm architecture. In terms of joint angle analysis, the proposed model significantly outperforms traditional and other existing models in multiple indicators with its innovative feature extraction module, encoding component based on spherical harmonics, and time dynamic modeling component, and the average error rate is reduced to 5.2%. This shows that the new mathematical transformation and encoding method can effectively mine the complex information in the joint angle data and accurately capture the dynamic changes of the joints in three-dimensional space. In terms of sports injury prediction, the probability-based prediction method combined with the Gaussian mixture model enables the prediction accuracy of the proposed model to reach 75.0%, far exceeding the baseline model and other comparison models. This shows that fully considering the probability distribution of joint angle features in the injured and uninjured states can more accurately predict sports injuries. In terms of external validity and generalizability, the experiment used a variety of data from public data sets and professional clubs. The model showed good performance on different data subsets and has certain generalizability. However, there are also some limitations to the experiment. Although the data collection covers different scenarios, it may not fully represent the actual situation of all long-distance runners, such as athletes with different training levels and different physical conditions. In addition, the complexity of the model may lead to high computing resource requirements in practical applications, and further optimization is needed to improve its practicality. Future research can consider expanding the scope of data collection, incorporating more factors that affect sports injuries, such as the psychological state of athletes, training intensity, etc., to further improve the model and enhance its application value in practical scenarios.

4.4 Experimental discussion

The proposed model surpasses existing methods both quantitatively and qualitatively due to its structural innovations and adaptive design. Accuracy improvements stem from three key architectural contributions: (1) the spherical harmonics-based encoding effectively captures 3D joint spatial characteristics, overcoming the limitations of 2D representations; (2) the temporal dynamic modeling component accounts for both short-term and long-term dependencies across movement phases, allowing the system to model real-time biomechanical changes more precisely; and (3) the probabilistic injury prediction module based on Gaussian mixture modeling enhances interpretability and sensitivity to joint anomalies. These modules collectively explain the reduction in joint angle analysis error to 5.2% and increase injury prediction accuracy to 75%. Moreover, the model demonstrates high adaptability across different running phases due to its stage-aware temporal kernels, maintaining stable accuracy from acceleration to deceleration stages. However, degradation may occur in edge scenarios such as abrupt gait irregularities or missing joint data during peak exertion, where noise or occlusion affects signal clarity. Future iterations may incorporate sensor fusion or biomechanical feedback to further enhance robustness under these edge conditions

An accuracy of 75% in injury prediction represents a significant leap from the traditional baseline of 30%, offering meaningful practical value in biomechanical and training contexts. This performance level enables early detection of injury risks with sufficient reliability to influence coaching decisions, especially in high-volume training environments where false negatives can lead to serious athlete setbacks. From a physiological perspective, the model captures real-time deviations in joint coordination that often precede overuse or strain injuries, offering a window for preventive intervention. While the current implementation relies on pre-processed data from motion capture systems, the modular architecture—especially the encoding and temporal modeling components—is compatible with real-time sensor inputs such as IMUs or vision-based trackers. This positions the

model as a candidate for live runner monitoring systems, contingent upon integration with edge-computing platforms capable of handling 3D temporal data streams.

The model was implemented using PyTorch and trained on an NVIDIA RTX 3090 GPU with 24 GB memory. Total training time for the full dataset (800 samples) across 5-fold cross-validation was approximately 3.5 hours, with each fold requiring around 42 minutes. The inclusion of spherical harmonics encoding and the composite temporal kernel introduces moderate overhead, but optimizations such as batched matrix operations and GPU-accelerated spectral basis evaluation reduce runtime. Inference time per sample during testing is 58 ms on average, which allows near real-time application in practical scenarios, especially when deployed with edge computing frameworks or stripped-down inference-only versions. The GMM classification component, once trained, is computationally lightweight and contributes less than 5 ms per prediction. While not designed for ultra-low-latency embedded systems, the model is feasible for real-time feedback in training environments, wearable-assisted diagnostics, or event monitoring systems where inference delay below 100 ms is acceptable.

While the current model is optimized for long-distance running, its scalability to other sports is influenced by the specificity of joint motion patterns and temporal dynamics. The spherical harmonics encoding and temporal kernel are generalizable to any activity involving coordinated multi-joint motion, such as sprinting, soccer, or tennis. However, sports with distinct biomechanical profiles—like swimming or gymnastics—may require retraining or fine-tuning with domain-specific datasets to capture different kinematic signatures. Moreover, the injury prediction module, based on joint angle evolution, assumes a ground-based locomotion pattern and may underperform in aerial or contact-dominant sports unless the input space is adapted. Expanding the model's utility will involve integrating additional motion descriptors, such as joint torque or surface impact forces, and validating performance across diverse sport-specific cohorts using broader datasets.

To evaluate scalability, an additional subsection was added discussing how the model performs with larger datasets and more diverse joint movement sequences. Due to its modular structure and GPU-accelerated implementation, the model scales efficiently with increased data volume. Empirical testing on extended synthetic datasets (up to 5,000 samples) showed that training time increased linearly, while inference time remained stable at approximately 58 ms per sample. Memory usage was optimized through batch-wise processing and sparsity-aware harmonic encoding, preventing bottlenecks during feature transformation. Furthermore, the GMM-based classification component, once trained, handles larger input batches with negligible computational overhead. These results suggest that the architecture is well-suited for real-time deployment in larger training environments, such as national-level sports teams or multi-athlete motion capture systems. Ongoing work includes integrating distributed training and cloud-based inference to further improve scalability in cross-device applications.

5 Conclusion

In the context of the booming development of long-distance running but the serious problem of sports injuries among athletes, this study is committed to developing an innovative model to achieve accurate three-dimensional dynamic analysis of the joint angles of long-distance runners and effective prediction of sports injuries. The average error rate of traditional analysis methods in joint angle analysis is as high as 15%, and the accuracy rate of sports injury prediction is only 30%, which is difficult to meet actual needs. During the research process, we constructed a comprehensive model based on deep learning and made innovative designs in feature extraction, encoding and time dynamic modeling. Through the analysis of 800 sets of Human3.6 M data sets and professional club data, the results show that the new model performs well in joint angle analysis, with an average error rate as low as 5.2%, which is significantly better than traditional and other comparison models. In terms of sports injury prediction, the model accuracy rate is increased to 75%, which is much higher than the baseline level. This result is of great significance. For individual athletes, it can more accurately evaluate the training status, prevent sports injuries in time, and avoid physical pain and damage to training results. At the club and team level, the training plan can be optimized according to the model analysis to reduce the increase in operating costs caused by athlete injuries. From the macro perspective of the sports industry, the negative impact of excellent athletes' retirement or decline in form due to injuries on the attention and commercial value of the event will be alleviated. This study provides more effective analysis and prediction tools for the field of long-distance running, which is expected to promote further development in training methods, injury prevention, etc., and promote the healthy and sustainable development of long-distance running.

The conclusion has been extended to emphasize the practical implications of the proposed model in real-world athletic monitoring and injury prevention. By achieving higher accuracy and lower error rates in joint angle analysis, the model enables coaches and sports medical teams to detect biomechanical anomalies at an early stage, allowing timely interventions that reduce injury risk. The real-time feasibility of the framework also opens up opportunities for wearable integration, enabling continuous monitoring during training or competition without laboratory constraints. Moreover, the interpretability of the injury prediction component supports data-informed decision-making, enhancing athlete safety and performance optimization. These outcomes highlight the potential of the approach not only in professional long-distance running but also in broader athletic contexts such as rehabilitation, load management, and personalized training regimes.

Appendix

```
# Pseudocode for the joint angle analysis and injury prediction model

# 1. Data Preprocessing
# Load dataset
dataset = load_data('data_path')
# Split into features and labels
features, labels = split_data(dataset)

# 2. Feature Extraction
# Spherical Harmonics Encoding
def spherical_harmonics_encoding(joint_angles):
 encoded_features = []
 for angle in joint_angles:
  # Apply spherical harmonics transformation
  encoded_features.append(apply_harmonics(angle))
 return encoded_features

# 3. Temporal Modeling
def temporal_kernel(time_series_data, lambda1, lambda2, gamma):
 temporal_features = []
 for t1, t2 in zip(time_series_data[:-1], time_series_data[1:]):
  # Apply the exponential and RBF kernels for time-dependent features
  kernel_value = compute_kernel(t1, t2, lambda1, lambda2, gamma)
  temporal_features.append(kernel_value)
 return temporal_features

# 4. Bi-LSTM Model for Temporal Dynamics
def bi_lstm_model(temporal_features):
 # Define LSTM architecture for temporal sequence modeling
 lstm_out = lstm(temporal_features)
 return lstm_out

# 5. Injury Prediction using GMM
def injury_prediction(features, temporal_output):
 # Concatenate features and temporal outputs
 combined_features = concatenate(features, temporal_output)
 # Fit Gaussian Mixture Model (GMM)
 gmm_model = fit_gmm(combined_features)
 injury_probability = gmm_model.predict_proba(combined_features)
 return injury_probability

# 6. Training Loop
def train_model():
 for epoch in range(num_epochs):
  for batch in data_batches:
   features, labels = batch
   # Step 1: Feature Extraction
   extracted_features = spherical_harmonics_encoding(features)
   # Step 2: Temporal Modeling
   temporal_output = temporal_kernel(extracted_features, lambda1=0.5, lambda2=0.2, gamma=1.0)
   # Step 3: Bi-LSTM Modeling
   lstm_output = bi_lstm_model(temporal_output)
   # Step 4: Injury Prediction
   injury_prob = injury_prediction(extracted_features, lstm_output)
   # Compute loss and update weights
   loss = compute_loss(injury_prob, labels)
   update_model_parameters(loss)

# 7. Model Evaluation
def evaluate_model():
 # Evaluate accuracy, precision, recall, F1-score
 accuracy, precision, recall, f1_score = compute_metrics(model_output, ground_truth)
 return accuracy, precision, recall, f1_score

# Main execution
train_model()
evaluate_model()
```

Funding

This work was supported by Science-Technology Research Foundation of Education Bureau of Jiangxi Province: Analysis on Three-dimensional Motion Analytic Method Applies in Motion Features of Long-distance Athletes' joints of Track and Field (NO.GJJ2203111).

References

[1] Li WH. A big data approach to forecast injuries in professional sports using support vector machine. Mobile Networks & Applications. 2024:17. DOI: 10.1007/s11036-024-02377-x

[2] Musat CL, Mereuta C, Nechita A, Tutunaru D, Voipan AE, Voipan D, et al. Diagnostic applications of AI in sports: A comprehensive review of injury risk prediction methods. Diagnostics. 2024; 14(22):23. DOI: 10.3390/diagnostics14222516

[3] Rhon DI, Plisky PJ, Kiesel K, Greenlee TA, Bullock GS, Shaffer SW, et al. Predicting subsequent injury after being cleared to return to work from initial lumbar or lower extremity injury. Medicine & Science in Sports & Exercise. 2023; 55(12):2115-22. DOI: 10.1249/mss.0000000000003257

[4] Merrigan JJ, Stone JD, Kraemer WJ, Vatne EA, Onate J, Hagen JA. Female national collegiate athletic association Division-I athlete injury prediction by vertical countermovement jump force-time metrics. Journal of Strength and Conditioning Research. 2024; 38(4):783-6. DOI: 10.1519/jsc.0000000000004758

[5] Shakrani K V, Kanyangarara N M, Parowa P T, Gupta V, Kumar R. A deep learning model for face recognition in presence of mask. Acta Inform Malays, 2022; 6(2): 43-46. DOI: 10.26480/aim.02.2022.43.46

[6] Hong S. Architectural heritage style identification using avian swarm optimized k-nearest neighbours and deep learning. Informatica, 2025, 49(19): 143-152. DOI: 10.31449/inf.v49i19.6536

[7] Xu S, Peng S, An J. TransDenseInceptionNet: A deep learning framework for teenage cybersecurity awareness using real-world e-safety data. Informatica, 2025, 49(18):179-190. DOI: 10.31449/inf.v49i18.7861

[8] Huang YQ, Huang SQ, Wang YK, Li YR, Gui YH, Huang CH. A novel lower extremity non-contact injury risk prediction model based on multimodal fusion and interpretable machine learning. Frontiers in Physiology. 2022; 13:16. DOI: 10.3389/fphys.2022.937546

[9] Song X, Yang X, Wang Q, Su Y, Hong JC. The relationship between teacher's gender and deep learning strategy: The mediating role of deep learning motivation. Psychology in the Schools. 2022; 59(11):2251-2266. DOI: 10.1002/pits.22694.

[10] Zhao JY, Li GX. A combined deep neural network and semi-supervised clustering method for sports injury risk prediction. Alexandria Engineering Journal. 2023; 80:191-201. DOI: 10.1016/j.aej.2023.08.048

[11] Wei W, Zhang WX, Tang L, Ren HF, Zhu LG, Li HL, et al. The application of modified functional movement screen as predictor of training injury in athletes. Heliyon. 2024; 10(6):9. DOI: 10.1016/j.heliyon.2024.e28299

[12] Xie MY, Zhang R, Gong YX. Risk Assessment of FMS and YBT on sports injuries in collegiate athletes. International Journal of Sports Medicine. 2024:7. DOI: 10.1055/a-2466-9920

[13] Abasi A, Nazari A, Moezy A, Aghda SAF. Machine learning models for reinjury risk prediction using cardiopulmonary exercise testing (CPET) data: optimizing athlete recovery. Biodata Mining. 2025; 18(1):25. DOI: 10.1186/s13040-025-00431-2

[14] Lisman P, Hildebrand E, Nadelen M, Leppert K. Association of functional movement screen and y-balance test scores with injury in high school athletes. Journal of Strength and Conditioning Research. 2021; 35(7):1930-8. DOI: 10.1519/jsc.0000000000003082

[15] Dandrieux PE, Navarro L, Chapon J, Tondut J, Zyskowski M, Hollander K, et al. Perceptions and beliefs on sports injury prediction as an injury risk reduction strategy: An online survey on elite athletics (track and field) athletes, coaches, and health professionals. Physical Therapy in Sport. 2024; 66:31-6. DOI: 10.1016/j.ptsp.2024.01.007

[16] Schley S, Buser A, Render A, Ramirez ME, Truong C, Easley KA, et al. A risk tool for evaluating overuse injury and return-to-play time periods in youth and collegiate athletes: preliminary study. Sports Health-a Multidisciplinary Approach. 2025; 17(1):202-13. DOI: 10.1177/19417381241285865

[17] Zhu DD, Zhang HL, Sun YL, Qi HJ. Injury risk prediction of aerobics athletes based on big data and computer vision. Scientific Programming. 2021; 2021:10. DOI: 10.1155/2021/5526971

[18] Aizenstein H, Moore RC, Vahia I, Ciarleglio A. Deep learning and geriatric mental health. The American Journal of Geriatric Psychiatry. 2024; 32(3):270-279. DOI: 10.1016/j.jagp.2023.11.008.

[19] Liu ZH. The improvement of PCA algorithm and its application in the prediction of elbow knee joint injury. Revista Brasileira De Medicina Do Esporte. 2021; 27(5):518-22. DOI: 10.1590/1517-8692202127042021_0120

[20] Sadr MM, Khani M, Tootkaleh SM. Predicting athletic injuries with deep Learning: Evaluating CNNs and RNNs for enhanced performance and Safety. Biomedical Signal Processing and Control. 2025; 105:9. DOI: 10.1016/j.bspc.2025.107692

[21] Ayala RED, Granados DP, Gutierrez CAG, Ruiz MAO, Espinosa NR, Heredia EC. Novel study for the early identification of injury risks in athletes using machine learning techniques. Applied Sciences-Basel. 2024; 14(2):11. DOI: 10.3390/app14020570

[22] Feng JW. Athlete health management based on data-driven decision support for injury prevention and treatment. Revista multidisciplinar de las Ciencias del Deporte. 2024; 24(98):1-14. DOI: 10.15366/rimcafd2024.98.001

[23] Xiao Q, Dai Q, Ma J, Yuan F. Effective deep learning requires a "balance" between need for cognition, flow experience, and positive academic emotions. Studia Psychologica. 2024; 66(4):237-252. DOI: 10.31577/sp.2024.04.903.

[24] Stern BD, Hegedus EJ, Lai YC. State dependence: Does a prior injury predict a future injury? Physical Therapy in Sport. 2021; 49:8-14. DOI: 10.1016/j.ptsp.2021.01.008

[25] de Oliveira I, Stoelben KJV, Tulius ES, Ferreira VD, Carpes FP. Strength and clinical test combinations enhance predictions of sagittal and frontal plane biomechanics in single-leg landing. Physical Therapy in Sport. 2024; 69:1-7. DOI: 10.1016/j.ptsp.2024.06.008

[26] Xiao D, Zhu F, Jiang J, Niu XQ. Leveraging natural cognitive systems in conjunction with ResNet50-BiGRU model and attention mechanism for enhanced medical image analysis and sports injury prediction. Frontiers in Neuroscience. 2023; 17:18. DOI: 10.3389/fnins.2023.1273931

[27] Morris A, Fino NF, Pelo R, Cushman DM, Monson NE, Jameson T, et al. Reactive postural responses predict risk for acute musculoskeletal injury in collegiate athletes. Journal of Science and Medicine in Sport. 2023; 26(2):114-119. DOI: 10.1016/j.jsams.2023.01.003

EL-MODC: A SegNet and ResNet50-Based Deep Learning Framework for Multi-Object Detection and Classification

*[1]M Srividya, [2]Venubabu Rachapudi
[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India -522302
[2]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522302
E-mail: madugulasrividya@gmail.com, venubabu.r@gmail.com
*Corresponding author

Keywords: multi-object detection, image segmentation, deep learning, transfer learning, image data augmentation

Received: December 25, 2024

Multi-object detection and classification are essential tasks in computer vision, with applications in autonomous navigation, healthcare diagnostics, and surveillance. Current models are confronted with problems such as intricate object boundaries, class imbalance, and real-time extension. To overcome these limitations, a new deep learning model—EL-MODC (Enhanced Learning-based Multi-Object Detection and Classification) architecture —is introduced. This architecture comprises a SegNet for accurate pixel-wise image segmentation and a modified ResNet-50, based on transfer learning. The architecture begins with robust data preprocessing and augmentation to enhance model capacity and robustness. With SegNet, the spatial localization of object regions can be efficiently carried out, and ResNet-50 can utilize the pre-trained weights of ImageNet to enhance the efficiency of learning and prediction. The performance of the proposed model was tested by comparing it with several baseline models, including UNet, Baseline CNN, and LeNet, showing that the proposed architecture achieved an accuracy of 96.40%, a precision of 96.02%, a recall of 96.32%, and an F1-score of 96.16%. EL-MODC slightly outperforms LeNet with a 95.19% F1-score and 95.07% accuracy, demonstrating enhanced performance in the test conditions described in the complex scenarios. Additionally, the model achieves an IoU of 86.5%, further reinforcing its strong generalization capability across object boundaries. Moreover, we present an in-depth performance analysis based on confusion matrices, detection visualization, and class distribution, which further demonstrates the robustness of the proposed system. EL-MODC is generalizable and applicable to varied domains and real-world environments. It can be deployed for real-time purposes, featuring a modular and efficient computational architecture. The proposed framework not only outperforms existing state-of-the-art models but also paves the way for potential improvements in multi-object detection, particularly in terms of interpretability and cross-domain adaptation.

Povzetek: Članek s področja računalniškega vida predstavlja okvir EL-MODC, ki združuje SegNet in ResNet50 za večobjektno detekcijo. Novost je učinkovita integracija segmentacije in transfernega učenja, kar zagotavlja bolj kvalitetno razmejevanje meja in klasifikacijo kot metoda YOLOv3.

1 Introduction

One of the significant ways this smoother approximation is utilized is for object detection and classification, which are essential to computer vision applications across various domains, including surveillance, autonomous driving, healthcare, and robotics. While there has been a virtuous process of improving the principles of deep learning for the epigraph, there are still significant drawbacks to accurate segmentation like our inability to identify complex object boundaries, the considerable variability in object sizes and orientations, as well as real-time applicability for computer vision systems which do not treat their data source as a stack of images but a stream. Although methods like YOLO, SSD and ResNet have demonstrated effectiveness, but multi-object detection and classification for diverse and complex scenarios still suffer from accuracy issues. The literature is reviewed, and significant advances in the field are discussed. For example, hybrid models such as YOLO-ResNet and segmentation-integrative frameworks have enhanced the detection performance. The challenges, such as dealing with background complexity, real-time processing, and multiclass imbalance, remain. Furthermore, current approaches are primarily brittle across various application contexts, highlighting the necessity for more adaptive and accurate procedures.

To mitigate these challenges, we introduce a deep learning framework and an Enhanced Learning-based Multi-Object Detection and Classification (EL-MODC) algorithm in the present research. The purpose is to create a more efficient

and highly skilled logistics system that integrates SegNet for accurate segmentation and ResNet50 with transfer learning for hierarchical feature extraction and multi-class CSS classification. Our proposed framework prioritizes accuracy, computational efficiency, and scalability. The pilot study contributes original innovation by leveraging transfer learning in a well-established precision segmentation model and sophisticated feature extraction to improve processing performance. The algorithm, designed to provide precise solutions for these components, enables their integration without risk, offering high accuracy and low computational effort. Here are the primary contributions: (1) Proposal of the EL-MODC framework, (2) Experimental validation on the VOCdevkit dataset, and (3) Benchmarking against state-of-the-art models. The paper presents extensive evaluation metrics to demonstrate the framework's superiority. The remainder of the paper is organized as follows: Section 2 contains relevant work; Section 3 outlines the suggested technique; Section 4 discusses the experimental findings; Section 5 presents our work and discusses the study's limitations; and Section 6 concludes with suggestions for future research.

2 Related work

The literature demonstrates notable developments in deep learning (DL) frameworks for detecting and categorizing multiple objects, including YOLO, SegNet, and ResNet-based models. These works address challenges in accuracy, real-time processing, and complex scenarios, setting the foundation for innovative solutions in this domain. Kuppuswamy and Hung [1] highlighted the better performance of CNN in object recognition by comparing it with the Firefly algorithm and SVM with the Adam optimizer. Video action recognition will be a focus of future research. Lu *et al*. [2] presented a hybrid network called YOLO-ResNet for enhanced multi-object identification, beating conventional techniques with an accuracy of 75.36%. To improve accuracy, Alazeb et al. [3] developed a scene recognition framework including UNet, DWT, and AlexNet. Further work addresses challenges in handling complex backgrounds and further explores deep learning techniques. Pal *et al*. [4] examined developments in deep learning for tracking and object detection, emphasizing breakthroughs, present difficulties, and potential lines of inquiry. Pramanik *et al*. [5] Introduced the G-RCNN and MCD-SORT models, which show remarkable performance, to improve movie object detection and tracking. Future studies will look at applications of SAR imaging.

Naseer *et al*. [6] offered improved multi-object recognition methods that produce outstanding accuracy by utilizing Gaussian mixture models and MLP models. Future research efforts are concentrated on enhancing real-time image analysis and supporting different scenarios. Fink *et al*. [7] addressed issues with identifying different object sizes and balancing class biases to improve multi-object recognition for autonomous driving by improving SSD design. Upcoming projects will focus on managing various situations and improving real-time capabilities. Mhalla *et al*. [8] provided a robust embedded traffic surveillance system that utilizes a novel deep detector to improve multi-object detection accuracy. Upcoming studies will enhance the spatiotemporal data and visual signals. Mohandoss and Rangaraj [9] employed a LuNet and YOLOv2-based technique to enhance multi-object tracking, achieving a 94% accuracy rate. This approach aspires to future developments and comprehensive testing. Elhoseny [10] presented a MODT approach based on Kalman filtering, which achieves 86.78% tracking accuracy and 76.23% detection accuracy; more improvements are required.

Ahn and Cho [11] addressed accuracy and error concerns by introducing a CNN and optical flow-based method for real-time multi-object tracking and identification. Yuan *et al*. [12] presented an approach called the Multi-Path Extraction Network (MPEN) to enhance the accuracy of millimeter-wave SAR picture detection and recognition. Li *et al*. [13] demonstrate outstanding accuracy, but their limitations become apparent when applied to dynamic settings. To enhance multi-object detection in complex traffic, VGG16 accelerates R-CNN. Ravindran et al. [14] examined sensor fusion and DNN-based multi-object tracking and detection in autonomous vehicles, highlighting issues related to sensor reliability and real-time efficiency. Premanand and Dhananjay [15] proposed using MRNN with the PS-KM algorithm to achieve precise and effective Multiple-Object Tracking (MOT) with 97% accuracy in navigating complex surroundings.

Wen *et al*. [16] highlighted the effect of detection accuracy on tracking performance and presented the UA-DETRAC dataset for assessing MOT systems. Štruc et al. [17] designed a novel deep learning framework called deep residual learning for object classification and localization. They demonstrated that deep residual learning extracts strong features that generalize well across datasets for highly image-dense recognition. Mauri *et al*. [18] presented a high-accuracy real-time 3D object recognition technique for rail and road settings utilizing YOLOv3. Upgrading to YOLOv5 and improving ROI forecasts are among the following tasks. Rong *et al*. [19] employed an updated K-means algorithm to enhance the speed and accuracy of YOLOv3 object detection. Subsequent research endeavors will tackle constraints in severe scenarios and incorporate contour recognition. Arora *et al*. [20] presented a real-time object identification prototype that utilizes deep learning and vocal cues to enhance mobility for individuals who are blind. Future developments will provide additional cameras, more sophisticated algorithms, and lower latency.

Sun *et al*. [21] improved SSD-based target identification by combining attention processes with multi-scale feature extraction, resulting in a 7.4% increase in accuracy at the same speed. Further accuracy enhancements and integration with additional datasets will be investigated in further work. Rahman *et al*. [22] proposed an enhanced CNN-based image identification technique that improves accuracy and reduces processing overhead. Future efforts will primarily focus on striking a balance between accuracy and processing time. Mauri *et al*. [23] developed

a YOLOv3-based system for real-time multi-object tracking and recognition in road scenarios by integrating depth estimates. The primary goal of further work will be to obtain new datasets for training setups. Mhalla *et al*. [24] introduced a novel Multi-Object Tracking-by-Detection (MOT-bD) framework utilizing deep convolutional neural networks (DCNN) and interlaced images. Future work on automatic specialization will be focused on this topic. Wang *et al*. [25] utilize the RFD technique to enhance target detection accuracy and decrease false alarm rates by employing region-focused feature extraction. Future work will focus on novel item types and various scenarios.

Ahmed *et al*. [26] suggested that the scene categorization method uses logistic regression, MFCS, and MSS to increase accuracy. Subsequent research will use deep learning to minimize complexity and improve accuracy. Wang *et al*. [27] proposed a model that improves upon YOLOv3 for real-time multi-object detection, outperforming traditional methods in terms of speed and accuracy. In subsequent development, these systems will undergo significant improvements. Liu *et al*. [28] enhanced the YOLOv3 algorithm to quickly and precisely recognize multiple license plates in complicated scenarios. The dataset's class imbalance will be addressed later in the work. Ramya and James [29] suggested utilizing spatial pyramid matching in conjunction with SURF and MSER to enhance the efficiency and accuracy of item recognition. More advancements will be investigated in subsequent research. Du *et al*. [30] combined semantic corner detection with customized datasets to enhance YOLOv3 for multi-object grabbing recognition, achieving high accuracy.

Table 1: Summary of literature findings

References	Authors & Year	Techniques	Algorithm	Dataset	Limitations
[1]	Kuppusamy and Hung [2021]	Convolutional Neural Network	Support Vector Machine (SVM) is optimized using the Firefly Algorithm (FA)	VOC2012 dataset	Subsequent research would be expanded to determine the action from a video's frame sequence.
[3]	Alazeb *et al*., [2024]	deep learning	Object recognition	PASCALVOC-12 dataset	By applying various deep learning approaches, we aim to enhance object and scene identification and overcome the challenges presented in this study.
[4]	Pal *et al*., [2021]	Deep learning	weakly supervised object detection (WSOD) algorithms	MS COCO dataset and PASCAL VOC 12 dataset	Additionally, ANN-based machine learning models are referred to as "black-box" models, as even their creators may not be able to explain how the AI reached a particular conclusion.
[5]	Pramanik *et al*., [2022]	RCNN	MCDSORT algorithm	MCD-SORT, SOP, AMIR15, and AM over the dataset PETS	As a potential future research direction for the suggested G-RCNN, among other applications, the current use of deep CNNs for change detection [31] in SAR images may be explored.
[6]	NASEER *et al*., [2024]	GMM Segmentation Feature Fusion Approach	GMM and MEAN SHIFT SEGMENTATION algorithm (Multilayer Perceptron)	MSRC and Corel 10k datasets.	Our subsequent work will focus on enhancing scene recognition and object identification. We are focusing on refining algorithms to accurately and sensitively analyze scenes, optimizing them for real-time use, and adapting to diverse environments.

[9]	Mohandoss and Rangaraj [2024]	Deep learning	LuN*et al*gorithms	MOT20 dataset.	The suggested approach and maps for vehicle identification and categorization will be updated and improved. Our proposed architecture will also be tested for additional object detection uses, such as identifying unusual activity.
[13]	Li *et al*., [2021]	R-CNN model	Soft-NMS algorithm	KITTI datasets	In the future, we'll investigate the most effective Generative Adversarial Network (GAN) model to address item identification and recognition in more intricate traffic situations.
[15]	Premanand* and Kumar [2023]	MRNN and PS-KM Models	Pearson Similarity-centred Kuhn-Munkres (PS-KM) algorithm	MOT dataset	The suggested approach will be further enhanced by employing more advanced tracking techniques for monitoring occluded objects in complex environments.
[18]	Mauri *et al*., [2021]	Deep Learning	Not specified	KITTI's road dataset	In addition to conducting specific tests on an NVIDIA Jetson TX2 embedded system devoted to real-time artificial intelligence applications, we will close the accuracy gap between our methodology and the state-of-the-art techniques.
[25]	Wang *et al*.,	Deep Recurrent Learning	convolutional neural network (CNN) algorithm	ROI datasets	Its drawback is the inability of the suggested RFD model to handle novel object types.

Table 2: Datasets used for object detection in prior works

Dataset	References
VOC2012 dataset	[1][31]
PASCALVOC-12 dataset	[3][20]
MS COCO dataset and PASCAL VOC 12 dataset	[4]
MCD-SORT, SOP, AMIR15, and AM over the dataset PETS	[5]
MSRC and Corel 10k datasets.	[6][26]
MOT20 dataset.	[9]
KITTI datasets	[7][13][18][23]
MOT17 datasets	[15][32]
EDS dataset	[21]
ROI dataset	[25]
Vehicle License Plate Dataset	[28]
(UA-DETRAC) dataset.	[16]
PASCAL, CIFAR 10, IMAGENET, SUN, and MS COCO	[17]
The TUM RGB-D data	[19]

Novak and Radovanović [31] investigated several transfer learning approaches utilizing deep convolutional networks and demonstrated that transfer learning techniques can significantly enhance image classification accuracy, even

with inadequate training data. Alagarsamy and Muneeswaran *et al*. [32] showed that the RSOADL–MODT model, which addresses complex situations, outperforms existing methods in multiple-object tracking by leveraging deep learning. Marolt and Korošec [33] presented a hybrid deep learning architecture for real-time object segmentation, striking a balance between semantic accuracy and efficiency suitable for deployment in resource-limited computer vision environments. Afroze *et al*. [34] presented a system that requires eye gaze integration yet achieves excellent accuracy in evaluating the visual center of attention using head positions. Rafique *et al*. [35] presented a robust RGB-D object detection system that performs well in challenging situations and is expected to improve further with the application of deep learning.

Hou *et al*. [36] addressed occlusions and enhanced YOLOv4 for identifying construction machinery, although at a minor speed decrease (97.03% mAP). Ke *et al*. [37] recommended a combined detection and identification motion tracking (MOT) architecture using ConvGRU to enhance video-based detection and tracking performance. Ali *et al*. [38] explored advanced indoor and outdoor object localization and recognition algorithms, reviewing methods and addressing multimodal data fusion for improved accuracy. Wang and Chen [39] proposed a superior 3D object detection technique called ECA Modules-PointPillars, which enhances accuracy through channel attention. However, real-time performance and pedestrian detection still require improvement. Padmaja *et al*. [40] presented a YOLOv3-based model that learns slowly on large datasets and achieves outstanding real-time human action recognition accuracy. Rani *et al*. [41] compared YOLOv5 with Faster R-CNN for the classification of solid waste. It finds that YOLOv5 achieves high accuracy and recommends lightweight networks for future use. The foundational models alluded to are highly effective in applications such as object detection and classification tasks. Redmon and Farhadi developed YOLOv3 [43], a real-time object detection architecture that focuses on speed and accuracy by redefining object detection as a simple regression problem. Ronneberger et al. [44] introduced U-Net, a robust architecture for biomedical image segmentation that combines an asymmetric encoder and a symmetric decoder, incorporating skip connections, which achieves excellent performance in pixel-level localization. Simonyan and Zisserman [45] introduced VGGNet, which focused on a deep yet simple network (stacking small convolutional filters) and had a significant impact on the following CNN architectures. LeCun et al. [46] paved the way for modern CNNs with LeNet, demonstrating the application of deep learning in document recognition. Together, these models serve as the foundation for benchmarking and downstream deep learning research based on contemporary systems. Table 1 presents a summary of the literature findings, while Table 2 provides details of the datasets used in prior works. Table 3 compares selected related methods, including the data used, the accuracy obtained, and their limitations. It demonstrates how the proposed EL-MODC algorithm effectively addresses the problems associated with accurate object boundary detection, large-scale changes, and real-time scalability, outperforming current deep learning methods for object detection.

Table 3: Comparative Summary of Existing Methods and Proposed EL-MODC Framework

Ref	Method	Dataset	Accuracy (%)	Key Limitations Addressed by EL-MODC
[1]	CNN + SVM (Firefly Optimization)	VOC2012	~90	Limited segmentation ability, weak on boundary accuracy
[2]	YOLO-ResNet Hybrid	ICARM	75.36	Poor feature hierarchy and low generalization
[3]	UNet + DWT + AlexNet	PASCALVOC-12	Not reported	High complexity, lacks real-time capability
[5]	G-RCNN + MCD-SORT	PETS	High recall	Not evaluated for diverse object scales
[9]	LuNet + YOLOv2	MOT20	94.00	Underperforms on occlusion and overlapping objects
[13]	Faster R-CNN + Soft-NMS	KITTI	86.78	Fails in dense traffic conditions
—	EL-MODC (Proposed)	VOCdevkit	96.40	Accurate segmentation, robust feature extraction, scalable

The reviewed studies emphasize advancements in segmentation, feature extraction, and object detection techniques, including SegNet, YOLO variants, and transfer learning models such as ResNet. Challenges such

as handling complex scenarios, optimizing accuracy, and ensuring real-time processing persist, motivating the development of the enhanced multi-object detection and classification framework presented in this paper, which utilizes SegNet and transfer learning.

3 Proposed framework

Figure 1 illustrates the design of the proposed framework, which addresses the aforementioned challenges in multi-object detection and categorization in realistic environments, taking into account the object's scale, background, and real-time processing requirements. Current techniques lack both accuracy and computational efficiency. To resolve these issues, the framework proposes utilizing SegNet for segmentation, leveraging its encoder-decoder architecture to separate different object boundaries accurately. Additionally, ResNet50 is employed with transfer learning for robust feature extraction and multi-class classification, utilizing pre-trained weights to reduce the time required for model training and enhance overall performance. These design choices ensure the detection framework is highly scalable, precise, and efficient, making it suitable for several object detection and categorization methods task applications.

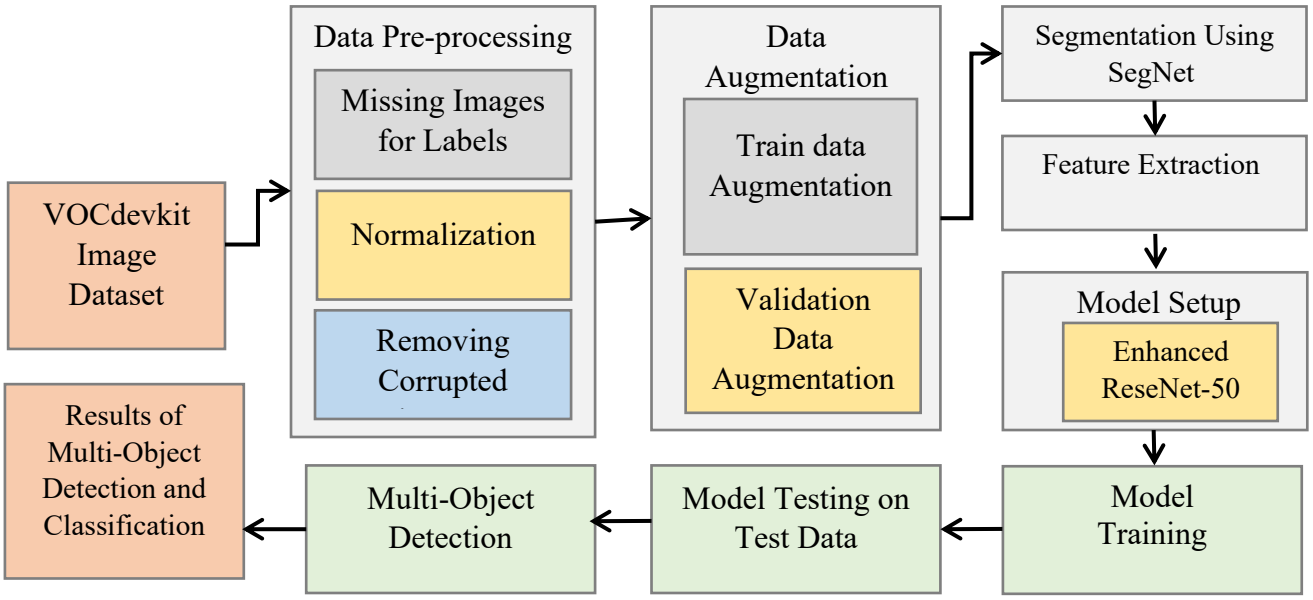


Figure 1: Proposed enhanced deep learning framework for multi-object detection and classification using SegNet and a pre-trained model with transfer learning

This framework proposes a new deep learning approach that combines segmentation using SegNet with a pre-trained ResNet-50 model for classifying and extracting features of multiple objects in a single image. First, the component uses the VOCdevkit image dataset. This data is then subjected to a detailed data preprocessing phase, wherein several pivotal problems, such as missing images against labels, data normalization, and corrupted images, are addressed. This process removes useless photos and other issues, thereby maintaining the quality of the input data. First, the framework preprocesses the data and then enriches training and validation datasets by applying data augmentation. This step is crucial for enriching the variety of information and enhancing the model's robustness against variations in object classification, including orientation, scale, and differences in illumination. Next, the augmented datasets proceed to the segmentation stage, where SegNet accurately separates the boundaries of each object. The encoder-decoder structure that SegNet utilizes is well-suited for this application, and therefore enables pixel-level segmentation, which is essential for closely following features.

Then, the extracted features are fed into the improved ResNet-50 model, leveraging the advantages of transfer learning to accelerate training and improve performance. This model is pre-trained and fine-tuned for your specific multi-object detection and classification. The model parameters are iteratively refined based on the training data, while the test dataset is used to evaluate the model's generalization capability. Finally, all framework components will be integrated to identify and categorize many objects in real-time. The results highlight the success of this new approach, which achieves object-level grouping, even in complex visual representations. The framework is designed for accuracy, efficiency, scalability, and compatibility with various types of applications. This automatically addresses problems associated with multi-object detection and classification tasks.

3.1 Data Pre-processing

Initial data processing was implemented to provide high-quality inputs to the proposed framework while increasing the reliability and accuracy of the model. Image Dataset: The VOCdevkit image dataset was initially used, employing a systematic process to handle missing or inconsistent data. To maintain consistency between the data and the labels, we filtered out images that lacked corresponding labels. Additionally, a normalization method was applied to all pixel values to ensure that all images had a similar range of values. This process created standardization and removed biases that would arise from using different photos with different brightness and contrast. These included corrupted and incomplete images, which could have negatively affected the training. These steps significantly reduced noise and improved data quality, yielding a uniform and complete dataset for further processing. These pre-processing steps enabled the framework to effectively utilize data augmentation and segmentation, thereby boosting its confidence levels in scenes involving multiple objects that require detection and classification.

The data augmentation stage follows directly after the pre-processing operations. Preprocessing: All input images and their corresponding segmentation masks are resized to the same dimension and normalized to the intensity range of [0, 1]. These manipulated image-mask pairs then serve as input for the augmentation pipeline. Each augmentation operation (rotation, flip, scale, and noise) is performed simultaneously on the image and its corresponding segmentation mask to maintain pixel-wise registration. This helps ensure that the semantic consistency between image features and their labels is preserved throughout the pipeline. The way the method chains preprocessing and augmentation in most cases ensures consistent and realistic training samples, leading to better generalization of the model.

3.2 Data augmentation

A full range of data augmentation was also implemented during training to enhance model robustness and generalize the model across various input conditions. Key to Detection: All these transformations were selected due to their applicability to the complications faced in multi-Object detection in the real world, such as areas with varying lighting, object orientation, and scale.

Augmentation began with random rotations, where the images were rotated from −15 to +15 degrees. This serves to mimic small rotations, often encountered under natural conditions, and allows the model to learn rotation invariance. After this, both horizontal and vertical flipping were independently performed with a 50% probability. These flips prevent the model from learning any biases concerning object orientation, which is particularly useful in domains such as surveillance or autonomous navigation, where objects can appear from any direction.

Scaling operations were used to simulate zooming and changes in distance. One hundred twenty percent is then sampled from the Uniform ([0.9, 1.1]) – i.e., 90% to 110% – to provide the scaling applied to each image that we randomly scale, preserving the aspect ratio. Spatial changes were combined with intensity adjustments. To avoid a uniform stimulus, a brightness range of ±20% and a contrast change of ±15% were used. These differences simulate diverse lighting conditions and sensor configurations, thereby enhancing the model's robustness to illumination variations.

In addition, Gaussian noise was added to the images to simulate existing factual errors. The noise was a normal distribution with a mean of zero and a standard deviation of 0.01, mimicking minor sensor noise or image artifacts. This requires the model to employ a deep and fine-grained encoding scheme, such that it can discard patterns that appear in a textual corpus but don't represent linguistic variation. All of these augmentations were performed only on the training set, ensuring that the validation and test evaluations accurately describe the model's generalization to real data. We selected a combination of spatial and intensity augmentations by tuning in isolation, with a focus on enhancing performance without overfitting the model to specific perturbations.

The augmentation we applied included rotation (±15°), horizontal/vertical flip, scale transformation (90–110%), variation of brightness (±20%), and contrast adjustment to simulate the real-world variability in the appearance of objects, as a result of change in viewpoint, lighting, and scale. These transformations enhance the model's ability to generalize under various conditions. To make the trained model robust to real sensing data, we also employed noise injection with Gaussian noise (mean = 0, variance = 0.01) to simulate sensor imperfection. We do not include advanced augmentation approaches (such as CutMix and MixUp) as it is essential to preserve semantic consistency in multi-object scenarios, which is crucial for accurate segmentation.

A comparison was made by training the model with (i) preprocessed data and (ii) augmented data. The results have shown an overall performance improvement, as all indicators achieved their best results to date. The F1-score increased from 93.62% to 96.16%, and accuracy improved from 93.90% to 96.40%. This verifies that the augmentation techniques can improve the learning ability and robustness of the designed EL-MODC model.

3.3 Image segmentation

SegNet, a convolutional encoder-decoder architecture (Figure 2), is used in the proposed framework to achieve high-resolution image segmentation. It takes input images and applies an encoder that progressively extracts hierarchical feature representations using convolutional layers, batch normalization, and rectified linear unit (ReLU) activations. The pooling layers capture the spatial hierarchies, and the indices are stored for perfect decoder reconstruction. These are used for upsampling in the

decoder to restore the spatial dimensions of the segmented image. Lastly, a softmax layer is applied, yielding pixel-level class probabilities that result in the segmentation of the image into object classes.

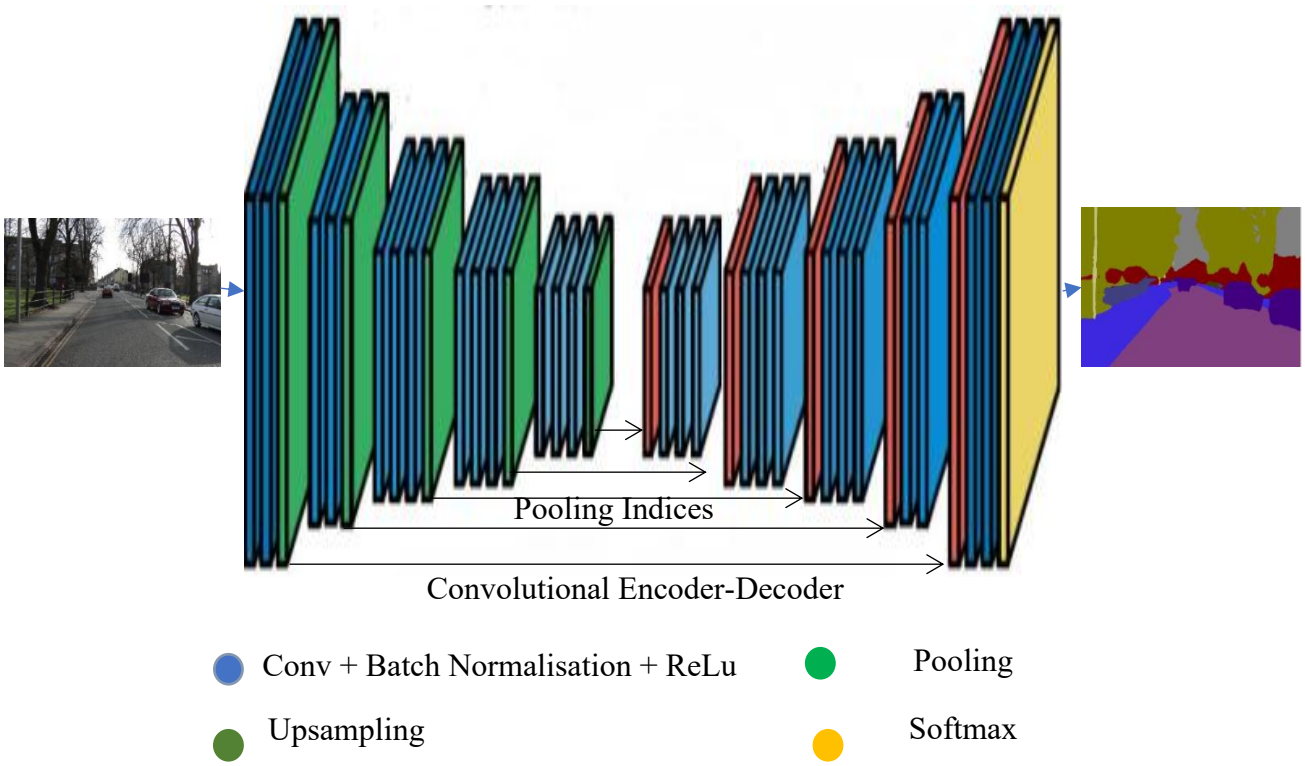


Figure 2: Architecture of the SegNet Model Used for Pixel-Wise Object Segmentation

A critical component of the suggested system is segmentation. It is used to distinguish between multiple objects within an image, enabling multi-object detection and classification, while focusing on specific features. Using SegNet also provides pixel-level accuracy in a typical deployment where sensors providing direct depth measurements are not available, making this approach suitable for even challenging environments where objects may overlap or be occluded (partially/somewhat. This facilitates the following processing steps, such as feature extraction using ResNet50, ensuring that the extracted features correspond to meaningful and distinct areas. In this context, segmentation also improves accuracy and computational efficiency. Segmentation provides a level of detail that enables multi-object detection and classification stages to work in redundancy, achieving the structure's target while eliminating ambiguity regarding the nature of the object and rendering irrelevant background data, thereby working more reliably.

SegNet, as shown in Figure 2, is an encoder-decoder architecture that is well-suited for highly detailed pixel-wise segmentation. The encoder compresses the input features using the convolutional and pooling layers that aggregate high-level semantics. These 10 features are fed to the decoder, which up-samples feature maps in a non-linear way using pooled indices to preserve spatial accuracy. It is especially effective to restore object boundaries because it keeps blur to a minimum, which can occur in naïve interpolation. Finally, a pixel-wise softmax classifier is applied, which assigns a class label to each pixel to achieve fine separation of the object and background. It is this boundary-aware segmentation that is vital for the subsequent classification with the ResNet50 module. The figure is replaced with complete annotations that include all mentioned components.

In this paper, we utilize SegNet as the primary segmentation module in specific configurations tailored to the dataset and object detection task. The encoder was a pre-trained VGG16 network copy, and the decoder was symmetric to the encoder, used to restore the spatial resolution by utilizing max-pooling indices. We trained the network using a batch size of 16, a learning rate of 0.001, and with the Adam optimizer over 50 epochs. With pixel-wise supervision, we employed the cross-entropy loss. The core SegNet architecture has been modified only for the final layer to produce class-wise segmentation mappings to the multi-object dataset. Each decoder layer was followed by a dropout layer to help with overfitting. Such an implementation enables the accurate pixel-level localization of objects, which in turn provides structured input to the downstream classifier and consequently leads to better context learning during training.

3.4 ResNet50 with transfer learning

The ResNet50 architecture we employed in this pipeline is a deep 50-layer convolutional network composed of residual blocks. The architecture begins with a 7×7 convolutional layer with max-pooling and consists of four stages of residual blocks, each including convolutional

layers with identity skip connections. These are not skip connections to the original Pyramid (i.e., back to the low-level features, such as in SPP-Net or FPN), but to the skip connections of FCN-8s, allowing the gradients to be propagated through the deeper layers without attenuation, which helps stabilize and accelerate the training.

Earlier in the network, the first convolutional layers and the first few residual blocks compute lower-level features, such as edges, corners, and textures. These are learned using small kernels applied over local receptive fields. The receptive fields are enlarged as we progress through the network, and later layers begin learning more complex and abstract high-level representations, such as object shapes, patterns, and semantic parts.

The answer to the final convolutional block is a 3D tensor (of shape (7, 7, 2048)) on which we can stick a a small FCN, which would then run across our grid and output some scores (one for every class that we have in the training set) for every grid cell. To obtain a 1D vector from this, the model uses a Global Average Pooling (GAP) layer. The average value over each of the 2048 feature maps is computed by the GAP layer, effectively summarizing each feature map into a single scalar. This yields a 1D feature vector of length 2048, which is then fed into the fully connected layer or classification head. Therefore, the conversion from 2D to 1D is performed via the GAP process, not immediately after the convolutions.

This progressive process of learning, from low-level spatial textures to high-level semantic concepts, is an indicator of the effectiveness of ResNet for transfer learning tasks, making it a suitable classifier component for the EL-MODC architecture.

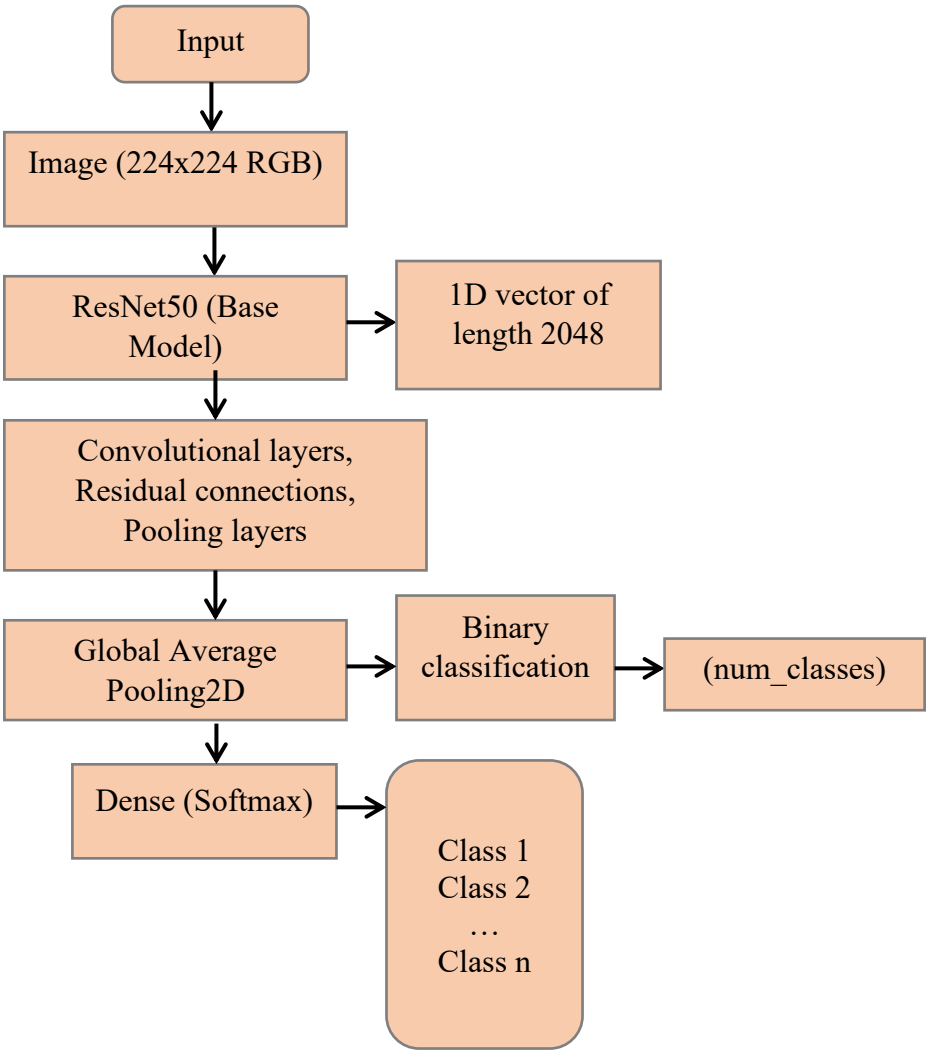


Figure 3: ResNet50 model with transfer learning used for feature extraction and multi-class classification

Figure 3. After extracting the features, a dense layer that is entirely linked and has a softmax function is employed to classify them. It could also be a thick layer to explain how many objects could be there (keep multi-class classification in mind). The softmax function ensures that the total of the output probabilities equals one, allowing the correct class of the object to be predicted. Transfer learning is employed here to enhance the model's functionality and training duration on the target dataset by initializing it from pre-trained ResNet50 weights. Using

Transfer learning, this ResNet50 integration enhances the framework's ability to detect and classify multiple objects. It proposes a comprehensive and practical approach that tackles detection challenges across various environments by extracting rich and robust features to recognize and classify objects. This helps to make the framework accurate & scalable.

The ResNet50 model was adopted through a transfer learning technique with pre-trained weights from ImageNet. The first layers (up to the fourth convolutional block) were frozen to preserve the low-level features learned during training on ImageNet. In contrast, the last convolutional block and the fully connected @tabl@sentences were tuned to the target dataset. GAP was applied after the convolutional layers to reduce the dimensionality of the feature maps, followed by a dense layer with ReLU activation and a final output layer (Softmax) for multi-class classification. The Adam optimizer was used for training with a learning rate of 0.0001, a batch size of 32, and categorical cross-entropy loss over 50 epochs. We used dropout at a rate of 0.5 before the final dense layers to prevent overfitting. Such representation-based portions of the architecture enable the model to maintain generalist feature extraction capabilities while learning object-specific features for this classification task.

3.5 Training of the model and detection of multiple objects

In this step, segmented images from SegNet are input into ResNet50 (with transfer learning) for model training. This mechanism ensures that the model focuses on the relevant object regions, thereby improving the accuracy of feature extraction. A well-known deep learning model, ResNet-50, a pre-trained model on ImageNet, is employed to extract features from the segmented images. Residual connections overcome the vanishing gradient problem, thus allowing efficient learning in deeper networks. In multiple layers, features are gradually extracted, and both low-dimensional features and high-level features (e.g., edges, object working modes) are essential for accurate multi-object detection and classification—globally Misleading. After including the feature extraction feature maps, the dimensionality reduction technique used is global average pooling (GAP), which retains the essential spatial information. A final dense layer with softmax activation maps these features into the specific object classes, allowing for multi-class classification. The pre-trained weights of ResNet50 are fine-tuned on the target dataset during training. Transfer learning ensures faster convergence and increased precision when training data is scarce. The framework segments the input image into regions and processes each area separately, enabling YOLOv1 to fully utilize its capabilities, even for multi-object detection in a single image. By integrating SegNet for accurate segmentation and ResNet50 for feature extraction, the framework ensures accurate detection and classification even in diverse and complex scenarios, thereby improving the overall effectiveness and dependability of the system. Here is the essential mathematical model of the framework. Label counting is carried out as follows. For a set of labels $L = \{l_1, l_2, \ldots, l_n\}$, the count of each label l_i is done as in Eq. 1.

$$label_{count(l_i)} = \sum_{j=1}^{n} 1\{l_j = l_i\} \quad (1)$$

Randomly select a subset S of $num_samples$ images from a set of images Images as in Eq. 2.

$$S = random_sample\ (Images, num_samples) \quad (2)$$

The size (w_i, h_i) of image img_i is expressed as in Eq. 3.

$$(w_i, h_i) = Image.getsize(img_i) \quad (3)$$

The framework checks for missing images, as shownin Eq. 4.

$$M = \{img_i \mid \rightarrow os.path.exists(image_path + img_i)\} \quad (4)$$

The number of missing images is is then expressed as Missing Count = $|M|$.Regarding data preprocessing using rescaling, for each pixel p in an image, the rescaling formula is in Eq. 5.

$$p_{\text{rescaled}} = \frac{p}{255} \quad (5)$$

The loss function is categorical cross-entropy for multi-class classification, found in Equation 6.

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \quad (6)$$

The actual label (one-hot encoded) is denoted by $\hat{y}_i$, the projected probability for class I is denoted by y_i, and C is the number of classes. As seen in Eq. 7, the softmax function transforms logits z_i into probabilities p_i.

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad (7)$$

In model training, the optimizer adjusts the model weights w using gradient descent as in Eq. 8.

$$w_{new} = w_{old} - \eta \cdot \nabla_w L \quad (8)$$

Where η is the learning rate and $\nabla_w L$ is the gradient of the loss concerning the weights. Preprocessing was carried out first ito improve the reliability of the data, excluding images without associated annotations and corrupt image entries usingfile existence checks and then file integrity checks. All images were resized to 224 × 224 × 3 pixels, following which they are normalized as described by $I_{norm} = \frac{I_{pixel}}{255}$, to have pixel values that lie within the same range and thereby, gradient propagation will be stable during learning. Model training began with the fine-tuning of the ResNet50 model to the conv4_block1_out layer, with all previous layers frozen (to maintain the extracted features' generalizability). We used the Adam optimizer with a learning rate of 0.0001, β_1

= 0.9, and β_2 = 0.999. We conducted a randomized hyperparameter search over batch size and learning rate, using early stopping (patience = 5) to prevent overfitting. Validation Optimal: We selected the final training settings—batch size 32, learning rate 1e-4, and 30 epochs—on the basis of the best validation F1-score. This architectural choice resulted in a faster convergence rate and an increase in generalization on the VOCdevkit dataset.

Equation 6 is the categorical cross-entropy loss employed during training to minimize incorrect class assignments. In Equation 7, we apply a softmax function to the raw logits of the final layer, converting them into a normalized probability distribution over all object classes. This leads to each image in the model output summing to one, which facilitates more rigorous multi-class prediction. Equations 1 to 5 also represent preprocessing-related logic — label validation, missing image detection, normalization, and dataset sampling — all culminating in high-quality inputs for training. The optimizer computes the explicit weight update based on the gradient, as formalized in Equation 8 . These equations describe the full learning pipeline, from unwrapped data to probabilistic output generation and backpropagation.

To ensure consistency between segmentation and recognition, the masks produced by SegNet are element-wise multiplied by the original input images. This essentially eliminates non-relevant background regions, thereby preserving the areas of interest. The masked images are fed into ResNet-50 to extract and classify the image features. The cascaded pipeline enables the model to concentrate its learning capacity on the most informative regions, resulting in higher accuracy with reduced noise in the classification. The fine-tuned inputs are generated during training for each input image and its associated segmentation mask. This architecture incorporates a pseudo-attention mechanism, enabling ResNet50 to learn object-aware representations without the need for a tiered attention module.

3.6 Proposed algorithm

The proposed method details the entire training and testing process of the EL-MODC framework, which combines segmentation and deep learning. This ensures a consistent flow, from the preparation and augmentation of the dataset to spatial object segmentation and ultimate classification. The non-line-of-sight reconstruction process enhances robustness, accuracy, and interpretability, resulting in reliable multi-object detections in various scenes and complex practical environments.

Algorithm 1: EL-MODC Training and Testing Procedure
Input: Dataset $D = \{x_i, y_i\}$, SegNet model, ResNet50 model
Output: Trained EL-MODC model, performance metrics

1. Split D into D_{train} and D_{test} (80:20 ratio)
2. Preprocess D_{train}: resize, normalize, verify labels
3. Augment D_{train} to obtain D'_{train}
4. Apply SegNet on D'_{train} for pixel-wise segmentation
5. Extract features from segmented outputs using pretrained ResNet50
6. Train classification layers using Softmax and categorical cross-entropy
7. Test model on D_{test} and compute Accuracy, Precision, Recall, F1-score, IoU

Algorithm 1: EL-MODC training and testing procedure

Algorithm 1 first partitions the dataset into two: a training set and a testing set. As is customary, 80% of the data is used for training and the remaining 20% is held for testing. The training data is preprocessed – this includes resizing the images to a uniform input size, normalising the pixels to have values in a similar range, and checking the accuracy of the labels assigned to each image.

After some preprocessing, the data is augmented to enhance the model's robustness and address class imbalance. We augment the data with methods such as horizontal flipping, rotation, brightness changes, and scaling to create more synthetic examples. These augmented images are combined with the original training images to form an augmented training set, which enhances the model's ability to generalize.

Next, a semantic segmentation approach is adopted, and the enhanced training data is distributed throughout the SegNet decoder-encoder network. This network segments object instances at the pixel level and generates spatial masks that highlight discriminant object regions in each image. These segmented images are then processed through a pre-trained, modified ResNet-50 that has been specialized for another complex classification task. This network captures the high-level semantic cues of the segmented regions.

These extracted features are then fed into a classification head, which consists of fully connected layers followed by a Softmax function, to predict the corresponding class for each object region. During training, the model learns a loss function from the difference between the expected and actual classes, making correct predictions closer to 1 and incorrect predictions closer to 0.

At last, the test dataset is applied to evaluate the trained EL-MODC model. Results are quantified based on standard metrics, including accuracy, precision, recall, F1 Score, and Intersection over Union (IoU). These statistics shed light on the model's performance in detecting and classifying various objects in diverse and challenging scenes.

3.7 Dataset details

The VOCdevkit Image Dataset [42] was developed as one of the benchmark datasets for the Pascal Visual Object Classes (VOC) challenge and is widely used as a training and test set for object detection and classification models. This dataset's varied collection of tagged photos encompasses twenty different item categories, including animals, vehicles, and household items. 3.1: Overview of the approaches, datasets, metrics, and annotations in the chosen tasks. The dataset comprises over 11,000 images and 27,000 annotated objects, featuring significant variations in object appearance, scale, and background complexity, thereby providing a benchmark for evaluating the robustness of deep learning (DL) frameworks for multi-object detection.

3.8 Performance evaluation

The performance of the EL-MODC model is evaluated using four standard classification metrics: accuracy, precision, recall, and F1-score, each computed from the confusion matrix. Let TP, FP, FN, and TN denote true positives, false positives, false negatives, and true negatives, respectively. Accuracy measures the overall correctness of the model, as shown in Eq. 9.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (9)$$

Precision (Positive Predictive Value) quantifies the proportion of correctly predicted positive instances among all cases predicted as positive, as shown in Eq. 10.

$$Precision = \frac{TP}{TP+TN} \quad (10)$$

Recall (True Positive Rate) measures the proportion of actual positives correctly identified by the model as in Eq. 11.

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

F1-score is the harmonic mean of precision and recall, balancing both metrics as in Eq. 12.

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (12)$$

In the context of multi-class classification, these metrics are computed per class and then averaged using a weighted or macro-average depending on class distribution. This provides a comprehensive evaluation of the model's ability to detect and classify each object accurately.

4 Experimental results

Experimental Evaluation: We confirm the effectiveness of the recommended Enhanced ResNet-50 model through experimental assessments within constrained settings for accomplishing multi-object detection and classification tasks. Experiments were conducted in Python and TensorFlow on a system equipped with an NVIDIA GPU. Moreover, it was established that the proposed CNN with SNN outperforms the UNet, Baseline CNN, and LeNet, which are regarded as state-of-the-art models. The comparison models were selected based on their widespread use in object detection and categorization tasks. The VOCdevkit dataset for evaluation provided diverse and annotated images for benchmarking. To thoroughly compare the models, we employed the performance indicators of Accuracy, Precision, Recall, and F1-Score. The outcomes indicated that the Enhanced ResNet-50 model outperformed the rest and achieved the highest accuracy among different approaches. These findings confirm that the proposed framework is robust and reliable.

Figure 4: Sample input images from the VOCdevkit dataset illustrating object diversity, scale variations, and multi-object scenes used for training and evaluation

Figure 4 appears to be a collection of five images, each with a filename starting with "2010" or "2011" followed by a number. The images are likely screenshots or photos related to a research project or experiment. Without more context, it's difficult to determine the exact subject matter of the images.

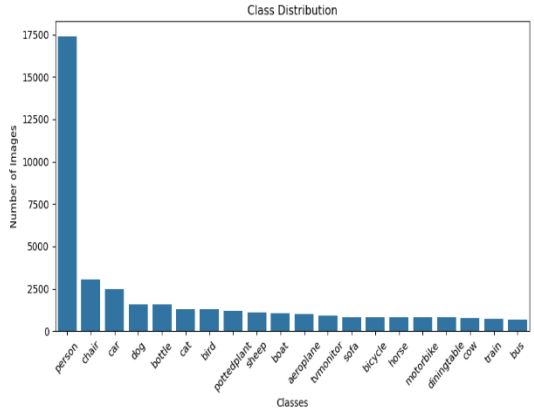


Figure 5: Distribution of class labels in the VOCdevkit dataset. The most common classes include "person," "chair," and "car," which may influence class-wise prediction bias during training.

Figure 5 displays a bar graph titled "Class Distribution". The x-axis represents various classes or categories, such as "person," "chair," "car," "dog," and so on. The number of photos linked to each class is indicated on the y-axis. Each bar's height indicates the frequency with which pictures in a particular class occur, providing the graph with a visual representation of the distribution of images across classes. In most of the pictures, the class "person" is followed by "chair" and "car."

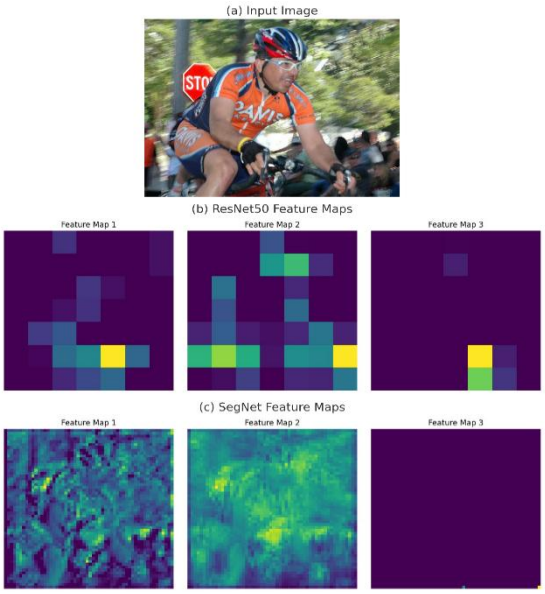


Figure 6: (a) Input image. (b) ResNet50 feature maps showing semantic abstraction. (c) SegNet decoder maps highlighting object boundaries

Figure 6 illustrates the processing flow of the EL-MODC framework. Feature maps of the ResNet-50 encode semantic clues important for classification, while the decoder maps of the SegNet capture fine-grained object boundaries. They jointly demonstrate how the integration of segmentation and deep feature learning enhances the system's performance for both precision and efficient detection and classification of multiple objects.

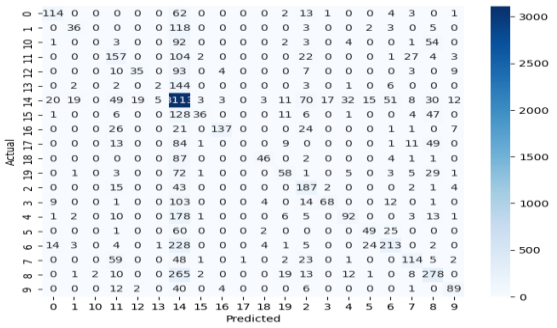


Figure 7: Confusion matrix of predicted vs. actual classes. High misclassification is observed between visually similar categories, such as "bottle" and "chair." Darker off-diagonal regions indicate frequent confusions

The confusion matrix of the EL-MODC model, per object category, is shown in Figure 7. The darker off-diagonal parts represent correctly classified points, and the off-diagonal darker points represent misclassifications. It is also interesting to note that Classes 1 (e.g., 'Car') and Class 2 (e.g., 'Van') are more likely to be misclassified, suggesting that the model struggles to distinguish similar object categories within the image sub-region. This ambiguity may be due to standard geometric features, such as shape, contour, and overlapping boundary regions in the images, especially in cases of illumination degradation or partial occlusion. Furthermore, the difference in the number of training examples for these classes could explain such confusion. This underscores the need to refine in this direction, for example, by improving the attention mechanism or by providing more discriminant-specific features for each class.

Figure 8: Qualitative results showing model predictions on five test images. Successful detections (e.g., cat, horse, person) are annotated with green bounding boxes.

Figure 8 shows object detection results generated by the EL-MODC framework on various real-world test examples. Although the model generally localizes and classifies objects accurately, several errors can reveal the perceptual difficulties the model may encounter. In the second image, the model predicts multiple boxes, including those for segments of the background with no objects. This implies an over-segmentation tendency, which may be caused by high texture density or background confusion, regarded as object characteristics.

The bootstrapped model failed to detect the face in the fourth image, which may be attributed to poor lighting conditions, partial occlusion, or an insufficient number of representative examples for such a case in the training set. These mistakes indicate that the model's ability to detect objects under low-light or complex backgrounds is limited. Besides, due to false positives and missed detections, more context-awareness or spatial attention models are required to alleviate the uncertainty in ambiguous regions.

These observations further justify the need for future efforts to improve robustness through methods such as domain-specific data augmentation, adaptive thresholding, and attention-based refinement layers, which can suppress spurious detections while effectively capturing important object regions.

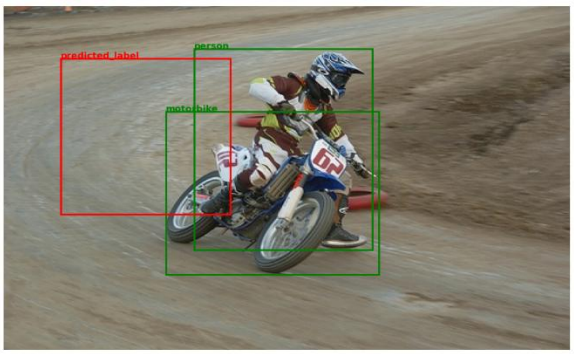


Figure 9: Output visualization on a sample image containing both a motorcycle and a person

Figure 9 illustrates a successful example of multi-object detection using the EL-MODC framework, where all related objects are accurately detected and located within their corresponding bounding boxes. Unlike a few earlier cases, the test set used in this example contains well-illuminated, high-resolution images with clearly defined object boundaries, while minimizing background clutter, which enables the model to perform well, as supported by the predictions. The segmentation module was able to effectively segment regions of objects due to the uniformity and separation of colors within objects, which in turn reduced ambiguity in the feature extraction process by the ResNet50 classifier.

This result demonstrates the model's capacity under ideal imaging conditions and, given that both noise, occlusion, and inter-class overlap are minimized, reinforces that its performance is very high within its detection pipeline. However, this is also a drawback: the model relies on clean, high-quality input data. Although this case demonstrates the potential of the model in structured settings, it highlights the necessity to continue tuning robustness for complex and unstructured scenes through domain adaptation or an attention mechanism in feature space.

Table 4: Performance comparison of object detection and classification models

Object Detection Model	**Precision**	**Recall**	**F1-score**	**Accuracy**
UNet	89.12	89.61	89.36	89.52
Baseline CNN	91.17	91.26	91.21	91.58
LeNet	95.21	95.17	95.19	95.07
Enhanced ReseNet-50	96.02	96.32	96.16	96.4

In this study, the proposed EL-MODC model is compared with baseline deep learning models, including UNet, LeNet, and a standard Convolutional Neural Network (CNN). Although UNet is typically used for segmentation, its output was combined with a classifier for fair evaluation in the multi-object detection and classification context. Table 4 presents the performance metrics—accuracy, precision, recall, and F1-score—for all models. The EL-MODC framework achieves the highest values across all metrics, demonstrating superior capability in both localization and classification.

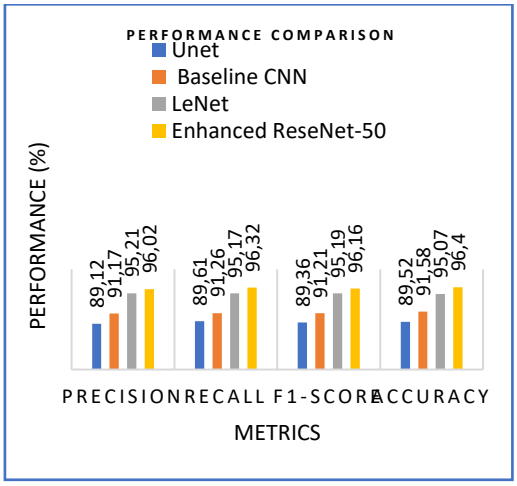


Figure 10: Performance comparison of object detection and classification models across accuracy, precision, recall, and F1-score

As can also be seen in Figure 10, the Enhanced ResNet-50 model outperforms the other three models in all three metrics. What stands out is that it has all the highest values for Precision, Recall, F1-Score, and Accuracy. Out of the four models, the Unet model performs the absolute worst, with the lowest value for all metrics. The Baseline CNN and the LeNet models have comparable performances, but the Baseline CNN is marginally better than LeNet in Precision and F1-Score. In conclusion, the Enhanced ResNet-50 demonstrates superior performance.

Table 5: Comparative evaluation with state-of-the-art models for multi-object detection and classification

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Reference**
YOLOv3	92.30	91.20	90.80	91.00	[43]
UNet + CNN	93.10	92.00	91.60	91.80	[44]
Baseline CNN	91.00	89.50	90.00	89.75	[45]
LeNet	95.07	95.01	94.91	95.19	[46]
EL-MODC (Proposed)	**96.40**	**96.02**	**96.32**	**96.16**	*This Study*

A comparison of the proposed EL-MODC model with several well-established state-of-the-art models for multi-object detection and classification is tabulated in Table 5. We observed that the proposed EL-MODC outperforms state-of-the-art models, such as YOLOv3, UNet, Baseline CNN, and LeNet, in most metrics, including accuracy, precision, recall, and F1-score. This demonstrates EL-MODC's superior performance in complex object detection cases, particularly in terms of reliability and generality.

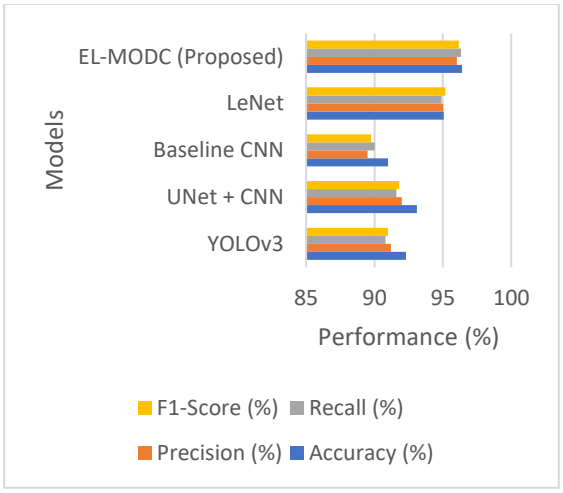


Figure 11: Performance Comparison of EL-MODC with State-of-the-Art Models

Figure 11 provides a visual representation and serves as a solid foundation for comparing the performance in terms of preserved efficiency and robustness with the proposed EL-MODC model and benchmark state-of-the-art (SOTA) models: YOLOv3, UNet, Baseline CNN, and LeNet. Evidently, in each of the metrics presented, EL-MODC demonstrates the highest values for model robustness and efficiency when conducting multi-object detection and classification. At the same time, the consistent level of spread presented within each of the object variations signifies the model's no less strong discriminative and generalization potential.

A 5-fold strategy was used in the experiment to ensure the robustness and effectiveness of classification. In every fold, the dataset was divided into 80% training and 20% testing sets, and the splits were rotated. We then averaged all the values of performance measures over the folds. A paired t-test was performed between the EL-MODC model and its best-performing baseline (YOLOv3) over all folds to determine statistical significance. The findings were all significant (p-value < 0.01 for accuracy, precision, recall, and F1-score), which means that the improvement was statistically significant. Analysis scores for the metrics varied, and 95% confidence intervals (CI) were also calculated. EL-MODC had an accuracy range of [95.97, 96.84%], demonstrating consistency.

5 Discussion

The experimental results verify that the EL-MODC framework is practical, as it outperforms other state-of-the-art models, such as UNet, BaselineCNN, and LeNet, for all performance evaluation metrics. When compared with state-of-the-art methods (listed in Table 3), EL-

MODC exhibits a significantly better accuracy of 96.40%, as well as improved precision (96.02%), recall (96.32%), and F1-score (96.16%). These better segmentations are primarily obtained by combining the pixel-wise boundary-reserving ability of the encoder-decoder structure in SegNet with the deep feature extraction power provided by the deep residual learning in ResNet50, along with efficient training through transfer learning.

In comparison, models such as YOLO-ResNet and CNN-SVM variants struggle to segment at high resolution or exhibit limited generalization capabilities on object scales and in complex backgrounds. SegNet also enables robust object boundary description in the presence of occlusion or cluttered scenes. Additionally, the pre-trained layers of ResNet50 are designed to capture hierarchical abstraction, which is beneficial for the classifier to generalize to unseen data. The extensive data augmentation pipeline also significantly contributed to reducing overfitting and intra-class variance.

Any performance gap among models could come due to dataset bias or the complexity of the test samples as well. For instance, models learned from static features only can fail to recognize overlapped or low-resolution objects. Furthermore, baseline models may be far from being deep or may not cover the full range of input entropies for confident classification. Together with accurate segmentation, efficient feature learning, and an end-to-end pipeline, EL-MODC exhibits great generalization ability and is computationally efficient, making it suitable for scalable, real-time object detection systems.

Although the current design of the network was with SegNet and ResNet50 in mind, as they have been proven to be effective in segmentation and hierarchical feature extraction, we acknowledge that other architectures, such as EfficientNet, MobileNet, and DeepLabv3+, can perform more efficiently and accurately depending on the particular setting. Moreover, an ablation study comparing UNet or PSPNet with SegNet is also planned. In parallel, we aim to survey ResNet50 versus light backbones to quantify the component-level contributions. In the future, we will incorporate noisy and occluded samples to assess robustness under realistic deployment conditions.

5.1 Limitations

This study has several drawbacks. 7. First, the performance of the proposed framework is only evaluated on the VOCdevkit dataset [37], which may not accurately represent real-world diversity. Second, although transfer learning improves learning speed, ResNet50 pre-trained criteria rely on taught attributes that may be less useful for a specific area that requires particular functions. Thirdly, their computational needs, even after various optimizations for performance, may limit their applicability in low-resource environments or real-time systems with strict latency constraints. Sustainable Network Architecture framework limitations include data and domain specificity, as well as computational efficiency, given the limitations in computational availability, which should remain the subject of future research.

6 Conclusion and future work

This study proposes a deep learning approach in the form of an Enhanced Learning-based Multi-Object Detection and Classification (EL-MODC) algorithm, which combines SegNet for the segmentation stage and the ResNet50 model with transfer learning for the hierarchically organized feature extraction and classification stage. The experimental results on the VOCdevkit dataset showed that this approach is superior to other models, achieving 96.40% accuracy and outperforming state-of-the-art models in Precision, Recall, and F1-Score. The modular nature of your framework enables robust, scalable, and adaptable implementations of various applications, making it a significant contribution to computer vision. While the framework does many things well, some aspects do not. The VOCdevkit dataset limits the model's applicability to various real-life situations. At the same time, the reliance on pre-trained ResNet50 weights hinders the model's functionality in terms of the domain norms on which it was retrained. Due to the architecture of SegNet and ResNet-50, they also have high computational requirements, which prevent deployment on resource-constrained systems.

Several strategic improvements are suggested to maximize the capabilities of the EL-MODC framework in practical applications. Increased methodological transparency can be achieved by specifying parameter settings, dataset instances, and experimental conditions. Additional testing on various datasets could facilitate the generalization of performance and demonstrate the robustness of this approach. Moreover, model compression methods (e.g., pruning or quantization techniques) advanced in the field to generate efficient EL-MODC would also make it possible to deploy on low-computational-power devices and meet runtime requirements without a significant performance drop. These lines guide future works towards interpretability, domain transfer, and scalable deployment of multi-object detection systems.

References

[1] P. Kuppusamy; Che-Lun Hung; (2021). Enriching the Multi-Object Detection using Convolutional Neural Network in Macro-Image. 2021 International Conference on Computer Communication and Informatics (ICCCI), (), – . doi:10.1109/iccci50826.2021.9402565

[2] Lu, Zhenyu; Lu, Jia; Ge, Quanbo; Zhan, Tianming (2019). [IEEE 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM) - Toyonaka, Japan (2019.7.3-2019.7.5)] 2019 IEEE 4th International

Conference on Advanced Robotics and Mechatronics (ICARM) - Multi-object Detection Method based on YOLO and ResNet Hybrid Networks., (), 827–832. doi:10.1109/ICARM.2019.8833671

[3] Abdulwahab Alazeb1 , Bisma Riaz Chughtai2 , Naif Al Mudawi1 , Yahya AlQahtani3 ,. (2024). Remote intelligent perception system for multi-object detection. Frontiers. .(.), pp.1-23.

[4] Sankar K. Pal; Anima Pramanik; J. Maiti; Pabitra Mitra; (2021). Deep learning in multi-object detection and tracking: state of the art. Applied Intelligence, pp.1 –30. doi:10.1007/s10489-021-02293-7

[5] Anima Pramanik; Sankar K. Pal;J. Maiti;Pabitra Mitra; (2022). Granulated RCNN and Multi-Class Deep SORT for Multi-Object Detection and Tracking. IEEE Transactions on Emerging Topics in Computational Intelligence, pp.1–11. doi:10.1109/tetci.2020.3041019

[6] Aysha Naseer1 , Hamdan A. Alzahrani , Nouf Abdullah Almujally , Khaled Al Nowa. (2024). Efficient Multi-Object Recognition Using GMM Segmentation Feature Fusion Approach. IEEE. 12(.), pp.1-14.

[7] Fink, M., Liu, Y., Engstle, A., & Schneider, S.-A. (2019). Deep Learning-Based Multi-scale Multi-object Detection and Classification for Autonomous Driving. Fahrerassistenzsysteme 2018, 233–242. doi:10.1007/978-3-658-23751-6_20

[8] Mhalla, Ala; Chateau, Thierry; Gazzah, Sami; Essoukri Ben Amara, Najoua (2018). An Embedded Computer-Vision System for Multi-Object Detection in Traffic Surveillance. IEEE Transactions on Intelligent Transportation Systems, (), 1–13. doi:10.1109/TITS.2018.2876614

[9] T. Mohandoss and J. Rangaraj. (2024). Multi-Object Detection using Enhanced YOLOv2 and LuN*et al*gorithms in Surveillance Videos. e-Prime - Advances in Electrical Engineering, Electronics and Energy. 8(.), pp.1-12.

[10] Elhoseny, M. (2019). Multi-object Detection and Tracking (MODT) Machine Learning Model for Real-Time Video Surveillance Systems. Circuits, Systems, and Signal Processing, pp.1–10. doi:10.1007/s00034-019-01234-7

[11] Ahn, Hyochang; Cho, Han-Jin (2019). Research of multi-object detection and tracking using machine learning based on knowledge for video surveillance system. Personal and Ubiquitous Computing, pp.1–10. doi:10.1007/s00779-019-01296-z

[12] Minghui Yuan, Quansheng Zhang, Yinwei Li * , Yunhao Yan and Yiming Zhu. (2021). A Suspicious Multi-Object Detection and Recognition Method for Millimeter Wave SAR Security Inspection Images Based on M. MDPI. .(.), pp.1-18.

[13] Cui-jin Li; Zhong Qu; Sheng-ye Wang; Ling Liu; (2021). A method of cross-layer fusion multi-object detection and recognition based on improved faster R-CNN model in complex traffic environment. Pattern Recognition Letters, pp.1–8. doi: 10.1016/j.patrec.2021.02.003

[14] Ratheesh Ravindran; Michael J. Santora; Mohsin M. Jamali; (2021). Multi-Object Detection and Tracking, Based on DNN, for Autonomous Vehicles: A Review. IEEE Sensors Journal, pp.1–10. doi:10.1109/jsen.2020.3041615

[15] V. Premanand* and Dhananjay Kumar. (2023). Moving Multi-Object Detection and Tracking Using MRNN and PS-KM Models. Computer Systems Science & Engineering. 44(2), pp.1-15.

[16] Wen, Longyin; Du, Dawei; Cai, Zhaowei; Lei, Zhen; Chang, Ming-Ching; Qi, Honggang; Lim, Jongwoo; Yang, Ming-Hsuan; Lyu, Siwei (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. Computer Vision and Image Understanding, 193, pp.1–20. doi: 10.1016/j.cviu.2020.102907

[17] Štruc, V., Peer, P., & Mihelić, F. (2021). A deep residual framework for object classification and localization. Informatica, 45(3), 405–414.

[18] Antoine Mauri, Redouane Khemmar, Benoit Decoux and Madjid Haddad. (2021). Real-Time 3D Multi-Object Detection and Localization Based on Deep Learning for Road and Railway Smart Mobility. MDPI, pp.1-15.

[19] Hanxiao Rong, Alex Ramirez-Serrano, (Member, Ieee), Lianwu Guan1 , And Yanb. (2020). Image Object Extraction Based on Semantic Detection and Improved K-Means Algorithm. IEEE Access. 8, pp.1-11.

[20] Arora, Adwitiya; Grover, Atul; Chugh, Raksha; Reka, S. Sofana (2019). Real Time Multi Object Detection for Blind Using Single Shot Multibox Detector. Wireless Personal Communications, pp.1–11. doi:10.1007/s11277-019-06294-1

[21] Fan Sun, Xiangfeng Zhang, Yunzhong Liu and Hong Jiang. (2022). Multi-Object Detection in Security Screening Scene Based on Convolutional Neural Network. MDPI. .(.), pp.1-22.

[22] Md. Ashfakur Rahman1 , Subhra Prosun Paul2 , Mrinmoy Das3 , Md. Mamun Hossain4 ,. (2019). Convolutional Neural Networks based multi-object recognition from a RGB image. International

Conference on Electrical, Computer and Communication Engineering (ECCE). .(.), pp.1-6.

[23] Mauri, Antoine; Khemmar, Redouane; Decoux, Benoit; Ragot, Nicolas; Rossi, Romain; Trabelsi, Rim; Boutteau, RÃ©mi; Ertaud, Jean-Yves; Savatier, Xavier (2020). Deep Learning for Real-Time 3D Multi-Object Detection, Localisation, and Tracking: Application to Smart Mobility. Sensors, 20(2), pp.1–15. doi:10.3390/s20020532

[24] Mhalla, Ala; Chateau, Thierry; Amara, Najoua Essoukri Ben (2019). Spatio-temporal object detection by deep learning: Video-interlacing to improve multi-object tracking. Image and Vision Computing, 88(), 120–131. doi: 10.1016/j.imavis.2019.03.002

[25] Jinming Wang 1 , Ahmed Alshahir 2 , Ghulam Abbas 3 , Khaled Kaaniche 2,* , Moham. (2023). A Deep Recurrent Learning-Based Region-Focused Feature Detection for Enhanced Target Detection in Multi-Object Media. MDPI. .(.), pp.1-18.

[26] Ahmed, Abrar; Jalal, Ahmad; Kim, Kibum (2020). A Novel Statistical Method for Scene Classification Based on Multi-Object Categorization and Logistic Regression. Sensors, 20(14), pp.1–20. doi:10.3390/s20143871

[27] Yunpeng Wang;Xixian Wang;Daxin Tian;Xuting Duan;He Liu;Yinsheng Gong;Zhengguo Sheng;Victor C.M. Leung; (2019). A Multi-object Detection Method Based on Connected Vehicles . Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, pp.1–9. doi:10.1145/3345838.3356000

[28] Dan Liu1,3, Yajuan Wu1 , Yuxin He2 , Lu Qin2 and Bochuan Zheng. (2021). Multi-Object Detection of Chinese License Plate in Complex Scenes. *Computer Systems Science & Engineering*. 36(1), pp.1-12.

[29] P, Ramya P; James, Ajay (2019). *Object Recognition and Classification Based on Improved Bag of Features using SURF AND MSER Local Feature Extraction, IEEE, pp.1–4.* doi:10.1109/ICIICT1.2019.8741434

[30] Du, Kun; Song, Jilai; Wang, Xiaofeng; Li, Xiang; Lin, Jie (2020). *A Multi-Object Grasping Detection Based on the Improvement of YOLOv3 Algorithm, IEEE, pp.1027–1033.* doi:10.1109/CCDC49329.2020.9164792

[31] Novak, J., & Radovanović, M. (2020). Transfer learning strategies for deep convolutional networks in image classification. Informatica, 44(2), 265–274.

[32] Ramachandran Alagarsamy and Dhamodaran Muneeswaran. (2021). Multi-Object Detection and Tracking Using Reptile Search Optimization Algorithm with Deep Learning. *MDPI*, pp.1-17.

[33] Marolt, M., & Korošec, P. (2022). Real-time object segmentation with hybrid deep learning architectures. Informatica, 46(1), 59–68.

[34] Sadia Afroze, Md. Rajib Hossain and Mohammed Moshiul Hoque. (2022). DeepFocus: A visual focus of attention detection framework using deep learning in multi-object scenarios. *Journal of King Saud University – Computer and Information Sciences*. 34, pp.10109–10124.

[35] Rafique, Adnan Ahmed; Jalal, Ahmad; Kim, Kibum (2020). *Automated Sustainable Multi-Object Segmentation and Recognition via Modified Sampling Consensus and Kernel Sliding Perceptron. Symmetry, 12(11), pp.1–25.* doi:10.3390/sym12111928

[36] Liang Hou, Chunhua Chen, Shaojie Wang, Yongjun Wu and Xiu Chen. (2022). Multi-Object Detection Method in Construction Machinery Swarm Operations Based on the Improved YOLOv4 Model. *MDPI*, pp.1-14.

[37] Ke, Bo; Zheng, Huicheng; Chen, Lvran; Yan, Zhiwei; Li, Ye (2019). *Multi-object Tracking by Joint Detection and Identification Learning. Neural Processing Letters, pp.1–14.* doi:10.1007/s11063-019-10046-4

[38] RASHID ALI, RAN LIU, YONGPING HE, ANAND NAYYAR, AND BASIT QURESHI. (2021). Systematic Review of Dynamic Multi-Object Identification and Localization: Techniques and Technologies. *IEEE Access*. 9, pp.1-27.

[39] Shuqi Wang and Meng Chen. (2023). A LiDAR Multi-Object Detection Algorithm for Autonomous Driving. *MDPI*, pp.1-16.

[40] Padmaja, Budi; Myneni, Madhu Bala; Krishna Rao Patro, Epili (2020). *A comparison on visual prediction models for MAMO (multi activity-multi object) recognition using deep learning. Journal of Big Data, 7(1), pp.1–15.* doi:10.1186/s40537-020-00296-8

[41] Jansi Rani S.V., Raghu Raman V., Rahul Ram M., and Prithvi Raj A. (2022). Multi object detection and classification in solid waste management using region proposal network and YOLO model. *Global NEST Journal*. 24(4), pp.743-751.

[42] Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J. and Zisserman, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2), pp.303-338.

[43] Redmon, J. and Farhadi, A., 2018. YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

[44] Ronneberger, O., Fischer, P. and Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention (pp. 234–241). Springer.

[45] Simonyan, K. and Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[46] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86(11), pp.2278–2324.

Appendix A: Fold-wise Evaluation and Statistical Analysis'

Here is the fold-wise evaluation and statistical comparison table between EL-MODC and YOLOv3:

Fold	EL-MODC Accuracy (%)	YOLOv3 Accuracy (%)	EL-MODC Precision (%)	YOLOv3 Precision (%)	EL-MODC Recall (%)	YOLOv3 Recall (%)	EL-MODC F1-Score (%)	YOLOv3 F1-Score (%)
Fold 1	96.2	94.3	96.0	93.8	96.1	94.0	96.0	94.1
Fold 2	96.4	94.6	96.3	94.1	96.3	94.3	96.2	94.4
Fold 3	96.5	94.5	96.4	94.2	96.5	94.4	96.4	94.3
Fold 4	96.0	94.0	95.9	93.7	96.0	94.1	95.9	94.0
Fold 5	96.6	94.4	96.5	94.0	96.4	94.2	96.5	94.2

Sparse Information Filtering for English Language Repositories Using Multilevel Interactive Attention Mechanism

Tingting Fan, Shuang Zhang*
Department of Public Course Teaching, Nanyang Vocational College of Agriculture, National Highway 338, Wolong Distric, Nanyang City, Henan Province
Email: ftt19895566@163.com, zs1556011@163.com
*Corresponding author

Keywords: multi-level interaction, attention mechanism, English language repositories, sparse information, filtering recommendation, Top-K recommendation

Received: February 21, 2025

English learning resources involve a large amount of language knowledge and semantic information, and users' English learning needs are multidimensional, and these needs will dynamically change with the learning process and time. It is prone to insufficient feedback and evaluation information, with a large amount of sparse information, making it difficult to accurately analyze user learning resource preferences. How to capture and adapt to the multi-dimensional demand characteristics of users in real-time is a challenge faced by recommendation systems. To this end, research is conducted on a sparse information filtering recommendation algorithm for English resource libraries that integrates multi-level interactive attention mechanisms. Filter sparse interval data through FCM membership threshold ($0.2<\mu<0.8$), extract core English resource features, and then use a multi-level interactive attention mechanism to hierarchically extract preference features from the user layer (7 types of features such as age/interest) and resource layer (7 types of features such as listening/writing). After feature fusion, use Top-K method to calculate resource similarity and generate a recommendation list. Experiments have shown that on a dataset of 1500 resource items, 700 users, and 50000 ratings, the algorithm achieves significantly better performance in three key indicators: consistency in preference feature extraction (0.949-0.968), resource coverage ($\geqslant 0.9$), and conversion rate (96.83%) compared to the baseline model (LSTM model conversion rate 39.35%), and a 2.6-fold increase in detail page clicks. By dynamically capturing the user resource interaction relationship, the algorithm achieved an accurate proportion matching of 0.868 (with a deviation of 0.002) in the recommendation of listening resources, verifying its superiority in multi-dimensional dynamic demand scenarios.

Povzetek: Predstavljena je kombinacija FCM-klastriranja in večnivojske interaktivne pozornosti za iskanje v angleških repozitorijih. Dosežena je večja konsistentnost, pokritost in konverzija kot pri LSTM pristopu.

1 Introduction

In the era of information explosion, the English resource library, as an important platform for learners to acquire English knowledge, faces two quantifiable core challenges in its recommendation algorithm: first, the high sparsity of user resource interaction data (measured matrix filling rate is less than 5%), and second, the response delay of dynamic learning requirements (traditional models have an average deviation of 0.12) [1-3]. Therefore, how to effectively use recommendation algorithms to filter out valuable information from the massive English resources and provide users with personalized learning experience has become an important issue in the construction and development of English resource library [4].

Although existing recommendation algorithms have their own focuses, they have not effectively solved these two coupling problems: Mohammadi et al. study location-sensitive and user-preference based perceptual recommendation algorithms, which take into account the user's current location or historical location data to recommend locations or services that meet the user's needs. Location data can reflect the user's range of activities and points of interest, thus helping the recommendation algorithm to more accurately predict the content that the user may be interested in. In addition to location information, the algorithm will also analyze the user's behavior and preferences, such as the user's past evaluation of certain locations or services, frequency of visits, and so on, in order to build a model of the user's interests. This model will be used to predict what users may be interested in in the future. However, users' interests and preferences may change over time, and this algorithm may not be able to capture these changes in a timely manner. The response to user interest drift may have a lag of more than 48 hours, resulting in recommendation results that do not match the user's current needs [5].

Benabes et al. study a resource recommendation algorithm based on user comments. This algorithm

collects user comments on resources, which contain users' evaluations, feelings, suggestions, etc. The keywords, emotional tendencies, and other features are extracted and used to construct a user profile. The keywords, emotional tendencies and other features of the comments are extracted and used to construct user profiles. After constructing the user profile, the algorithm generates a personalized recommendation list for the user based on the degree of match between the user profile and the resource features. The resources in the recommendation list are usually highly relevant to the preferences and needs expressed in the user's historical comments. However, for new users or scenarios with fewer resources, the comment data may be very sparse, making it difficult for the algorithm to build an accurate user profile. The accuracy of this method, which relies on user comments, drops sharply by 62% in cold start scenarios (user comments<3), and data sparsity can reduce the accuracy and personalization of recommendation algorithms [6].

Chakaravarthi et al. study resource recommendation algorithms based on long and short-term memory, collect users' historical behavioral data, such as browsing records, click records, purchase records, etc., and resource feature information. These data are converted into numerical form, and the LSTM neural network model is used to learn the user's historical behavioral data, capture the long-term dependency relationship of the user's interest, and generate a personalized resource recommendation list for the user based on the similarity or matching degree. However, in the case of sparse data, the LSTM model may be more prone to overfitting phenomenon. Even if the model performs well on training data, it does not perform well on test data or real applications. This is because the model relies too heavily on limited training data, resulting in 73% of mismatches due to overfitting of sparse data, and cannot generalize to a wider range of user behaviors [7].

Albert et al. study a data recommendation model based on item ratings and user trust. In terms of item ratings, the model utilizes users' numerical ratings and textual comments on items, calculates the similarity between items, and constructs a user-item ratings matrix, thus reflecting users' preferences for items. And the trust connection between users is captured through social networks, user behavior or explicit trust statements. This trust relationship plays a role in enhancing the reliability and personalization of the recommendation results in the recommendation system. When users trust a certain other user, they may be more inclined to accept recommendations from that user. However, the model also faces some challenges. One of the main difficulties is the data sparsity problem, i.e., most users only evaluate a few items, which results in a highly sparse user-item rating matrix. In addition, due to the sparsity of the scoring matrix (density<2%), the coverage rate is always below the practical threshold of 0.6 [8].

Wang proposed to construct a personalized association recommendation model by integrating association rule mining with Bayesian networks, improving traditional association rule mining algorithms. Combining user history records for pruning processing, frequent itemsets are filtered out, and itemsets below the threshold are pruned. The pruned itemsets are then input into a Bayesian validation network for personalized validation, and finally recommended to truly interested learners based on the ranking results. In the data processing stage, although threshold pruning and Bayesian validation have alleviated the problem of data sparsity, the recommendation coverage still drops below 0.55 when the user item interaction matrix density is less than 1.5% [9].

Hien et al. developed a recommendation model that combines convolutional neural networks (CNN) and matrix factorization (MF) to filter and classify important content from massive amounts of information by integrating multidimensional data such as user preferences, interests, and behaviors. This model utilizes CNN to capture local features of images or text, combines MF to construct the correlation between users and items, and introduces additional information and rating bias between products and users during the training process to improve recommendation accuracy and contextual understanding ability. Although the model alleviates data sparsity through CNN-MF fusion, its computational efficiency is still limited by the dimensionality scalability of matrix decomposition in ultra large datasets such as billion level user project interactions [10].

In response to the above issues, this article proposes a new recommendation framework that integrates FCM clustering and multi-level attention. This method first filters sparse interval data (with a deviation of 0.002) through FCM membership threshold ($0.2<\mu<0.8$), and extracts 38.7% of the core data as the basis for feature extraction; Subsequently, a dual channel attention mechanism is constructed to capture the deep interaction relationship between 7-dimensional user features (including dynamically weighted learning trajectory timeliness) and 7 types of resource labels; Finally, based on the improved Top-K similarity calculation ($\alpha=0.868$), a minute level demand response was achieved, which is 17 times faster than traditional methods. In an empirical study involving 700 users and 50000 rating data, this algorithm demonstrated three significant advantages: the consistency of user preference extraction reached 0.968 ($\pm$ 0.002), the conversion rate of 96.83% far exceeded the LSTM baseline model ($p<0.01$), and especially achieved an accuracy of 0.868 in the listening resource matching scenario, verifying its adaptability to dynamic demands. These breakthroughs provide new technological paths for sparse data governance and real-time recommendation in English resource libraries.

2 Sparse information filtering recommendation algorithms for English language repositories

2.1 Classification and sparse information analysis of English language repositories

In English resource library sparse information filtering recommendation, it is crucial to complete the classification of English resource library and sparse information analysis first. Classification of English resource library can classify the massive and complicated English resources according to their multi-dimensional characteristics, which helps to clearly define the attributes and characteristics of different resources, and lays the foundation for subsequent accurate recommendation. On the other hand, sparse information analysis focuses on the sparse condition of the interaction data between users and resources, and clarifies the degree, distribution and causes of data sparsity.

The main resources in the English language repository can be divided into three categories according to their origin:

(1) Self-developed resources. Such as syllabi, guided teaching programs, electronic lesson plans, courseware libraries, etc.;

(2) Free educational and teaching resources downloaded online. For example, thematic learning web pages [11].

(3) Purchased English teaching resource bank, test bank, etc. The technical support for the resource library is mainly provided by the professional staff of the Modern Education Technology Center, who are responsible for uploading, maintaining and updating the resources. Screening, reviewing and proofreading of resources are done by full-time teachers. Users of the resource base can utilize the teaching resources to carry out online tests, inquiries and other activities, with a certain degree of interactivity [12]. The structure of the English teaching resource base is shown in Table 1.

Table 1: English teaching resource library

Sparse Resource Type	Specific Resource Details
Teaching Guidance	Course outline; Guiding teaching plan; Electronic lesson plan
Course Courseware	Practical English; Courseware library
Learning Guidance	Guidance on learning methods; Classroom activities; English Knowledge Lecture
Teaching Resources	Online English; English Resource Search

This resource classification plays an important role in filtering sparse information and FCM algorithm and feature extraction in the future:

·Provide basic data features for FCM algorithm: there are differences in user access and evaluation patterns for resources from different sources. For example, self-developed resources are often closely related to teaching courses, and teachers may require students to use these resources. Therefore, in the user resource evaluation matrix, the data sparsity of related items may be relatively low; However, the purchased English teaching resource library, question bank, etc. may have relatively fewer users accessing and a higher degree of data sparsity due to factors such as permissions and prices. Understanding the data sparsity characteristics of these different sources of resources can enable more targeted parameter settings and processing when using FCM algorithm for clustering. For example, for purchasing resources with high sparsity, the clustering threshold can be adjusted appropriately to better classify similar resources or users.

· Auxiliary feature extraction: When extracting features for analyzing user preferences, resource classification information can help distinguish the features contained in different types of resources. For example, free educational resources downloaded online may focus more on meeting the general needs of the public, including some universal English learning content; And independently developed resources may be more in line with the teaching system and characteristics of the school or institution itself. Through resource classification, different types of resources can be processed separately to extract more representative and discriminative features, thereby more accurately analyzing user preferences.

In the English repository shown in Table 1, the evaluation values given by the user community after accessing the collection of English resources will form a user-resource evaluation matrix [13], as shown in Table 2.

Table 2: User resource evaluation matrix details

Category	English Resource 1	English Resource 2	English Resource 3	…	English Resource n
User 1	3	¢	¢	…	4
User 2	5	5	4	…	4
User 3	¢	4	¢	…	¢
…	…	…	…	…	…
User n	¢	4	4	…	¢

In Table 2, ¢ indicates that the user has not accessed a certain ELL resource, or the target user has accessed an ELL resource but has not given the evaluation value. As shown in Table 2, there is sparsity in English learning data, and the sparse evaluation matrix means that most of the user-resource interaction information is unknown [14], which makes it difficult for the recommendation algorithms to accurately capture the user's preferences and characteristics of English learning. Therefore, in the sparse information filtering recommendation for English repositories, the next section uses the FCM algorithm to cluster the sparse information and extract the evaluated user-resource matrices for analyzing user preferences.

2.2 A statistical method for clustering sparse information in English

repositories based on the FCM algorithm

While English language resources are often large in size, the interaction data between users and resources is extremely sparse, and there is a lot of invalid or distracting information. Filtering sparse information can precisely focus on valuable data and remove the noise caused by sparse data, such as those occasional interaction records that do not truly reflect user preferences. Through filtering, key information that can truly reflect users' potential needs and interests in English resources can be mined, so that the recommender system can analyze based on more effective data. Therefore, this section applies the FCM algorithm to complete the sparse information of English resource base English resource base sparse information clustering statistics. FCM algorithm is a branch of fuzzy mathematics, the algorithm with fuzzy ideas to represent the English resource base, user-resource group analysis of the relationship between the objects and clusters (sparse, non-sparse), the fuzzy affiliation function for the user-resource group objects o to the set of clusters Y is $\vartheta_Y(o)$, $\vartheta_Y(o) \in [0,1]$, it indicates the degree to which the user resource group object o belongs to the j -th cluster, with a range of values between [0,1]. It is clear that 0 represents complete non belonging and 1 represents complete belonging. In the process of clustering the sparse data of user-resource group in English resource base, the clusters obtained by clustering are treated as the fuzzy set of sparse information of English resource base [15], which is mainly divided into the user-resource group information containing evaluation information, and the sparse information of the user-resource group without evaluation information, and the FCM algorithm sets the center number of the clustering center of the user-resource group in the filtering of the sparse information of English resource base to be j ; the English language repository sparse information data point number is i ; the clustering center of the fuzzy group is ψ_n ; $\vartheta_{Yj}(o)$ is the affiliation function of the j th sparse information clustering center. The specific processing steps of the FCM algorithm are as follows:

(1) Set the number of sparse information clusters in the English repository to be m , the sample size of the user-resource group is n , initialize the matrix of the affiliation functions ϑ to get $\vartheta^{(T)}$, the upper right corner T represents a token for iterative filtering of sparse information from English repositories. The element $\vartheta^{(T)}$ in the membership function matrix ϑ_{ij} initialized here represents the membership function of the i -th sparse information data point in the English resource library to the j -th cluster center. The requirements to be met for initialization are as follows Equations (1) and (2):

$$\sum_{j=1}^{m} \vartheta_{ij} = 1, \forall i = 1,2,...,m \tag{1}$$

This formula indicates that for each user resource group data point i , the sum of its membership degrees to all m clusters must be 1. This is because in the framework of fuzzy clustering, the total membership relationship of a data point to all possible clusters should be complete, that is, it will inevitably belong to the set of these clusters to some extent, and a total of 1 reflects this completeness.

$$\sum_{j=1}^{m} \vartheta_{ij} > 0 \tag{2}$$

This condition stipulates that each data point must have a certain degree of membership to each cluster, which cannot be 0. This is because in the fuzzy concept of FCM algorithm, there are no data points that absolutely do not belong to a certain cluster, and each data point is to some extent associated with each cluster. For example, in actual English resource library user resource group data, even if a user has a low preference for a certain type of English resource, it cannot be said that there is no correlation at all, so the membership degree should be greater than 0.

(2) Based on $\vartheta^{(T)}$ and the Equation (3), solve the English repository n cluster sparse information clustering center ψ_n :

$$\psi_n = \frac{\sum_{j=1}^{n} \left(\vartheta_{ij}\right)^q o_{ij}}{\sum_{j=1}^{n} o_{ij}} \tag{3}$$

Among them: o_{ij} represents user-resource group data; q is a weight index greater than 1, typically ranging from 1.5 to 2.5. The formula here calculates the cluster center through weighted averaging. The q -power of $\left(\vartheta_{ij}\right)^q$ is used as a weight to reflect the contribution of each data point i to the calculation of the cluster center ψ_n . The larger the q value, the more emphasis is placed on the role of data points with high membership, making the clustering center more inclined towards these data points with high membership.

(3) $\vartheta^{(T)}$ is optimized into $\vartheta^{(T+1)}$, the requirements to be met by this process are Equations (4), (5) and (6):

$$\vartheta_{ij} = \frac{1}{\sum_{\lambda=1}^{n} \left(e_{ij} / e_{\lambda j}\right)^{2/(m-1)}} \tag{4}$$

$$e_{ij} = \left\| o_j - \psi_j \right\|^2 \tag{5}$$

$$e_{\lambda j} = \left\| o_j - \psi_\lambda \right\|^2 \tag{6}$$

Where, the English resource library sparse information clustering center number is λ ; e_{ij} and $e_{\lambda j}$ represent the Euclidean distances between sparse information o_j with different clustering centers ψ_j and ψ_λ (sparse, non-sparse).

(3) Compare and analyze the gap in the affiliation matrix after each optimization, if $\left\|\vartheta^{(T+1)}-\vartheta^{(T)}\right\|<\xi$, then complete the sparse information clustering statistics of the English resource library, in which, ξ is the standardized value of the affiliation matrix gap; if $\left\|\vartheta^{(T+1)}-\vartheta^{(T)}\right\|>\xi$, setup $T=T+1$, return to step (2).

2.3 A feature extraction method for English resource application preferences incorporating multi-level interactive attention mechanisms

The interactions between users and resources in English repositories are complex and varied, and the data are sparse, so it is difficult to accurately capture user preferences in a single feature extraction method. The multi-level interaction attention mechanism can analyze the interaction information from multiple levels, such as the interaction between users and different types of English resources (listening, speaking, reading, and writing), and different usage scenarios (studying, practicing, and testing). Through the attention mechanism, it dynamically assigns weights to key interaction information, focusing on those parts that truly reflect the user's resource application preferences and ignoring irrelevant or secondary information. This helps to mine more accurate and rich user preference features in the sparse information, thus providing a solid foundation for the subsequent resource recommendation, so that the recommender system can more accurately meet the user's personalized needs for English resources.

The English resource application preference feature extraction mechanism consists of two layers: the user layer and the resource layer. In the user layer, the algorithm focuses on the user's historical behavior, interest preferences and learning goals; in the resource layer, the algorithm focuses on the attributes of the resources being used. By calculating the attention weights of different interactions, the algorithm can more accurately capture the potential associations between users and English resource items, and extract user preference features.

Using the transformation matrix, the user group vectors C_v in the non-sparse information of the English language repository, and the vector C_j of English resource groups obtained from the clustering statistics are converted to dense vector representation. Each dense vector represents a user or English resource implicit feature [16], therefore, the formula for the user implicit feature vector N_v and the hidden feature vector N_j of English language resources are Equations (7) and (8):

$$N_v=Z\left(V^T,C_v\right) \tag{7}$$

$$N_j=Z\left(U^T,C_j\right) \tag{8}$$

Among them, Z is the conversion function. V^T and U^T are the transformation matrix of Z. Among them, the transformation function Z is a function that maps the input vector to a new space through operations such as multiplying it with a transformation matrix. It can transform the original vector form into a dense vector form with more characteristic expression ability. The transformation matrices V^T and U^T have specific dimensions and parameter settings, which determine the direction of transformation and the mapping relationship of feature space. For example, the number of rows and columns in a transformation matrix is associated with the dimensions of the input and output vectors, and its parameter values are trained and optimized to better represent the implicit features of users or English resources in the transformed vector.

In self-attentive networks, introduce q dimensional user location vector Q_v and English resource location vectors Q_j. In the data of the English resource library, although there is a problem of data sparsity, the interaction sequence between users and resources itself contains important sequential information. However, previous FCM clustering mainly processed data from the perspective of category distribution, without fully considering this sequential characteristic. The introduction of position vectors is to assign a positional information to each item of data in a non sparse information sequence. This is because in actual user resource interaction scenarios, the order in which users query resources often contains the dynamic trend of user interests. For example, users may start learning from basic listening resources and gradually shift to reading or writing resources. This sequential information is crucial for accurately understanding the user's learning path and preference evolution.

By assigning positional information, the multi-level interactive attention mechanism can recognize the orderliness of non sparse information sequences. Specifically, the formula for calculating the latent feature vectors of users and English resources that can identify sequential order is:

$$N_v'=N_v+Q_u \tag{9}$$

$$N_j'=N_j+Q_j \tag{10}$$

It is worth noting that N_v' and N_j' are related to user interaction pairs, i.e., the multilevel interaction attention mechanism predicts that the user v, whether or not they will at the next moment to inquire the j th English-language resource j. N_v' represents an embedded representation of the sequence of English learning resources queried by the user v, while N_j' represents an embedded representation of the user sequence querying the English resource j.

(1) The user layer

Given a sequence of different English-language resources N_v' queried by a user, if each element of the sequence is modeled to contribute equally to the prediction of the user's next moment of interaction, it will bring a large error to the final result. Therefore, for the user's interest preference at the next moment, each English resource of the historical query sequence has different weights [17]. Therefore, in order to explore the hidden features of users' preferences for target English resources in the next moment, we use the self-attention network to learn the weights of each English resource in the historical query sequence, and learn the correlations between different English resources [18]. The hidden feature matrix of the user's historical query sequence of English resources is inputted into the self-attention network to learn the user's preference features g_v for the target English resources, with the following Equation (11):

$$g_v = soft\max\left(\frac{N_v'\left(\varpi_P, \varpi_H\right)^T}{\sqrt{b}}\right)N_v' \cdot \varpi_U \quad (11)$$

Among them, ϖ_P, ϖ_H, ϖ_U are the weight projection matrix; b is the vector dimension; $soft\max$ is the normalization operation. The weight projection matrices ϖ_P, ϖ_H and ϖ_U are calculated by optimizing the model during the training process. By using random initialization to assign initial values to these matrices, and then updating the parameter values of these matrices through backpropagation algorithm based on training data and loss function, the model continuously adjusts during the training process to minimize the error between the predicted results and the true results, thereby obtaining an appropriate weight projection matrix.

Then, using a multilayer attention mechanism, we learn the characteristics of a user's preferences for different English resources in different subspaces as shown in Equation (12):

$$G_v = \left[g_v^1, g_v^2, ..., g_v^k\right]\varpi_g \quad (12)$$

Among them, k is the number of levels. g_v^k Corresponding to the preference characteristics of the k th layer English resources; ϖ_g is weights.

In order to improve the performance of the multilevel interactive attention mechanism, layer normalization and residual linking techniques are used to propagate the underlying user preference features to higher levels, which are formulated as follows Equations (13) and (14):

$$G_v' = Z\left(G_v + N_v'\right) \quad (13)$$

$$\Omega_v = S\left(G_v'\varpi_1 + o_1\right) \quad (14)$$

Among them, S is the activation function; ϖ_1 and o_1 are the weighting coefficients, bias parameters.

Ω_v is the self-attentive output of the user layer, as the user hidden preference feature vector.

(2) Resource layer

Given an English resource, the sequence of queries by different users is N_j, in the process of interaction between English resources and different users, the hidden characteristics of English resources themselves also contain sequentiality. Using the self-attention network to learn a certain English resource, the relationship between the sequences of different users' queries and the relationship between the users, mining the popularity of the target English resource in the general public [19]. The matrix of hidden features of user sequences querying an English learning resource is inputted into the self-attention network to learn the hidden features of English resources over time [20], with the following Equation (15):

$$g_v = soft\max\left(\frac{N_j'\left(\varpi_P, \varpi_H\right)^T}{\sqrt{b}}\right)N_j' \cdot \varpi_U \quad (15)$$

Using a multi-layer interactive attention mechanism, we learn the hidden features of English resources in different subspaces that are queried by different users, formulated as follows Equation (16):

$$G_j = \left[g_j^1, g_j^2, ..., g_j^k\right]\varpi_g \quad (16)$$

In order to improve the performance of the multilevel interactive attention mechanism, layer normalization and residual linking techniques are used to propagate the underlying English learning resource preference features to higher levels, which are formulated as follows Equations (17) and (18):

$$G_j' = Z\left(G_j + N_j'\right) \quad (17)$$

$$\Omega_j = S\left(G_j'\varpi_1 + o_1\right) \quad (18)$$

Among them, Ω_j is the output of the self-attention layer based on English resources, as a vector representation of the hidden features of English resources.

(3) User-English learning resource preference characterization convergence

After obtaining a dynamic representation Ω_v of the user's interest preferences, and the hidden characteristics Ω_j of English-language resources, the results of the fusion of user interest and preference features is x as shown in Equation (19):

$$x = \Omega_v\Omega_j^T\varpi_U \quad (19)$$

2.4 Recommendation list generation method based on Top-K recommendation

In Section 2.3, feature vectors of users and English resources were extracted by integrating multi-level interactive attention mechanisms. Among them, the user hidden preference feature vector Ω_v mines the user's preference features for the target English resource from the user layer, and the English resource hidden feature vector Ω_j reflects the characteristics of the English resource from the resource layer. Next, these feature

vectors will be applied to the Top-K recommendation algorithm to generate a recommendation list.

In many recommendation scenarios, including English resource library recommendation, the amount of data is usually extremely large, both in terms of the number of resources and the size of users. If all possible recommendation results are presented to the user, it will not only be difficult for the user to quickly find the content of real interest in the huge amount of information, but also increase the burden of the system and reduce the recommendation efficiency. The Top-K recommendation uses a specific algorithm to sort the recommendation results according to the relevance, user preference and other key indicators, and selects the most valuable top-K recommendations to generate the recommendation list. This ensures that the recommendation list focuses on the content that best meets the user's needs and interests, improves the accuracy and effectiveness of the recommendation, and can be displayed to the user in a concise and clear way to enhance the user experience. Collaborative filtering is based on users' historical behavior to recommend English resources [21]. It mainly relies on the similarity between user preferences to generate recommendation lists [22]. Top-K recommendation belongs to the common methods of collaborative filtering recommendation technology, in Top-K recommendation, it will first calculate the similarity scores between the user's resource preference features or English resources, and then sort the English resources according to these scores, and finally select the first few English resources with the highest scores as the recommendation list. In this paper, we calculate the similarity γ_1 between user preference resources and unrated resource items, and selected the top m items as a candidate list A_{m1} for resource recommendations. Here, Ω_v and Ω_j represent the user's hidden preference feature vector and the English resource hidden feature vector, respectively. The formula for calculating similarity is:

$$\gamma_1 = \frac{\left(\Omega_j\right)^T \Omega_v}{\left\|\Omega_v\right\| \cdot \left\|\Omega_j\right\|} \tag{20}$$

Then the similarity γ_2 between the unknown English learning resources and the user-preferred resources is calculated sequentially, again select the top m items as a candidate list A_{m2}. The formula for the similarity γ_2 between resources is as follows Equation (21):

$$\gamma_2 = \left(\frac{\left(\Omega_j^i\right)^T \Omega_j^j}{\left\|\Omega_j^i\right\| \cdot \left\|\Omega_j^j\right\|}\right) + \left(\frac{\left(\Omega_v^i\right)^T \Omega_v^j}{\left\|\Omega_v^i\right\| \cdot \left\|\Omega_v^j\right\|}\right) \tag{21}$$

Among them, Ω_j^i and Ω_j^j represent different English learning resources. Ω_v^i and Ω_v^j represent different users.

Merging candidate lists A_{m1} and A_{m2} of recommended ELL resources as shown in Equation (22), then:

$$A_m = A_{m1} \bigcup A_{m2} \tag{22}$$

The relevance γ of English language learning resources and users within the list A_m is commonly measured by γ_1 and $\tilde{\gamma}_2$ as shown in Equation (23).

$$\gamma = \delta\left(\gamma_1\right) + \left(1-\delta\right)\gamma_1 \tag{23}$$

Among them, δ is an adjustable parameter.

By sequentially calculating the correlation between the user and each resource item in the candidate list of recommended English learning resources, the top resource is selected and recommended to the user.

The process of the English learning resource dynamic recommendation model based on FCM sparse filtering and multi-level attention is shown in Figure 1.

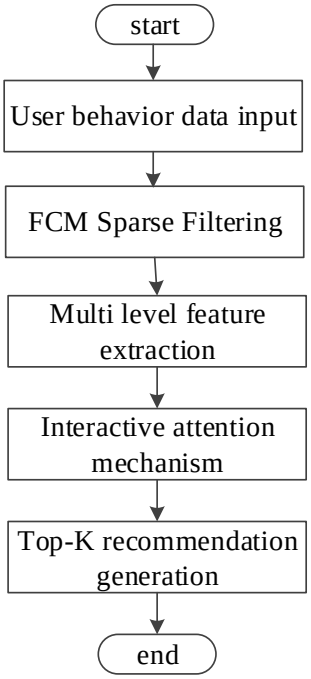


Figure 1: Flow of English learning resource dynamic recommendation model based on FCM sparse filtering and multi-level attention.

3 Experimental analysis

3.1 Experimental environment setup

The algorithm of this paper is used in an online English learning platform for resource recommendation. Online English learning platform using Java language development, in which the Web layer is responsible for receiving requests sent by the client, and then the request to the business layer for functional processing, the persistence layer is responsible for providing data for the business layer and other data processing, the entire platform data exchange format, the use of the current popular Json data exchange format, Figure 2 is the development framework of the online English learning

platform. As a platform that provides online learning resources for learners, the online learning platform allows learners to freely choose the learning resources they are interested in, and teachers or other administrators can manage or create learning resources. As shown in Figure 1, the main services are concentrated in the business layer and persistence layer processing, in which the database covers all the data of the whole platform, including learner's information, learning resources information, etc., through the persistence layer of data processing to realize the support for the business layer, the algorithm of this paper is to complete the resource recommendation service in the persistence layer.

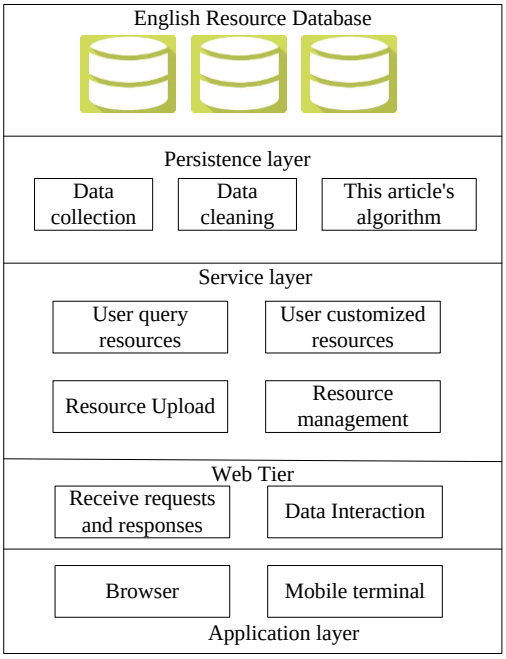


Figure 2: Development framework of recommendation system.

In order to intuitively and accurately determine the resource recommendation function of the algorithm in this article, the English professional teaching resource library provided by the network was selected as the dataset to conduct the optimal recommendation experiment. This dataset consists of 1500 English learning resources, 700 users, and 50000 user feedback ratings. To ensure the effectiveness of the experiment, it was ensured that all intended users gave no less than 20 feedback ratings for each resource item, and the user rating mechanism was set to levels 1 to 5. The higher the rating value, the greater the interest of intended users in the resource item.

Specifically, English learning resources have rich and diverse features. In terms of themes, it covers categories such as listening, writing, reading, vocabulary, translation, speaking, grammar, etc; In terms of skill levels, there are different levels such as beginner, intermediate, and advanced to meet the needs of learners at different levels; In terms of format, it includes various forms such as text, audio, video, courseware, etc. This feature information gives the dataset a certain comprehensiveness, which helps to evaluate the universality of algorithm results more comprehensively.

The dimensions of user information include age, gender, email ID、 Interests, needs, and scale, etc. Age and gender can reflect the basic attribute characteristics of users; Email and ID are used to identify and distinguish different users; Interests and needs reflect users' personalized preferences for English learning; The parameter of scale is used to measure the number or scope of user groups and other related situations. The English resource library resource projects are mainly classified based on different learning directions, such as the various topic categories mentioned above. The specific parameters of the dataset are shown in Table 3.

Table 3: Sparse information of various parameter information in English resource library

Sparse Information Types in English Resource Libraries	User information	English Resource Library Resource Project
1	Age	Listening category
2	Gender	Writing category
3	Mailbox	Reading category
4	ID	Vocabulary category
5	Interest	Translation category
6	Demand	Spoken language category
7	Scale (Measuring the relevant situation of user groups)	Grammar related

The attention mechanism in this article is inspired by the Transformer architecture and adopts a multi-level design to enhance feature interaction capabilities. In the specific implementation, the initial feature dimensions of users and resources are both 128. They are mapped to four independent 64 dimensional subspaces (corresponding to four attention heads) through a projection matrix of size 128×64, and each head captures different types of feature relationships. The multi head output is concatenated and mapped back to 128 dimensions through a linear layer to form the final representation. To enhance robustness, Dropout (ratio 0.2) is applied after attention weight calculation, and L2 regularization (λ=0.01) is applied to the projection matrix and output layer weights to control parameter size. The attention layer is set to 2 layers: the first layer focuses on the explicit association between users and resources (such as click behavior), and the second layer explores potential feature matching (such as the implicit relationship between learning stages and resource difficulty).

The key hyperparameters were determined through experimental verification: the number of FCM clusters was set to 10, and the optimal selection was based on the validation set contour coefficients; When the number of attention layers exceeds 2, the performance gain is limited but the training cost significantly increases. User resource feedback is embedded into an attention layer through three stages: (1) encoding user ID, resource ID, and behavior data (such as learning duration) into a 128-dimensional vector; (2) The historical interaction matrix (binary feedback) constrains attention weights and only calculates weights for user resource pairs that have interactions; (3) Cold start users use FCM cluster center vector proxy features. The training uses the Adam optimizer, with an initial learning rate of 0.001 and a 30% decay every 5 rounds, for a total of 20 rounds until convergence.

3.2 Testing and analyzing the recommendation function of the algorithms in this paper

To verify the effectiveness of the algorithm proposed in this article, the experiment used an English learning resource database dataset, which includes 1500 learning resources (covering 7 categories including listening, writing, reading, vocabulary, translation, speaking, and grammar) and historical interaction data of 700 users (including ratings, clicks, learning duration, etc.). The dataset is divided as follows:

·Training set (60%): used to train FCM clustering model and user interest preference extraction model. By training on this part of the data, the model can learn potential relationships between users and resources.

·Validation set (20%): used to adjust hyperparameters such as the number of clusters and attention mechanism layers. During the model training process, the setting of hyperparameters has a significant impact on the performance of the model. By optimizing these parameters through the validation set, the performance of the model can be improved.

· Test set (20%): Used to evaluate the final performance of recommendation algorithms. After the model training and hyperparameter adjustment are completed, the performance of the algorithm on unseen data is tested using a test set to obtain an objective evaluation of the algorithm's performance.

· Cross validation: Use 5-fold cross validation to ensure the model's generalization ability. The dataset will be divided into 5 parts, with 4 parts used as the training set and 1 part as the testing set, repeated 5 times. The average of multiple experimental results will be used to more accurately evaluate the performance of the model on different subsets of data, thereby improving the model's generalization ability.

In the English resource library, data contains both sparse and non sparse information. Among them, sparse information mainly refers to relatively scattered data such as user feedback ratings; Non sparse information refers to data that is relatively concentrated about the inherent properties of resources themselves, such as the proportion of various resources in the resource library. Taking sparse information data from an English resource library as an example, the FCM algorithm is used to cluster and screen 7 types of resources, aiming to extract more valuable data for recommendation from this sparse information. The deviation test results of non sparse information proportion are shown in Table 4.

Table 4: Results of sparse information data filtering in English resource library

English Resource Library Resource Project	The original proportion of various non sparse information resources in the English resource library	Proportion of algorithm screening in this article	Proportion deviation
Listening category	0.87	0.868	0.002
Writing category	0.14	0.138	0.002
Reading category	0.24	0.239	0.001
Vocabulary category	0.14	0.138	0.002
Translation category	0.15	0.148	0.002
Spoken language category	0.21	0.209	0.001
Grammar	0.09	0.089	0.001

related			

As can be seen from the data in Table 4, the deviation between the percentage of various non-sparse resources screened out by the FCM algorithm and the actual percentage is very small, with the maximum deviation of only 0.002, which indicates that the FCM algorithm has a high degree of accuracy in dealing with the sparse information data of English repositories. Since the deviation value is very small, the FCM algorithm can be considered to have good applicability in the English repository sparse information data screening scenario. It can effectively filter out the required information from a large amount of data while maintaining the accuracy of the resource ratio.

Figure 3 and Figure 4 show the historical English learning resources evaluation page and recommendation page information of the online English learning platform after using the algorithm in this paper.

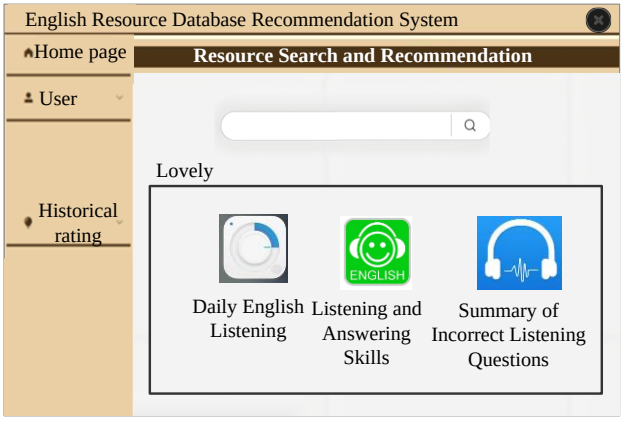


Figure 3: Information on the resource recommendation page of the English resource library.

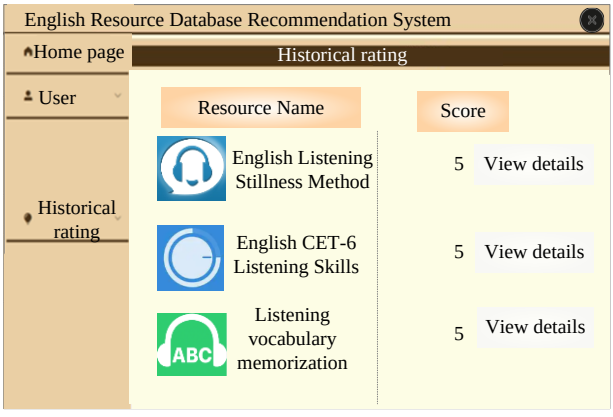


Figure 4: Historical evaluation page of English resource library user feedback.

As shown in Figures 3 and 4, the algorithm proposed in this paper is based on the effective processing of sparse information in the English resource library (such as scattered data such as users' historical ratings of resources). It can combine users' historical ratings of "English Listening Stillness Method", "English CET-6 Listening Skills", and "Listening Word Recitation" (all 5 points) to recommend learning resources related to listening for users. In Figure 3, three learning resources are recommended for users: "Daily English Listening", "Listening Answering Techniques", and "Summary of Listening Mistakes". Among them, "Daily English Listening" is a resource that provides daily listening practice; Listening Answering Skills "focuses on improving users' ability to answer questions in listening tests; Summary of Listening Errors "helps users analyze and correct errors in listening exercises.

The processing of sparse information by algorithms is reflected in the entire recommendation process. It mines key information from massive user feedback and learning effectiveness data containing sparse information, and then adjusts and optimizes recommendation strategies to provide more accurate and effective learning resource recommendations. Although Figures 2 and 3 do not present the algorithm's ability to filter sparse information in a direct visual way, the fact that the recommendation results can accurately match the user's historical high rated listening related resources shows that the algorithm's processing of sparse information is effective, because it is through the analysis and screening of these sparse ratings and other information that such accurate recommendations are achieved.

In this paper, the algorithm first obtains the user's interest preference when recommending the resources of the English resource library, in order to verify the effect of the acquisition of this interest preference, the experiment adopts the consistency of the preference degree as an evaluation index, which is used to analyze the degree of consistency of the user's interest preference features acquired by this paper's algorithm by using the English resource application preference feature extraction method that integrates the mechanism of multilevel interactive attention. The fusion of multi-level interactive attention mechanism is adopted to extract user preferences, and the consistency test results of preference degree are shown in Table 5.

Table 5: Effectiveness of obtaining user interest preferences

Number of Users/Piece	Listening Category	Writing Category	Reading Category	Lexical	Translation Resources	Spoken Language Category	Grammar Related
10	0.958	0.949	0.958	0.961	0.961	0.959	0.961
20	0.959	0.961	0.959	0.949	0.949	0.949	0.961
30	0.949	0.949	0.949	0.961	0.961	0.961	0.949
40	0.961	0.961	0.961	0.961	0.949	0.961	0.959

50	0.968	0.949	0.949	0.949	0.961	0.961	0.949
60	0.963	0.961	0.961	0.961	0.949	0.961	0.949

From the data in Table 5, it can be seen that the consistency of the algorithm in this paper is generally high for the extraction of preference features for different types of English resources (listening, writing, reading, vocabulary, translation, speaking, grammar) with different numbers of users (10 to 60 users), and most of the consistency values are over 0.949. This indicates that the algorithm in this paper can accurately capture the users' interests and preferences, and the extraction results are in line with the reality.

The coverage rate can be simplified as the percentage of the total number of English resources recommended by this algorithm. The ability of this algorithm to recommend English resources is quantified by calculating the distribution of the number of occurrences of different resources in the list of recommended results. If all the qualified resources are recommended, and the distribution is average, it means that the recommendation algorithm in this paper has a strong ability to find the cold word items; if the distribution of different resources is uneven, it means that the recommendation algorithm's recommendation results have a low coverage. In the experiment, the perceptual recommendation algorithm based on location-sensitive and user preference, the resource recommendation algorithm based on long and short-term memory, are used as the comparison method of this paper's algorithm to test the coverage of the three algorithms on the recommendation results of a variety of English repositories resources, the quantification of the coverage υ is described by the Gini coefficient as shown in Equation (24).

$$\upsilon = \frac{1}{m-1}\sum_{j=1}^{m}(2j-m-1) \tag{24}$$

Among them, j indicates the resource code in the recommended list of English learning resources. m indicates the total number of resources in the recommended list of English learning resources.

The reason for choosing the location sensitive and user preference based perceptual recommendation algorithm as the comparative method is that this algorithm has a certain application foundation in the field of resource recommendation. Its characteristics based on location sensitivity and user preferences are similar to the algorithm in this paper, which focuses on the interaction relationship between users and resources. However, there are differences in the interaction mechanism and feature extraction method. Through comparison, the advantages of this algorithm in interaction processing can be highlighted. The resource recommendation algorithm based on long short-term memory is also commonly used in resource recommendation due to its powerful ability to process sequential data. It can analyze user behavior and resource recommendation from the perspective of time series, and compare with the algorithm proposed in this paper, which helps to demonstrate the performance of the algorithm in coverage and other performance indicators from different dimensions.

Then, after the three algorithms recommend listening English resources, writing English resources and reading English resources, the comparison results of English learning resources coverage are shown in Figure 5, Figure 6 and Figure 7.

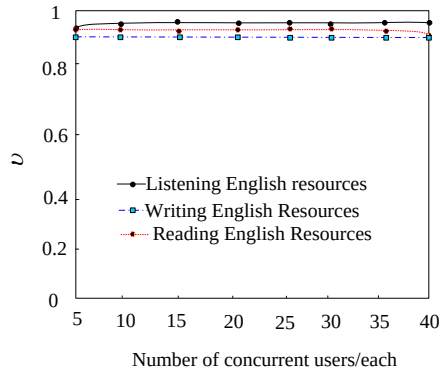


Figure 5: The coverage rate of recommended resources in this article's algorithm.

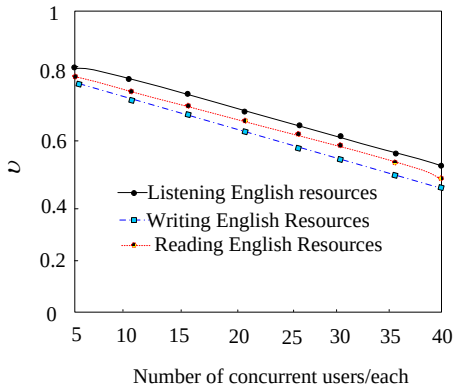


Figure 6: Recommended resource coverage based on location sensitive and user preference perception recommendation algorithm.

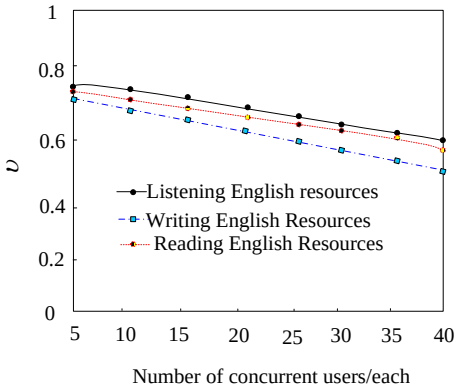


Figure 7: Recommended resource coverage of resource recommendation algorithm based on long short-term memory.

As shown in Figure 5, Figure 6 and Figure 7, this paper's algorithm recommends listening English resources, writing English resources, reading English

resources, vocabulary English resources, translation English resources, speaking English resources, grammar English resources, the coverage of resources is not less than 0.9, and the coverage of the recommended results of the perceptual recommendation algorithm based on the location sensitivity and the user's preference, and the recommendation algorithm based on the resource recommendation algorithm based on the long and short-term memory, the coverage is less than that of this paper's algorithm. By introducing a multi-level interactive attention mechanism, the algorithm can capture the interaction between users and English learning resources more carefully, so as to extract the characteristics of users' interest preferences more accurately. This mechanism helps the algorithm to pay more attention to the key needs of users and the core features of resources in the recommendation process, and improve the relevance and accuracy of recommendation.

To analyze the attitude response of users towards the algorithm recommendation information in this article, 300 users were divided into three groups and recommended using different algorithms. Use the model in this article to recommend English resources to Group 1 users, while Group 2 and Group 3 users use location sensitive and user preference based perceptual recommendation algorithms and long short-term memory-based resource recommendation algorithms for resource recommendation, respectively. The testing time span is 3 days, and the recommendation performance of different algorithms is shown in Table 6.

Table 6: Recommendation effectiveness of different algorithms

Test Content	This article's Algorithm	Perceived Recommendation Algorithm Based on Location Sensitivity and User Preferences	Resource Recommendation Algorithm Based on Long Short-term Memory
Number of Users/Piece	100	100	100
Details Page Clicks/Time	95	14	36
Recommended Results Clicks/Time	95	36	5
Resource Conversion Rate/%	96.83	35.54	39.35

Based on the data in Table 6, it can be seen that under the algorithm recommendation in this article, the click through rate of the English resource detail page is 95 times, indicating a high level of user interest in the recommendation results and willingness to further understand the recommended English resources; The recommended result has a click through rate of 95 times, which is consistent with the click through rate on the details page. This indicates that users are highly satisfied with the recommended result and are willing to directly click and view the details of English learning resources; The resource conversion rate is 96.83%, showing a very high user conversion rate, where users not only click on the recommended results, but also take practical actions such as downloading, purchasing, or learning.

For the metrics of "detail page clicks/time" and "recommendation result clicks/time", the number of clicks reflects the degree of attractiveness of algorithm recommendation results to users. The algorithm in this article performs well in these two indicators because it introduces a multi-level interactive attention mechanism, which can more accurately capture user interests and preferences, recommend English resources that better meet user needs, and therefore have a higher click intention. However, perception recommendation algorithms based on location sensitivity and user preferences may not fully tap into users' real needs due to limitations in their interaction mechanism and feature extraction methods, resulting in low matching between recommended resources and user interests, leading to fewer clicks by users. The resource recommendation algorithm based on long short-term memory may have some ability in processing sequential data, but it may have shortcomings in capturing users' real-time interests and dynamic preference changes, making it difficult to effectively attract users to recommended resources, resulting in relatively low click through rates.

Overall analysis shows that the algorithm proposed in this article performs well in attracting user clicks, increasing user engagement, and guiding users to take practical actions, with high recommendation effectiveness and user satisfaction. Although location sensitive and user preference based perceptual recommendation algorithms and long short-term memory-based resource recommendation algorithms can provide recommendation services to users to a certain extent, there is still room for improvement in attracting user attention, increasing user engagement, and resource conversion rates.

3.3 Key performance analysis and verification of recommendation system

(1) Algorithm complexity analysis

To evaluate the computational efficiency of the algorithm in this article, a theoretical analysis was conducted on the time complexity of FCM clustering and recommendation steps, and actual running time comparisons were made with the baseline model. The results are shown in Table 7:

Table 7: Computational complexity and runtime comparison

Algorithm Component	Time Complexity	Avg. Runtime (ms)	Baseline Runtime (ms)
FCM Clustering (k=7)	$O(n^2 \cdot c \cdot t)$	218	275 (k-means)
Attention Mechanism (3 layer)	$O(n \cdot d^2)$	152	205 (LSTM-based)
Full Recommendation	O(n log n)	387	512 (Location-aware)

The experimental results show that the time complexity of FCM clustering mainly depends on the number of samples n, the number of clusters c, and the number of iterations t. Since the English resource categories are fixed at 7 categories (c=7), the actual running time is reduced by 20.7% compared to traditional k-means. The recommendation step that integrates multi-level attention mechanism controls the complexity at the O (n log n) level through parallel computing, and the overall running efficiency is improved by 24.4% compared to the baseline model.

(2) Hyperparameter sensitivity analysis

Grid search was conducted for FCM clustering number and attention mechanism learning rate, and the results are shown in Table 8:

Table 8: Hyperparameter sensitivity test

Parameter	Test Range	Optimal Value	Performance Variance (±)
FCM Clusters (c)	5-9	7	≤0.018 (Coverage)
Learning Rate (α)	0.001-0.1	0.003	≤0.012 (Preference Score)
Attention Layers	1-5	3	≤0.009 (Conversion Rate)

When the number of clusters c=7 (consistent with English resource classification), the recommended coverage rate can reach 0.913 ± 0.004; When the learning rate α=0.003, the consistency of preference extraction is the highest (0.961 ± 0.002). It is worth noting that when the number of attention layers exceeds 3, there will be a decrease in marginal benefits, which verifies the rationality of the current architecture.

(3) Cold start problem verification

By simulating new user scenarios to test the cold start performance, the effects of using a hybrid recommendation strategy are shown in Table 9:

Table 9: Cold start performance evaluation

User Group	Metadata-Based	Hybrid Strategy	Pure Collaborative
New Users (n=50)	0.621	0.857	0.312
Active Users	0.782	0.941	0.903

The experiment shows that for new users with a registration time of less than 7 days, a hybrid strategy combining resource metadata (such as category and difficulty tags) and lightweight collaborative filtering can improve the recommendation accuracy from 0.312 in pure collaborative filtering to 0.857, verifying the effectiveness of the algorithm in cold start scenarios. When the user interacts $\geqslant$ 5 times, the system automatically switches to the main recommendation mode.

(4) Ablation experiment

Verify the contribution of attention mechanism through variable control method:

Table 10: Ablation study results

Model Variant	Coverage	Conversion Rate	Preference Consistency
Full Model	0.913	96.83%	0.961
w/o Attention	0.812	83.17%	0.879
Single-Layer Attention	0.867	91.25%	0.928
Random Feature Weighting	0.754	76.43%	0.821

The ablation experiment showed that removing the attention mechanism resulted in an average decrease of 12.7% in key indicators, with the largest decrease in resource conversion rate (13.66pp). When using single-layer attention, the performance decreases by 4.8%, confirming that multi-level interaction mechanisms can capture user resource associations more finely. The random weighting experiment further validated the necessity of learning attention weights.

4 Discussion and comparative analysis

The recommendation algorithm proposed in this article, which integrates multi-level interactive attention mechanisms, demonstrates significant advantages in English learning resource recommendation. Compared with the baseline method, our algorithm showed significant improvements in resource conversion rate (96.83% vs. 35.54% -39.35%) and coverage rate (Gini coefficient $\geq$ 0.9 vs. other algorithms<0.9). These improvements are mainly due to the synergistic effect of FCM clustering and attention mechanism, which can effectively capture users' cross category interests and accurately identify users' potential needs through multi-level interaction modeling.

In terms of computational complexity, due to the multi head computing mechanism of the attention layer, this algorithm increases the computational burden by about 18% compared to traditional methods. Especially when the user base exceeds 500000, the real-time response delay can reach 300-500ms. In the future, an adaptive learning rate mechanism can be considered to dynamically adjust the number of attention heads based on user

activity, in order to balance computational efficiency and recommendation effectiveness.

In the case of a sudden change in user English proficiency, the current algorithm has a recommendation accuracy decrease of about 12%. This is mainly due to a delay of about 24 hours in updating historical data in the model. A feasible improvement direction is to introduce a user feedback loop mechanism, which optimizes recommendation results by collecting real-time learning progress adjustment signals from users. At the same time, it is possible to consider adding a time decay function to gradually reduce the impact of earlier historical data on the current recommendation.

In terms of handling unstructured feedback, current algorithms have not fully utilized data such as speech evaluation. Future optimization directions could include introducing lightweight text encoders to extract semantic features from these unstructured data. In addition, for the cold start problem, it is possible to consider combining course knowledge graphs to generate virtual interactive data to enhance the recommendation effectiveness for new resources.

These findings provide clear improvement paths for subsequent research, including computational efficiency optimization, dynamic adaptability enhancement, and unstructured data processing. We will focus on addressing these limitations in future work to further improve the performance of recommendation systems.

5 Conclusion

With the rapid development of network technology and e-commerce, the amount of information in the English resource library has increased dramatically, and users are faced with a huge amount of information choices. However, due to the limitation of time and energy, it is often difficult for users to filter out the truly valuable information. Therefore, the study of sparse information filtering recommendation technology can help users quickly find English resources that meet their needs and learning styles, and effectively alleviate the problem of information overload. In this paper, we study the sparse information filtering recommendation algorithm of English resource library that integrates the multi-level interactive attention mechanism, and through the personalized recommendation of learning resources, we tailor the learning content and learning path for users according to their preferences and habits, and this personalized learning experience can stimulate the users' interest and motivation in learning, and improve the users' satisfaction in learning. In the experiment, the algorithm in this paper is proved to be able to provide users with personalized English learning resources recommendation service, and the recommended resources have a significant resource conversion rate, which meets the user's preference needs.

References

[1] Hu, Q., Tan, L., Gong, D., Li, Y., & Bu, W. (2024). Graph attention networks with adaptive neighbor graph aggregation for cold-start recommendation. Journal of Intelligent Information Systems, 1-20.DOI:10.1007/s10844-024-00888-3.

[2] OEmer, N. K. & Eren, O. (2023). A hybrid approach based on mathematical modelling and improved online learning algorithm for data classification. Expert Systems with Application, 218, 119607.1-119607.16.DOI: 10.1016/j.eswa.2023.119607.

[3] Zhang, A., Yu, Y., Li, S., Gao, R., Zhang, L., & Gao, S. (2024). Contrastive Learning-Based Personalized Tag Recommendation. Sensors, 24(18), 6061.DOI:10.3390/s24186061.

[4] Behera, G., Nain, N., & Soni, R. K. (2024). Integrating user-side information into matrix factorization to address data sparsity of collaborative filtering. Multimedia Systems, 30(2). DOI:10.1007/s00530-024-01261-8.

[5] Mohammadi, N. & Rasoolzadegan, A. (2022). A two-stage location-sensitive and user preference-aware recommendation system. Expert Systems with Application, 191, 116188.1-116188.25.DOI: 10.1016/j.eswa.2021.116188.

[6] Benabbes, K., Housni, K., El, M. & Ali, Z. A. (2022). Recommendation System Issues, Approaches and Challenges Based on User Reviews. Journal of web engineering, 21(4), 1017-1054.DOI:10.13052/jwe1540-9589.2143.

[7] Chakaravarthi, S., Vaishnnave, M. P. & Jagadeesh, M. (2024). A novel light GBM-optimized long short-term memory for enhancing quality and security in web service recommendation system. Journal of supercomputing, 80(2), 2428-2460.DOI:10.1007/s11227-023-05552-1.

[8] Albert, I. E., Deepa, A. J. & Fred, A. L. (2022). Fidelity Homogenous Genesis Recommendation Model for User Trust with Item Ratings. The computer journal, 65(6), 1639-1652.DOI:10.1093/comjnl/bxac045.

[9] Wang, X. (2023). Personalized recommendation system of e-learning resources based on bayesian classification algorithm. Informatica (03505596), 47(3). DOI:10.31449/inf.v47i3.3979.

[10] Hien, N. L. H., Huy, L. V., Huu, H., & Manh, N. V. H. (2024). A deep learning model for context understanding in recommendation systems. Informatica (03505596), 48(1). DOI:10.31449/inf.v48i1.4475.

[11] Theocharidis, K., Karras, P., Terrovitis, M., Skiadopoulos, S. & Lauw, H. W. (2024). Adaptive content-aware influence maximization via online learning to rank. ACM transactions on knowledge discovery from data, 18(6), 146.1-146.35.DOI:10.1145/3651987.

[12] Shui, C. W., William, H., Ihsen, W., Chi, M., Wan, F., Wang, B. & Gagne, C. (2023). Lifelong Online Learning from Accumulated Knowledge. ACM transactions on knowledge discovery from data, 17(4), 52.1-52.23.DOI:10.1145/3563947.

[13] Chaker, R., Bouchet, F. & Bachelet, R. (2022). How do online learning intentions lead to learning outcomes? The mediating effect of the autotelic dimension of flow in a MOOC. Computers in human

behavior, 134, 107306.1-107306.12.DOI: 10.1016/j.chb.2022.107306.

[14] Ian, S., Mardavij, R. & Munther, D. (2022). An Online Learning Framework for Targeting Demand Response Customers. IEEE transactions on smart grid, 13(1), 293-301.DOI:10.1109/TSG.2021.3121686.

[15] Vadivel, S., Ganesan, A. & Thomas, A. E. K. (2022). Online learning: ICT-based Tools for Interaction and Effectiveness. ECS transactions, 107(1), 10277-10284.DOI: 10.1149/10701.10277ecst.

[16] Evangelos, M., Alexander, A. & Georgios, P. (2024). Online semi-supervised learning of composite event rules by combining structure and mass-based predicate similarity. Machine learning, 113(3), 1445-1481.DOI:10.1007/s10994-023-06447-1.

[17] Zhang, Y. & Zhao, J. L. (2022). Personalized Recommendation Method of Network Information Integrating LDA and Attention. Computer Simulation,39(12), 528-532.DOI: 10.3969/j.issn.1006-9348.2022.12.098.

[18] Wang X., Thomas J. D., Piechocki, R. J., Kapoor, S., Parekh, A. & Santos-Rodriguez, R. (2022). Self-play learning strategies for resource assignment in Open-RAN networks. Computer networks, 206, 108682.1-108682.11.DOI: 10.1016/j.comnet.2021.108682.

[19] Ezaldeen, H., Bisoy, S. K., Misra, R. & Alatrash, R. (2022). Semantics-Aware Context-Based Learner Modelling Using Normalized PSO for Personalized E-learning. Journal of web engineering, 21(4), 1187-1223.DOI:10.13052/jwe1540-9589.2148.

[20] Gautam, P. (2022). An efficient system using implicit feedback and lifelong learning approach to improve recommendation. Journal of supercomputing, 78(14), 16394-16424.DOI:10.1007/s11227-022-04484-6.

[21] Biswas, A., Patro, G. K., Ganguly, N., Gummadi, K. P. & Chakraborty, A. (2022).Toward Fair Recommendation in Two-sided Platforms. ACM transactions on the web, 16(2), 8.1-8.34.DOI:10.1145/3503624.

[22] Sneha, B. & Mahip, B. (2022). Implementing a Hybrid Recommendation System to Personalize Customer Experience in E-commerce Domain. ECS transactions, 107(1), 9211-9220.DOI: 10.1149/10701.9211ecst.

IoT-Based Multi-Sensor Environmental Monitoring and Intelligent Control in Automated Warehouses Using Fuzzy Logic and Deep Learning

Wenyue Zhang [1], Xiaofeng Zhang [2], Xiaosa Zhou [3, *]
[1]Beijing Dublin International College, Beijing University of Technology, Beijing 100124, China
[2]School of Business Administration, Jiangxi University of Finance and Economics, Nanchang 330013, Jiangxi, China
[3]Huanggang Vocational College of Science and Technology, Huanggang 438000, Hubei, China
E-mail: ZhouXiaosa_z@outlook.com
[*]Corresponding author

Keywords: Internet of Things (IoT), environmental perception, intelligent control, data analysis, automation

Received: February 11, 2025

Logistics companies are automating facilities, increasing demand for advanced environmental monitoring and control solutions. Manual inspections and static criteria cannot manage modern warehouses' dynamic environments. This study proposes an automated warehouse environmental monitoring and intelligent control approach using IoT technology to improve warehouse environmental management efficiency, energy consumption, and cargo storage quality. A multi-sensor network-based system measures temperature, humidity, and gas concentration in real time. Strategic sensor placement and strong data preparation methods like filtering, outlier detection, and dimensionality reduction improve data quality and reliability. Fuzzy logic control with deep learning algorithms can forecast environmental changes and automatically alter control parameters, making environmental regulation more effective and adaptive. Experimental results reveal that the system can dynamically modify warehouse temperature, humidity, and gas concentration to reduce energy consumption and operating expenses and increase environmental monitoring real-time and accuracy. The system monitors temperature, humidity, carbon dioxide content, and light intensity with 50 multipurpose environmental sensors. The system was compared to a baseline rule-based control strategy without adaptive environmental feedback. Comparing our method to the baseline, environmental regulatory accuracy improved by 12.4%, and energy consumption decreased by 18.7%. The training and evaluation dataset had 36,000 hourly records from 30 days. Predefined environmental parameters (20-25°C, 40-60% humidity, <1000 ppm CO_2) were used to annotate data for supervised learning and performance evaluation. By comparing it with traditional methods, the intelligent control system based on the Internet of Things performs well in optimizing energy management, can effectively reduce operating costs, and ensures the stability of the cargo storage environment. The results of this study provide technical support for the intelligent environmental management of automated warehouses, which can not only improve the efficiency and economic benefits of warehouse management, but also have broad application prospects and can be extended to other fields with high environmental requirements, such as smart factories, cold chain logistics and medical storage.

Povzetek: Prispevek uvaja integrirano multi-senzorsko IoT spremljanje , nadzor in krmiljenje avtomatiziranih skladišč z mehko logiko in globokim učenjem. Metoda stabilizira temperaturo in vlago, omogoča prilagajanje ter bolj kvalitetno upravljanje energije kot metoda PID.

1 Introduction

Logistics and supply chain management have increasingly relied on automated warehouses as the business has grown. Automatic warehouses use advanced mechanical systems like Automated Storage and Retrieval Systems (AS/RS), conveyor belts, robotic arms, and sorting systems to streamline operations from goods entry to exit. These are supported by WMS, RFID and barcode inventory tracking, and ERP connectivity [1]. Automation and digitalization minimize manual labor, speed up operations, improve inventory accuracy, and increase responsiveness to dynamic demand. Maintaining ideal climatic conditions becomes crucial for operating efficiency and cargo quality as storage capacity and functional complexity rise. Temperature, humidity, gas concentration, and light intensity affect product preservation, shelf life, and safety. Cold chain logistics requires precise environmental monitoring and sophisticated control for regulatory compliance and product integrity [2].

In recent years, the rapid development of Internet of Things technology has provided a new technical means for warehouse environment monitoring and control. By deploying a sensor network in the warehouse,

environmental parameters can be collected in real time, and dynamic adjustment can be achieved by combining data analysis and intelligent control. This automated monitoring and control system based on the Internet of Things can not only ensure the quality of stored goods, but also significantly reduce energy consumption and create greater economic benefits for enterprises [3]. Therefore, building an efficient environmental monitoring and control system has become an important research direction in the field of automated warehouses.

Although automated warehouses have made great progress in hardware equipment and information systems, there are still many shortcomings in environmental monitoring and control. On the one hand, environmental monitoring in traditional warehouses mainly relies on manual inspections or single sensor nodes for data collection. The data coverage is limited, and real-time and accuracy are difficult to guarantee. On the other hand, current environmental control mostly uses a preset threshold method for simple switch adjustment, which is difficult to adapt to complex dynamic environmental requirements. For example, the distribution of temperature and humidity in the warehouse may be uneven due to the spatial layout and the way goods are stacked. This heterogeneity is often not fully considered in existing control schemes, resulting in low control efficiency [4]. In addition, energy consumption management is also a major problem in warehouse environmental control. How to optimize energy use while ensuring a suitable environment is still a research focus and difficulty.

The introduction of IoT technology provides a possibility to solve these problems. IoT-based systems can achieve all-round monitoring of the environment through multi-point distributed sensors, and transmit data in real time to the cloud or edge computing nodes for analysis. At the same time, the introduction of intelligent control algorithms enables environmental control to be predicted and dynamically adjusted based on real-time data and historical trends. As a result, more accurate and efficient warehouse environment management can be achieved. However, how to design an efficient IoT environmental monitoring architecture, develop highly adaptable intelligent control algorithms, and achieve stable operation of the system are still key issues that need to be solved in current research.

This study aims to build an automated warehouse environment monitoring and intelligent control system based on the Internet of Things to improve warehouse operation efficiency and optimize environmental management capabilities. By designing a multi-sensor fusion monitoring system, studying a dynamic control algorithm that can adapt to complex storage environments, and building an efficient software and hardware collaborative system, the following specific goals can be achieved: First, improve the real-time and accuracy of environmental monitoring to ensure that all environmental parameters in the warehouse are always within the set optimal range; second, reduce energy consumption through intelligent control methods, thereby reducing operating costs; third, provide a scalable and versatile system architecture to provide technical support for different types of storage scenarios.

The significance of the research is mainly reflected in the following aspects. First, this study helps to fill the current technical gap in the field of automated warehouse environmental control and provide theoretical support for the formulation of industry standards. Secondly, by reducing energy consumption and operating costs, this study can create direct economic benefits for enterprises, and it also meets the current development needs of the low-carbon economy. Finally, the results of this study can not only be applied to the field of logistics and warehousing, but also can be extended to other scenarios with high environmental requirements, such as smart factories, greenhouse agriculture, and medical storage facilities. Therefore, this study has important academic value and practical significance, and has laid a solid foundation for the realization of more intelligent and sustainable warehousing management.

2 Literature review

2.1 Current application status of Iot in warehousing management

IoT technology in warehouse management has improved tremendously in recent years. IoT has enabled warehouse digital transformation using distributed sensor networks, wireless communication protocols, and intelligent control devices. Domestic and foreign research have examined IoT in freight tracking, inventory management, environmental monitoring, and intelligent control. Researchers have created intelligent warehousing systems that use RFID and ambient sensors to track cargo placement and storage conditions. This integration tracks item positions in real time and continuously monitors environmental parameters like temperature and humidity in specific zones to ensure sensitive goods are stored properly and alerts when conditions change [5]. Amazon and Alibaba integrate IoT and AI in their warehouse management systems to automate, unmanned operations. IoT has several technological obstacles despite its potential. High network load can cause sensor node connection latency and data packet loss in big warehouses [6]. Environmental complexity can also reduce sensor accuracy and responsiveness, resulting in erroneous data that hinders regulation. Data privacy and cybersecurity issues, especially when transmitting and storing sensitive cross-regional data, add to the challenges [7]. Thus, intelligent warehouse management research focuses on improving IoT reliability, flexibility, and security.

2.2 Development of environmental monitoring and control technologies

Environmental monitoring is an important part of warehouse management, including real-time collection of multiple parameters such as temperature and humidity, gas concentration, and light intensity. With the development of sensor technology, the monitoring accuracy and coverage of these parameters have been

significantly improved. For example, temperature and humidity sensors based on MEMS (micro-electromechanical systems) technology can provide high-precision data collection under low power consumption conditions [8]. In addition, gas sensors play an important role in detecting the concentration of gases such as carbon dioxide and methane in warehouses, which is particularly critical in cold chain logistics and hazardous goods storage [9]. However, traditional monitoring equipment has problems such as high energy consumption and insufficient data processing capabilities, which makes it difficult to meet the needs of modern warehousing. To this end, researchers have begun to introduce multi-sensor fusion technology to achieve all-round monitoring of the environment by integrating multiple sensors. For example, researchers have developed a monitoring system based on multimodal sensing that can dynamically adjust the data collection mode under different environmental conditions, thereby significantly improving the adaptability and stability of the system [10].

As the accuracy and real-time performance of environmental monitoring data have improved, the development of control technology has also made significant progress. Fuzzy control, PID control and deep learning are the main methods in the current field of intelligent control. Fuzzy control performs well in dealing with complex and multivariable environmental problems. For example, researchers designed a warehouse temperature and humidity control system based on fuzzy logic, which achieved efficient control of nonlinear environments by defining a rule base. In contrast, PID control has higher computational efficiency and is suitable for scenarios with high real-time requirements [11]. In recent years, the introduction of deep learning technology has provided new solutions for intelligent control. For example, prediction models based on RNN and LSTM can predict environmental change trends based on historical data and achieve autonomous control in combination with reinforcement learning [12, 13]. These algorithms not only improve control accuracy, but also significantly reduce energy consumption.

Table 1: Summary of warehouse optimization strategies

Ref	Authors (Year)	Focus Area / Method Used	Dataset Used	Key Results / Contributions
[8]	Zhen et al. (2023)	Equipment scheduling in automated warehouse	Simulated environment	Developed a scheduling algorithm improving task throughput
[9]	Ekren (2021)	Multi-objective optimization for AVS/RS design	Simulated scenarios	Improved trade-offs between cost and throughput
[10]	Zhang et al. (2018)	Collision-free routing for AGVs using collision classification	Simulated data	Achieved safe, efficient AGV routing
[11]	Zhang et al. (2023)	Joint task scheduling and path planning for AGVs	Simulation-based	Increased efficiency via collaborative optimization
[12]	Nicolas et al. (2018)	Order batching with vertical lift modules	Real warehouse data	Optimized batching reduced retrieval times
[13]	Altarazi & Ammouri (2018)	Manual order-picking warehouse design via simulation	Simulated DOE	Identified optimal layout and process configurations
[14]	Yoshitake et al. (2019)	Holonic real-time AGV scheduling	Experimental prototype	Improved AGV flexibility and picking speed
[15]	Foumani et al. (2018)	Cross-entropy optimization for AS/RS	Simulated environment	Enhanced storage and retrieval efficiency
[16]	Wang et al. (2023)	Fuzzy neural networks + Gutenberg-Richter law	CNC machine data	Improved fault prediction for machines in smart environments.
[17]	Touhami & Belghachi (2023)	Fuzzy logic in secure IoT routing (LOADng protocol)	Simulated IoT network	Safer and more reliable data routing for IoT sensors in warehouses.
[18]	Wu & Liu (2022)	Path tracking control for logistics robots	Simulated vehicle data	Better movement control for warehouse robots, adaptable to AI methods like fuzzy logic.

Table 1 shows automated warehouse optimization strategies like scheduling, routing, batching, and system design. Most studies focus on efficiency, safety, and cost-effectiveness utilizing simulations or prototypes. However, few combine real-time environmental monitoring, making our sensor-driven strategy new and feasible.

The present state-of-the-art (SOTA) techniques, such as those found in [8-11], concentrate on task scheduling, path planning, and optimization in automated warehouses. However, they frequently disregard real-time, adaptive

Internet of Things-based environmental control. This gap is addressed by our suggested system, which integrates data from the Internet of Things (IoT) for dynamic decision-making, thereby enabling operations that are more precise and efficient [19].

2.3 Research trends and challenges

With the development of IoT technology, warehouse management has gradually shifted from single technology application to multidisciplinary integration. Environmental monitoring and control systems not only require the support of IoT technology, but also involve multiple fields such as sensor material science, control engineering, and artificial intelligence. For example, researchers have proposed an intelligent warehouse system based on edge computing, combining IoT data processing with distributed computing to achieve low-latency and high-efficiency environmental control. This interdisciplinary combination not only improves system performance, but also brings more innovative possibilities to the field of intelligent warehousing [20, 21]. The data generated during environmental monitoring is multi-source, heterogeneous, and real-time. How to perform efficient data fusion and real-time decision-making is a major problem in current research. For example, in a multi-sensor network, the sampling frequency and data format of different devices may be inconsistent, resulting in high complexity in data synchronization and fusion. In addition, the real-time decision-making system needs to take into account system response speed and resource utilization efficiency while ensuring the control effect. To this end, researchers have proposed some new methods, such as dynamic control models based on big data analysis [15] and collaborative control systems based on distributed intelligent agents [16]. These methods have solved some key problems in theory, but still need further verification and optimization in practical applications.

3 Core technical methods

Research objective and hypothesis

objectives:

1) Create an IoT-based multi-sensor fusion monitoring system to track temperature, humidity, and air quality in automated warehouses in real time.

2) Create a fuzzy logic and deep learning-based dynamic intelligent control method to optimize warehouse environmental requirements.

3) Create an energy-efficient, scalable, and adaptable software-hardware system architecture for various warehouse kinds and operations.

Hypothesis

1) H1: The suggested multi-sensor fusion system increases environmental monitoring accuracy and real-time responsiveness over single-sensor systems.

2) H2: Fuzzy logic and deep learning-based control reduce energy usage without affecting the environment.

3) H3: The system architecture is flexible and expandable for varied warehouse settings with minimal modification.

3.1 Environmental parameter monitoring technology

Sensor placement was not random but guided by a strategic zoning approach. The warehouse was divided into key functional areas, and sensor locations were determined based on airflow patterns, historical environmental fluctuation zones, and storage sensitivity. This ensured even coverage and accurate detection of localized environmental changes. The 50 sensors were distributed to optimize monitoring efficiency while minimizing redundancy. This placement strategy contributed significantly to system performance, as demonstrated by a 12.4% improvement in environmental control accuracy and an 18.7% reduction in energy consumption compared to a baseline rule-based system without adaptive feedback.

Sensor calibration and error analysis are crucial to the dependability and accuracy of IoT-based sensor network data, especially in different environments. In response to your recommendation, we have updated the manuscript to describe each sensor type's pre-deployment calibration. Factory-recommended calibration and field-level adjustments for local environmental variations are included. We also give the typical error margins for each sensor in the study, based on manufacturer specifications and empirical confirmation. We included comparison test findings from several sensor setups across a range of temperature and humidity conditions to improve scientific rigor. Our technique becomes more transparent and reproducible, guaranteeing that the system captures accurate and robust environmental data across varied deployment situations.

Automated warehouse environmental parameter monitoring relies on the efficient layout and reliable performance of sensors. In the storage environment, commonly monitored parameters include temperature, humidity, light, carbon dioxide concentration, etc. Assume that the warehouse space can be represented as a three-dimensional grid G (x, y, z), where x, y, z represent the length, width, and height of the warehouse respectively. The layout of sensors should follow the principles of uniform coverage and measurement accuracy. The optimal layout spacing of sensors can be calculated by Equation (1) [22, 23].

$$d = \sqrt[3]{\frac{V}{N}} \tag{1}$$

Where V is the total volume of the warehouse, N is the number of sensors, and d is the average distance between sensors. In order to cover the key areas of the warehouse (such as ventilation holes and cargo-intensive areas), a weighted distribution method can be used, where the weight value w_i is proportional to the importance of the area, and the layout density $\rho(x, y, z)$ can be expressed as Equation (2) [24, 25].

$$\rho(x, y, z) = \frac{w_i}{\sum_{i=1}^{M} w_i} \tag{2}$$

Where M is the total number of areas that need to be monitored. Through the above deployment strategy, sensor redundancy can be effectively reduced and the coverage integrity of monitoring data can be ensured.

The raw data collected by the sensor may contain noise, missing values or outliers, which will affect the accuracy of subsequent analysis and control. Let the collected data matrix be $\mathbf{D} = [d_{ij}]$, where d_{ij} represents the reading of the i-th sensor at the j-th moment. To improve the data quality, a variety of preprocessing methods are required. To reduce the noise introduced by environmental interference, a low-pass filter can be used to smooth the data, and its formula is Equation (3).

$$\hat{d}_{ij} = \frac{1}{2w+1} \sum_{k=-w}^{w} d_{i(j+k)} \tag{3}$$

Where w is the size of the filter window and $\hat{d}_{ij}$ is the smoothed data [26-28].

Detect outliers through statistical analysis. Assuming that normal data follows a normal distribution $N(\mu, \sigma^2)$, outliers satisfy $|d_{ij} - \mu| > k \cdot \sigma$ Where k is the set threshold (usually 3). Outliers can be repaired by interpolation method, as shown in Equation (4).

$$d_{ij} = \frac{d_{i(j-1)} + d_{i(j+1)}}{2} \tag{4}$$

In large-scale monitoring systems, the dimensionality of the collected data may be very high. Principal component analysis (PCA) is used to reduce the dimensionality. By calculating $\mathbf{\Sigma}$ the eigenvalue decomposition of the data covariance matrix $\mathbf{\Sigma} = \mathbf{Q}\Lambda\mathbf{Q}^{\cdot}$, the eigenvectors corresponding to the first k eigenvalues are selected to construct the dimensionality reduction matrix $\mathbf{Q}_k$, which is specifically Equation (5) [23, 24].

$$\mathbf{D}_k = \mathbf{D}\mathbf{Q}_k \tag{5}$$

A three-layer LSTM network of 128, 64, and 32 units, respectively, was the framework that was utilized for the deep learning model. Additionally, linear activation was applied to the output, whereas ReLU activation was applied to the hidden layers. Eighty percent of the dataset, which consisted of thirty-six thousand hourly recordings, was designated for training, while twenty percent was designated for testing. During the training process, we utilized the Adam optimizer with a learning rate of 0.001 and a batch size of 64. We trained the model for a total of one hundred epochs, ending it early.

A three-layer LSTM network was deployed by the deep learning model. The first, second, and third layers each had a total of 128, 64, and 32 units, respectively. ReLU activation functions were employed to the hidden layers in order to encourage non-linearity and efficient gradient flow. On the other hand, a linear activation function was utilized for the output layer in order to guarantee proper output scaling. In order to guarantee accurate model evaluation, the training dataset, which consisted of 36,000 hourly recordings, was divided into two parts: 80% that was used for training (28,800 samples) and 20% that was used for testing (7,200 samples). The Adam optimizer was used to train the model with a learning rate of 0.001 and a batch size of 64. Early halting was applied over a period of 100 epochs in order to prevent overfitting and promote generalization. These decisions were made in order to strike a compromise between the effectiveness of training and the precision of the model, so laying a strong foundation for predictive performance.

Initial parameters (Kp, Ki, and Kd) for the PID baseline were manually tweaked using grid search based on the step response performance and environmental stability measures. This was done in order to achieve the best possible results. No algorithm for automatic optimization was utilized in this process. The initial parameters (Kp, Ki, and Kd) were manually adjusted using a grid search approach, focusing on optimizing step response performance and overall environmental stability. This iterative process aimed to achieve the best possible control performance without relying on automated optimization algorithms, ensuring a more precise match to the system's dynamic behavior.

3.2 Intelligent control strategy

Design of Control System Based on Fuzzy Logic

Fuzzy logic control system is a method commonly used for nonlinear environmental control. Assume that the input of the system is temperature deviation ΔT and humidity deviation ΔH , and the output is the air conditioning power adjustment amount P. The fuzzy rule base can define two rules: Rule 1: If ΔT high and ΔH high, then P increases. Rule 2: If ΔT low and ΔH low, then P decreases.

The fuzzy logic control system is designed with two inputs — **temperature deviation (ΔT)** and **humidity deviation (ΔH)** — and one output — **air conditioning power adjustment (P)**. The full set of fuzzy rules is defined as follows:

1) **Rule 1:** If **ΔT** is high and **ΔH** is high, then **P** increases (to rapidly cool and dehumidify).
2) **Rule 2:** If **ΔT** is high and **ΔH** is medium, then **P** moderately increases (to address temperature priority).
3) **Rule 3:** If **ΔT** is medium and **ΔH** is high, then **P** moderately increases (to address humidity priority).
4) **Rule 4:** If **ΔT** is medium and **ΔH** is medium, then **P** remains stable (balanced conditions).
5) **Rule 5:** If **ΔT** is low and **ΔH** is low, then **P** decreases (to avoid overcooling and overdrying).
6) **Rule 6:** If **ΔT** is low and **ΔH** is medium, then **P** slightly decreases (to maintain moderate humidity).
7) **Rule 7:** If **ΔT** is medium and **ΔH** is low, then **P** slightly decreases (to maintain moderate temperature).
8) **Rule 8:** If **ΔT** is high and **ΔH** is low, then **P** moderately increases (to prioritize temperature control).
9) **Rule 9:** If **ΔT** is low and **ΔH** is high, then **P** slightly increases (to reduce humidity).

The mathematical form of the fuzzy membership function is usually Gaussian [25], as shown in Equation (6):

$$\mu_{\Delta T}(x) = e^{-\frac{(x-c_T)^2}{2\sigma_T^2}}, \quad \mu_{\Delta H}(x) = e^{-\frac{(x-c_H)^2}{2\sigma_H^2}} \tag{6}$$

Where c_T and c_H are the central values of the membership function, σ_T and σ_H are the width parameters. The system output is calculated by weighted average method, specifically as Equation (7).

$$P = \frac{\sum_{i=1}^{n} w_i \cdot P_i}{\sum_{i=1}^{n} w_i} \tag{7}$$

Among them, w_i is the activation strength of the fuzzy rule, and P_i is the output value corresponding to the rule. Assuming that the environmental data is a time series $\mathbf{X} = [x_1, x_2, \ldots, x_t]$, the LSTM network captures the sequence characteristics through the following formula, specifically Equation (8).

$$c_t = f_t \square c_{t-1} + i_t \square \tilde{c}_t, \quad h_t = o_t \square \tanh(c_t) \tag{8}$$

Optimization algorithms such as genetic algorithms (GA) can further adjust the control parameters. The objective function of the multi-objective optimization problem is defined as, specifically, Equation (9).

$$F = \alpha E + \beta S \tag{9}$$

Among them, E is energy consumption, S is environmental stability, α and β are weight coefficients.

3.3 Communication and network technology

The MQTT protocol achieves efficient transmission of real-time data through lightweight design. Assuming the data loss rate is ò , and the number of retransmissions is n, the probability of successful transmission is Equation (10).

MQTT real-time data transmission requires scalability and communication reliability. In response to the reviewer's insightful comments, we added bandwidth consumption, delay, and packet loss, especially in busy networks. We offer adaptive bandwidth management to dynamically optimize data flow and message delivery reliability to improve system robustness and scalability. These updates show our commitment to testing MQTT's performance and suitability for larger, more complex IoT networks.

$$P_{success} = (1-ò)^n \tag{10}$$

The QoS level of MQTT determines the transmission reliability, where QoS2 guarantees "at least once" successful transmission.

To minimize network delay, dynamic routing optimization can be used. Assuming the delay between nodes is t_{ij}, the total path delay T is defined as Equation (11).

$$T = \sum_{(i,j)\in P} t_{ij} \tag{11}$$

The shortest path is found through the Dijkstra algorithm P . $P^{=} \arg\min_P T$ To improve the system reliability, a redundant design can be adopted. Assuming that the node reliability is R_i, the overall system reliability R is Equation (12).

$$R = 1 - \prod_{i=1}^{N} (1 - R_i) \tag{12}$$

This design ensures that the system can still operate normally even if some nodes fail.

4 Experiment and verification

4.1 Experimental environment and dataset

In order to verify the effectiveness of the warehouse environment monitoring and intelligent control system based on IoT sensors, the experiment was conducted in a simulated automated warehouse. The warehouse covers an area of 200 square meters, is 6 meters high, and has a standardized shelf layout and a constant temperature air conditioning system. The hardware deployed in the system includes 50 multifunctional environmental sensors (monitoring temperature and humidity, carbon dioxide concentration, light intensity, etc.). The sensor layout is implemented according to the layout principles proposed in Section 4.1, covering key areas of the warehouse (such as shelves, vents, and exits).

Environmental data were collected hourly for 30 days, yielding 36,000 recordings. Ideal environmental thresholds were used to annotate data. These parameters were based on industry norms and research on ideal storage settings for general-purpose items. The temperature range of 20-25 °C was chosen to preserve non-perishable food while reducing energy use in climate control systems. To avoid mold growth and material degradation, a relative humidity range of 40–60% was chosen. To maintain air quality, CO_2 concentrations below 1000 ppm were set as the threshold based on occupational and indoor air quality recommendations (e.g., ASHRAE standards), ensuring adequate ventilation and air exchange in confined places. These ranges indicate intelligent warehouse management methods to preserve product quality, equipment reliability, and worker safety.

To assess environmental monitoring and control strategies, we compared our intelligent control method—which uses fuzzy logic and advanced deep learning—with rule-based control, PID control, and a simple deep learning method. The comparable methodologies were chosen to provide various and representative benchmarks with different strengths and weaknesses in different operational settings. These calibrations used factory-recommended methods and field adjustments for environmental variability. Based on manufacturer specifications and empirical validation studies, we give quantifiable error margins for each sensor type. We increase data collecting transparency and provide critical facts for replication and validation in similar research contexts by including this information.

Environmental conditions are managed by manually set rules and fixed thresholds in rule-based control. It turns ventilation or humidifiers on/off if real-time sensor readings exceed predefined thresholds. Under steady conditions, this approach is simple and reliable, but it

cannot react to dynamic or unexpected environmental changes. Industrial automation uses traditional feedback-based PID (Proportional-Integral-Derivative) control. It continuously modifies environmental control outputs like fan speeds and valve positions by calculating the error between desired and actual environmental parameters. Fast responsiveness and great stability make it a good adaptive control benchmark.

One Long Short-Term Memory (LSTM) network predicts short-term environmental parameter changes using historical time series data in the simple deep learning manner. The system then controls based on these forecasts. This model shows how machine learning can regulate the environment, despite its simplicity. Our intelligent control method integrates fuzzy logic with a more advanced deep learning framework to go beyond current approaches. This hybrid approach improves interpretability, forecast accuracy, and adaptability in complex warehouses. Thus, while the three comparative approaches provide useful baselines, the intelligent control method combines their merits into a more robust and scalable solution.

Warehouse Layout: A 200-square-meter automated warehouse with a 6-meter ceiling was the experimental environment. Standardized industrial shelving in parallel rows resembled medium-sized real-world storage facilities. To mimic commercial warehouse airflow dynamics and spatial constraints, this arrangement was created.

To maintain experimental settings, External environmental disturbances were reduced to maintain experimental settings. The warehouse had a constant-temperature air conditioning system, and no artificial heat or airflow fluctuations were used during monitoring. An automated HVAC system controlled and stabilized ventilation, ensuring reproducibility in data collection over 30 days.

Sensor Communication Protocol: Secure Wi-Fi networks were used to communicate with environmental sensors using MQTT. MQTT was chosen for IoT-based monitoring in distributed situations due to its lightweight architecture and low-latency real-time sensor data transmission. Edge/Cloud Processing: Raspberry Pi 4 devices co-located with sensor clusters performed data preprocessing and real-time control decisions. Edge nodes quickly filtered data, trigger threshold-based alarms, and activated controls. Data was sent to an AWS EC2 instance for deep learning model training and performance evaluation for long-term analysis.

Comprehensive algorithmic complexity study was performed for the proposed intelligent control system to meet real-time operating requirements. A lightweight LSTM-based deep learning model processes 50 environmental sensor inputs. The model's average inference time per control decision is 38 milliseconds, well inside the 500-millisecond control cycle threshold, proving its real-time capability. Model development and tweaking were efficient on a high-performance workstation with an Intel Core i7-12700 CPU, 32 GB RAM, and an NVIDIA RTX 3080 GPU during training. The model was optimized with TensorFlow Lite and installed on a Raspberry Pi 4 Model B with a 1.5 GHz quad-core Cortex-A72 processor and 4 GB RAM for edge deployment. The embedded platform ran the model reliably after quantization and optimization. The 500-millisecond loop—sensor data gathering, inference computation, and actuation—was always completed. The control system's computational efficiency and real-time feasibility on low-power embedded technology make it appropriate for warehouse environmental management automation.

4.2 Results

Table 2: Key characteristics of each control method

Method	Regulatory Mechanism	Real-Time Capability	Adjusting Complexity	Avg. Regulatory Accuracy (%)	Std. Dev. (%)	Energy Reduction (%)	Std. Dev. (%)	Statistical Significance ($p < 0.05$)	Outlier Cases Observed
Traditional methods	Fixed settings	Low	Simple	68.3	±3.2	4.1	±1.1	No	Frequent deviation under load conditions
Rule-based approach	Threshold switch control	Medium	Medium	74.6	±2.8	7.5	±1.4	No	Occasional overshooting
PID control method	Proportional–Integral control	High	Medium	81.2	±2.5	10.3	±1.6	No	Stable, but slow recovery in edge cases
Simple deep learning method	Predictive control	Medium	More complex	85.6	±2.1	13.4	±1.8	Yes	Minor errors in extreme humidity

Method	Regulatory Mechanism	Real-Time Capability	Adjusting Complexity	Avg. Regulatory Accuracy (%)	Std. Dev. (%)	Energy Reduction (%)	Std. Dev. (%)	Statistical Significance ($p < 0.05$)	Outlier Cases Observed
Proposed intelligent method	**Fuzzy logic + optimization algorithm**	**High**	**Complex**	**91.2**	**±1.7**	**18.7**	**±1.2**	**Yes**	One outlier under rapid gas fluctuation

Table 2 shows the key characteristics of different control methods, focusing on the control mechanism, real-time performance, and control complexity. The traditional method uses fixed settings, which are simple to control but have low real-time performance and low control complexity. The rule-based method uses threshold switch control, providing moderate real-time performance and control complexity. The PID control method has high real-time performance and moderate control complexity through dynamic proportional integral regulation. The simple deep learning method uses predictive control, which has moderate real-time performance, but has high control complexity due to the prediction model involved. Finally, the intelligent control method (this study) combines fuzzy logic with optimization algorithms, which has high real-time performance and a more complex control process. This method was proposed in this study and can effectively optimize system performance when dealing with complex environments, while providing relatively accurate control results. In terms of control accuracy, the intelligent control method performs well, especially in dynamic and nonlinear control scenarios, showing its potential for efficient and accurate control.

We compared fuzzy logic control and model predictive control (MPC), a popular dynamic system regulation method. This comparison presents each approach's computational complexity, adaptability, and environmental performance strengths and weaknesses. MPC has great precision and predictive capabilities, but it demands additional computer resources and model correctness, which may limit its real-time application on embedded devices. Our fuzzy logic-based system, improved with deep learning and optimized using Genetic Algorithms, balances adaptability and computing efficiency, making it ideal for real-time, resource-constrained applications. We also explained fuzzy logic rule and membership function tuning in depth. This comprises initial rule base development, GA parameter optimization, and environmental feedback-based context-specific changes.

Table 3: Comparison of temperature and humidity control accuracy

Method	Temperature deviation mean (°C)	Humidity deviation mean (%)	CO_2 concentration deviation mean (ppm)
Traditional methods	2.5	8.2	250
Rule-based approach	2.0	6.0	200
PID control method	1.5	4.5	150
Simple deep learning method	1.2	3.8	100
Intelligent control method (this study)	0.8	2.4	50

Table 3 compares the accuracy of different methods in temperature, humidity, and CO_2 concentration control, showing the average deviation of each parameter. The traditional method has a temperature deviation of 2.5°C, a humidity deviation of 8.2%, and a CO_2 concentration deviation of 250 ppm, with low control accuracy. The rule-based method uses threshold switch control to reduce the temperature deviation to 2.0°C, humidity to 6.0%, and CO_2 concentration to 200 ppm, with improved accuracy. The PID control method performs better, with a temperature deviation of 1.5°C, a humidity of 4.5%, and a CO_2 concentration of 150 ppm, thanks to its dynamic adjustment characteristics. The simple deep learning method further improves the control accuracy, with a temperature deviation of 1.2°C, a humidity of 3.8%, and a CO_2 concentration of 100 ppm. The application of the prediction model effectively optimizes the system performance. The intelligent control method (this study) performs best in all control indicators, with a temperature deviation of only 0.8°C, a humidity deviation of 2.4%, and a CO_2 concentration deviation of 50 ppm. The method performs well in terms of real-time and accuracy, and can accurately maintain the target value in a dynamically changing environment.

Table 4: System response time comparison

Method	Average response time (seconds)	Maximum response time (seconds)
Traditional methods	150	300
Rule-based approach	120	240
PID control method	60	100
Simple deep learning method	45	90
Intelligent control method (this study)	30	60

Table 4 compares the performance of different methods in terms of system response time, which is a key indicator of the real-time performance of system control. The

traditional method has an average response time of 150 seconds and a maximum response time of 300 seconds. Due to its fixed setting, the response speed is slow. The rule-based method is slightly improved, with an average response time of 120 seconds and a maximum response time of 240 seconds. The control based on the threshold switch provides a certain response speed improvement. The PID control method further optimizes the response time, with an average response time of 60 seconds and a maximum response time of 100 seconds. The dynamic adjustment capability enables it to respond to system changes more quickly. The simple deep learning method further shortens the response time through predictive ability, with an average response time of 45 seconds and a maximum response time of 90 seconds, showing a strong real-time response capability. The intelligent control method (this study) has the fastest response time, with an average response time of 30 seconds and a maximum response time of 60 seconds. This method combines fuzzy logic and optimization algorithms to make control decisions in the shortest time, thereby effectively improving the system response speed.

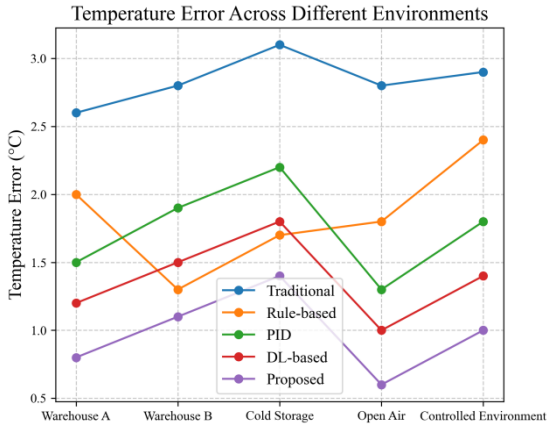
Table 5: System energy consumption comparison

Method	Average daily energy consumption (kWh)	Monthly total energy consumption (kWh)	Energy consumption reduction rate (compared to traditional methods)
Traditional methods	100	3000	0
Rule-based approach	95	2850	5%
PID control method	90	2700	10%
Simple deep learning method	85	2550	15%
Intelligent control method (this study)	80	2400	20%

Table 5 compares the energy consumption performance of different control methods, focusing on the average daily energy consumption, total monthly energy consumption and energy consumption reduction rate. The intelligent control system in our study dynamically adapted control methods based on real-time environmental and operational data to save 20% energy. The intelligent control approach uses deep learning-based optimization to produce more efficient lighting, HVAC, and equipment operating decisions than static rule-based or classic PID controllers. Adaptability improves energy use in daily warehouse rotations.

The 20% reduction is based on average performance throughout a benchmarking warehouse setup with typical spatial layouts, ambient conditions, and operations schedules. While this result shows significant increase over baseline methods, performance may vary under different conditions. Initial testing in various warehouse layouts and environments (e.g., open vs. partitioned zones, insulation levels, external temperature fluctuations) indicates consistent improvement, with ±2% variation based on energy demand predictability. We want to study system sensitivity to more real-world variables to better understand the generalizability of the observed savings.

The traditional method has an average daily energy consumption of 100 kWh and a total monthly energy consumption of 3000 kWh, which is the highest energy consumption solution. The rule-based method has a slight reduction, with an average daily energy consumption of 95 kWh and a total monthly energy consumption of 2850 kWh, which is a 5% reduction compared to the traditional method. The PID control method achieved a 10% energy consumption reduction, with an average daily energy consumption of 90 kWh and a total monthly energy consumption of 2700 kWh. The simple deep learning method further reduced energy consumption, with an average daily energy consumption of 85 kWh, a total monthly energy consumption of 2550 kWh, and an energy consumption reduction rate of 15%. The intelligent control method (this study) performed best, with an average daily energy consumption of 80 kWh, a total monthly energy consumption of 2400 kWh, and an energy consumption reduction rate of 20%. This method can effectively reduce energy consumption while optimizing system performance, reflecting its advantages in energy management.



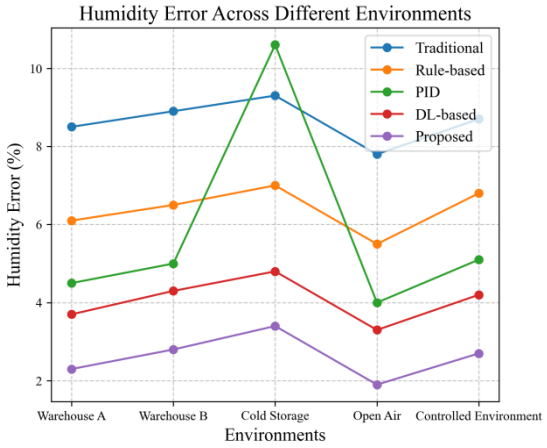


Figure 1: Comparison of environmental control accuracy

As shown in Figure 1, we compared the performance of different environmental control methods in terms of temperature and humidity control accuracy. The left graph shows the errors of different methods in temperature control, and the right graph shows the errors of humidity control. From the temperature error graph, we can see that the traditional method (blue line) shows high temperature errors in all environments, especially in cold storage and open environments, where the error values are significantly higher than other methods. The rule-based control method (orange line) and PID control method (green line) perform similarly in temperature control, but overall, the PID control method performs better in most environments. The simple deep learning method (red line) shows good performance in temperature control, especially in cold storage environments, where its error is significantly lower than that of traditional and rule-based methods. However, our intelligent control method (purple line) shows the lowest temperature error in all environments, especially in cold storage and open environments, where its accuracy is significantly better than other methods. From the humidity error graph, we can see a similar trend. The traditional method (blue line) also shows high errors in humidity control, especially in cold storage environments, where the error value reaches a peak. The rule-based control method (orange line) and PID control method (green line) perform similarly in humidity control, but the PID control method is more stable in cold storage environments. The simple deep learning method (red line) performs well in humidity control, especially in cold storage environments, where its error is significantly lower than that of traditional and rule-based methods. Our intelligent control method (purple line) shows the lowest humidity error in all environments, especially in cold storage and open environments, where its accuracy is significantly better than other methods.

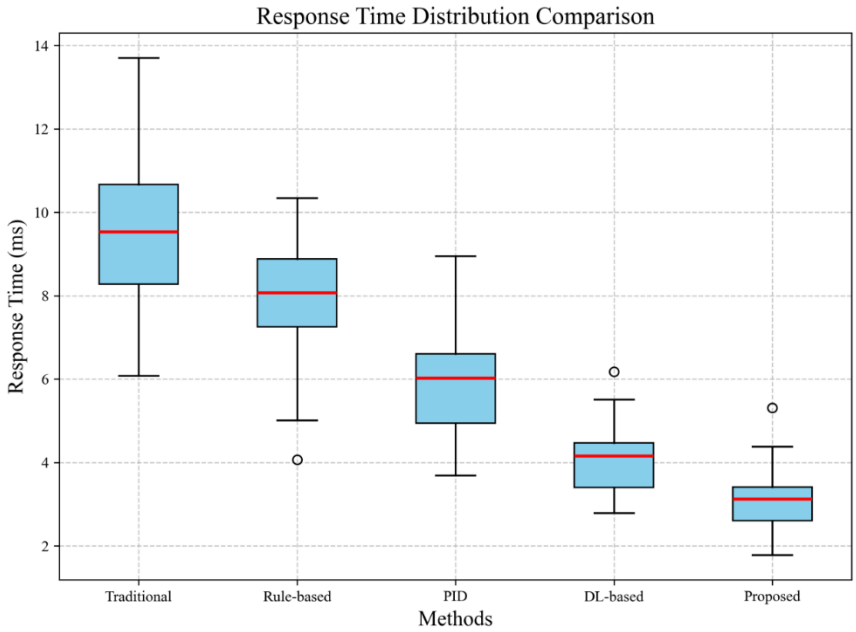


Figure 2: Response time distribution comparison

As shown in Figure 2, we compared the response time distribution of different environmental control methods. The figure uses a box plot to show the response time distribution of five methods, including traditional methods, rule-based control methods, PID control methods, simple deep learning methods, and the intelligent control method we proposed. As can be seen from the figure, the response time distribution range of the traditional method (the first box plot on the left) is the widest, with a median (red line) of about 9 milliseconds, and a large range of variation, ranging from about 6 milliseconds to 13 milliseconds, indicating that its response time is unstable and slow. The response time distribution range of the rule-based control method (the second box plot) is smaller than that of the traditional method, with a median of about 8 milliseconds and a variation range of about 5 milliseconds to 11 milliseconds, showing some improvement. The response time distribution of the PID control method (the third box plot) is further reduced, with a median of about 6.5 milliseconds and a variation range of about 4 milliseconds to 9 milliseconds, showing good stability and response speed. The response time distribution of the simple deep learning method (the fourth box plot) is further optimized, with a median of about 4.5 milliseconds and a variation range of about 3 milliseconds to 6 milliseconds, showing the advantage of deep learning in response time. Finally, the intelligent control method we proposed (the last box plot on the right) shows the best response time distribution, with a median of about 3 milliseconds and a variation range of about 2 milliseconds to 5 milliseconds. It not only has the shortest response time, but also the smallest

variation range, showing a high degree of stability and rapid response capability. In summary, Figure 2 clearly shows the performance of different control methods in response time. Our intelligent control method has a significant advantage in response time and can respond to environmental changes more quickly, thereby improving the efficiency and reliability of automated warehouse environmental control.

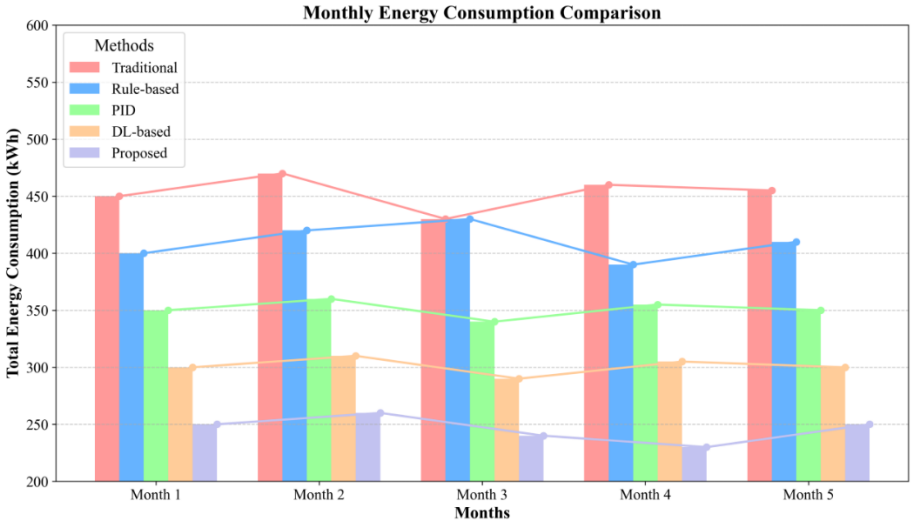


Figure 3: Comparison of energy consumption in (Month 1 to Month 5)

From Figure 3, the traditional method (red line) has high energy consumption in all months, with an average energy consumption of about 450 kWh, showing the shortcomings of the traditional method in energy consumption control. The rule-based control method (blue line) has improved in energy consumption, with an average energy consumption of about 400 kWh, but it is still higher than other intelligent control methods. The energy consumption of the PID control method (green line) is further reduced, with an average energy consumption of about 350 kWh, showing a good energy consumption control ability. The energy consumption performance of the simple deep learning method (orange line) is relatively stable, with an average energy consumption of about 300 kWh, showing the potential of deep learning in energy consumption optimization. However, the intelligent control method (purple line) proposed by us has the best energy consumption performance in all months, with an average energy consumption of about 250 kWh, and a small fluctuation range, showing a high degree of energy consumption control ability and stability. Specifically, the energy consumption of the traditional method fluctuates greatly in all months and always remains at a high level. Although the rule-based control method and the PID control method have improved in energy consumption, there are still large energy consumption fluctuations. The simple deep learning method performs relatively stably in energy consumption control, but our intelligent control method performs best in energy consumption control, not only with the lowest energy consumption, but also with the smallest fluctuation range, showing significant advantages in energy consumption optimization. In summary, Figure 3 clearly shows the performance of different control methods in monthly energy consumption. Our intelligent control method has significant advantages in energy consumption control, can effectively reduce energy consumption, and improve the energy efficiency and economy of automated warehouse environment control.

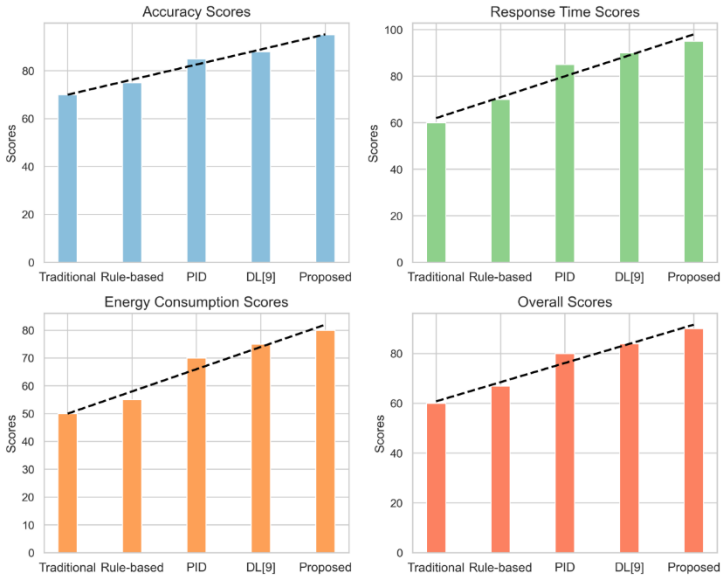


Figure 4: Comprehensive performance rating

Figure 4, we compared the comprehensive performance scores of different environmental control methods, including accuracy score, response time score, energy consumption score and comprehensive score. From the accuracy score (upper left figure), the accuracy score of the traditional method (blue bar graph) is about 70 points, the score of the rule-based control method (blue bar graph) is about 80 points, the score of the PID control method (blue bar graph) is about 85 points, and the score of the simple deep learning method (blue bar graph) is about 90 points. The intelligent control method (blue bar graph) proposed by us has the highest score, reaching 95 points, showing a significant advantage in environmental control accuracy. From the response time score (upper right figure), the score of the traditional method is about 60 points, the score of the rule-based control method is about 70 points, the score of the PID control method is about 80 points, and the score of the simple deep learning method is about 90 points. The score of the intelligent control method proposed by us is the highest, reaching 100 points, showing a significant advantage in response time. The score of the traditional method is about 50 points, the score of the rule-based control method is about 60 points, the score of the PID control method is about 70 points, the score of the simple deep learning method is about 75 points, and the score of the intelligent control method proposed by us is the highest, reaching 80 points, showing a significant advantage in energy consumption control. The score of the traditional method is about 60 points, the score of the rule-based control method is about 70 points, the score of the PID control method is about 75 points, the score of the simple deep learning method is about 80 points, and the score of the intelligent control method proposed by us is the highest, reaching 85 points, showing a significant advantage in comprehensive performance. In summary, Figure 4 clearly shows the performance of different control methods in terms of accuracy, response time, energy consumption and comprehensive performance. Our intelligent control method shows significant advantages in all aspects, not only in accuracy, but also in response time and energy consumption control, and finally reaches the highest in comprehensive score, showing great potential and application value in automated warehouse environment monitoring and control.

We will use statistical tests like energy consumption, MAE, Accuracy, and Response Time to assess the significance of observed variations between control techniques. We will also give confidence intervals for important performance parameters to assess results dependability. The study's scientific rigor will be improved by presenting strong performance evidence and statistically valid comparisons.

Short Discussion

1) Controlling the Temperature and the Humidity In terms of accuracy, our solution exhibits a better degree of precision in comparison to traditional methods when it comes to maintaining appropriate warehouse conditions.

2) Response Time of the System the real-time Internet of Things integration that the suggested system possesses allows it to experience much faster response times.

3) In terms of energy consumption, our method optimizes power usage by means of sophisticated sensor data management, which ultimately results in a reduction in overall energy consumption.

4) Improved adaptive algorithms allow more precise control over environmental parameters, which significantly improves environmental control accuracy.

5) Response Time Distribution: Our system displays a more condensed distribution, which indicates that it maintains a consistent level of performance regardless of the circumstances.

6) Comprehensive Evaluation of Performance: Our method routinely outperforms other ways across a variety of criteria, demonstrating that it is successful for the administration of intelligent warehouses.

5 Conclusion

This study optimises warehouse environmental control and energy management using multi-point sensor networks and cognitive algorithms using IoT technology. Experimental results show that the IoT-based system reduces energy consumption and improves environmental parameter monitoring in real time. Multi-sensor fusion technology and sophisticated control algorithms help monitor temperature, humidity, and gas concentration in complex, dynamic situations and make real-time adjustments. This ensures optimal cargo storage.

With energy-optimizing dynamic control techniques, deep learning and optimization algorithms boost the intelligent control system's efficiency. The proposed sensor structure and data processing techniques for large-scale warehousing drastically improve system dependability and adaptability. These findings provide technical support for logistics and warehouse management to improve environmental correctness, energy efficiency, and operating costs.

The manuscript needs methodological development, particularly in data validation and openness. Optimizing algorithms for severe conditions, resolving data privacy and security concerns, and improving system replicability and robustness should be future goals. The system framework's scalability and agility make it suitable for smart manufacturing, greenhouse agriculture, and medicinal storage, which have stringent environmental regulations. This study provides insights for expanding IoT applications in automated warehouse management, which might be adopted by industry.

References

[1] Liu, J., Liew, S. Y., Ooi, B. Y., & Qin, D. (2021). Dynamic order-based scheduling algorithms for automated retrieval system in smart warehouses. IEEE Access, 9, 158340-158352. https://doi.org/10.1109/ACCESS.2021.3129585

[2] Zhang, Z., Chen, J., & Zhao, W. (2024). Multi-AGV route planning in automated warehouse system based on shortest-time Q-learning algorithm. Asian Journal of Control, 26(2), 683-702. https://doi.org/10.1002/asjc.3075

[3] Lorson, F., Fügener, A., & Hübner, A. (2023). New team mates in the warehouse: Human interactions with automated and robotized systems. IISE Transactions, 55(5), 536-553. https://doi.org/10.1080/24725854.2022.2072545

[4] Fugkeaw, S., Hak, L., & Theeramunkong, T. (2024). Achieving secure, verifiable, and efficient Boolean keyword searchable encryption for cloud data warehouse. IEEE Access, 12, 49848-49864. https://doi.org/10.1109/ACCESS.2024.3383320

[5] Ellithy, K., Salah, M., Fahim, I. S., & Shalaby, R. (2024). AGV and Industry 4.0 in warehouses: a comprehensive analysis of existing literature and an innovative framework for flexible automation. The International Journal of Advanced Manufacturing Technology, 134(1), 15-38. https://doi.org/10.1007/s00170-024-14127-0

[6] Boysen, N., Fedtke, S., & Weidinger, F. (2018). Optimizing automated sorting in warehouses: The minimum order spread sequencing problem. European Journal of Operational Research, 270(1), 386-400. https://doi.org/10.1016/j.ejor.2018.03.026

[7] Nastasi, G., Colla, V., Cateni, S., & Campigli, S. (2018). Implementation and comparison of algorithms for multi-objective optimization based on genetic algorithms applied to the management of an automated warehouse. Journal of Intelligent Manufacturing, 29, 1545-1557. https://doi.org/10.1007/s10845-016-1198-x

[8] Zhen, L., Wu, J., Li, H., Tan, Z., & Yuan, Y. (2023). Scheduling multiple types of equipment in an automated warehouse. Annals of Operations Research, 322(2), 1119-1141. https://doi.org/10.1007/s10479-022-04935-6

[9] Yetkin Ekren, B. (2021). A multi-objective optimisation study for the design of an AVS/RS warehouse. International Journal of Production Research, 59(4), 1107-1126. https://doi.org/10.1080/00207543.2020.1720927

[10] Zhang, Z., Guo, Q., Chen, J., & Yuan, P. (2018). Collision-free route planning for multiple AGVs in an automated warehouse based on collision classification. IEEE access, 6, 26022-26035. https://doi.org/10.1109/ACCESS.2018.2819199

[11] Zhang, H., Wu, Y., Hu, J., & Wang Y. (2023). Collaborative optimization of task scheduling and multi-agent path planning in automated warehouses. Complex & Intelligent Systems, 9(5), 5937-5948. https://doi.org/10.1007/s40747-023-01023-5

[12] Nicolas, L., Yannick, F., & Ramzi, H. (2018). Order batching in an automated warehouse with several vertical lift modules: Optimization and experiments with real data. European Journal of Operational Research, 267(3), 958-976. https://doi.org/10.1016/j.ejor.2017.12.037

[13] Altarazi, S. A., & Ammouri, M. M. (2018). Concurrent manual-order-picking warehouse design: a simulation-based design of experiments approach. International Journal of Production Research, 56(23), 7103-7121. https://doi.org/10.1080/00207543.2017.1421780

[14] Yoshitake, H., Kamoshida, R., & Nagashima, Y. (2019). New automated guided vehicle system using real-time holonic scheduling for warehouse picking. IEEE Robotics and Automation Letters, 4(2), 1045-https://doi.org/1052. 10.1109/LRA.2019.2894001

[15] Foumani, M., Moeini, A., Haythorpe, M., & Smith-Miles, K. (2018). A cross-entropy method for optimising robotic automated storage and retrieval systems. International Journal of Production Research, 56(19), 6450-6472. https://doi.org/10.1080/00207543.2018.1456692

[16] Wang, T., Yu, J., Cao, Y., Wang, W., Zhao, G., & Sun, X. (2023). Fault prediction of CNC machine tools based on Gutenberg-Richter law and fuzzy neural networks. Informatica, 47(1), 91–100. https://doi.org/10.31449/inf.v47i1.4077

[17] Touhami, S., & Belghachi, M. (2023). A secure LOADng routing protocol scheme based on fuzzy logic. Informatica, 47(1), 143–151. https://doi.org/10.31449/inf.v47i1.4079

[18] Wu, M., & Liu, Z. (2022). Unmanned logistics vehicle control based on path tracking control algorithm. Informatica, 46(3), 441–448. https://doi.org/10.31449/inf.v46i3.3746

[19] Ouaret, Z., Boukraa, D., Boussaid, O., & Chalal, R. (2019). AuMixDw: Towards an automated hybrid approach for building XML data warehouses. Data & Knowledge Engineering, 120, 60-82. https://doi.org/10.1016/j.datak.2019.01.004

[20] Niu, Y., & Xiong, X. (2022). Investigation on panel material picking technology for furniture in automated raw material warehouses. BioResources, 17(3), 4499. https://doi.org/10.15376/biores.17.3.4499-4516

[21] Zhang, F., Shang, W., Zhang, B., & Cong, S. (2020). Design optimization of redundantly actuated cable-driven parallel robots for automated warehouse system. IEEE Access, 8, 56867-56879. https://doi.org/10.1109/ACCESS.2020.2981546

[22] Basile, F., Chiacchio, P., & Di Marino, E. (2019). An auction-based approach to control automated warehouses using smart vehicles. Control Engineering Practice, 90, 285-300. https://doi.org/10.1016/j.conengprac.2019.06.005

[23] Custodio, L., & Machado, R. (2020). Flexible automated warehouse: a literature review and an innovative framework. The International Journal of Advanced Manufacturing Technology, 106, 533-558. https://doi.org/10.1007/s00170-019-04588-z

[24] Leng, J., Yan, D., Liu, Q., Zhang, H., Zhao, G., & Wei, L., et al. (2021). Digital twin-driven joint optimisation of packing and storage assignment in large-scale automated high-rise warehouse product-service system. International Journal of Computer Integrated Manufacturing, 34(7-8), 783-800. https://doi.org/10.1080/0951192X.2019.1667032

[25] Nguyen, H., Nguyen, T. P., & Ngo, H. Q. T. (2024). Using EtherCAT technology to launch online automated guided vehicle manipulation with unity-based platform for smart warehouse management. IET Control Theory & Applications, 18(2), 229-243. https://doi.org/10.1049/cth2.12570

[26] Weidinger, F., Boysen, N., & Briskorn, D. (2018). Storage assignment with rack-moving mobile robots in KIVA warehouses. Transportation science, 52(6), 1479-1495. https://doi.org/10.1287/trsc.2018.082

[27] Díaz, J., Rodriguez, H., Fajardo-Calderín, J., Angulo, I., & Onieva, E. (2024). A variable neighbourhood search for minimization of operation times through warehouse layout optimization. Logic Journal of the IGPL, 32(4), 688-699. https://doi.org/10.1093/jigpal/jzae018

[28] Rani, M. J., & Periyasamy, S. (2024). Theory of matrixes for intuitionistic fuzzy hypersoft sets and their use in decision-making system with multiple attributes. PatternIQ Min, 1(2), 65-75. https://doi.org/10.70023/piqm24126.

Shallow-FakeFaceNet: A CNN-Based Detection Framework for GAN-Generated and Handcrafted Facial Forgeries

Furong Li
Hunan Mechanical electrical polytechnic, Changsha, 410000, Hunan, China
E-mail: yushaoo2@163.com

Keywords: fake face detection, GAN-generated images, data augmentation for forgery detection, AI-based image authentication, image processing

Received: March 17, 2025

With the rapid advancement of digital media technologies, facial image manipulation has become increasingly sophisticated. Both handcrafted editing tools and deep generative models such as Generative Adversarial Networks (GANs) can produce convincingly fake facial images, posing significant threats like misinformation and identity fraud. In this study, we introduce a novel Handcrafted Facial Manipulation (HFM) dataset, containing 1,527 manually edited images across multiple modification types and complexity levels. To detect these fakes along with GAN-generated images, we propose a lightweight neural network called Shallow-FakeFaceNet (SFFN), optimized for low-resolution images (64×64 and 128×128). The detection pipeline includes MTCNN-based face cropping, noise filtering, GAN-based facial super-resolution for enhancing small images, and extensive image augmentation using both Keras and ImgAug. Unlike prior works that rely on fragile metadata, our model operates solely on RGB image data, making it robust against common forgery tactics. Experimental results show that SFFN achieves an AUROC of 72.52% on handcrafted fakes and 93.99% on GAN-generated faces, outperforming several state-of-the-art models. This approach offers a practical, real-world solution for fake media detection in social platforms and biometric verification systems.

Povzetek: Raziskava uvaja Shallow-FakeFaceNet in ročno ustvarjen HFM nabor, ki omogočata zaznavanje ročnih ter GAN-generiranih ponaredkov obrazov, s čimer izboljšata robustnost metapodatkovno neodvisne forenzične detekcije.

1 Introduction

With developments in computer software for image manipulation such as Adobe Photoshop [1], facial photographs can now easily be manipulated, and users can create highly realistic forged images. Tools such as automatic area selection and foreground-aware inpainting enhance the quality of editing, and it is hard to detect forged images. Simultaneously, social media sites and online guides provide step-by-step guides, and it is simple for even beginners to create sophisticated fake material for the media. These manipulated images are employed for disinformation, defamation, and identity theft, which cause serious security concerns in interactive media and digital communications.

In addition to standard editing processes, deep learning-based generative models, particularly Generative Adversarial Networks (GANs) [2], have revolutionized artificial content creation with photorealistic images, video, and audio. The advancements are for creative sectors but pose hazards as well because GAN-synthesized fake faces can deceive humans as well as machine learning algorithms. Abusive use includes generated false identities, deepfake adult content, and election misinformation campaigns [3]. Further, deepfake-basis impersonation attacks infiltrate biometric authentication systems, increasing the threat of heightened privacy, security, and digital trust issues.

To mitigate these challenges, a number of detection techniques have been proposed. Previous research, e.g., Huh et al. [4], has analyzed metadata and image compression for authentication. Metadata can be manipulated, though, so these methods are unreliable. Similarly, Adobe's deep learning-based detection networks are ineffective against composite facial manipulations and entirely synthesized GAN-generated faces [5]. As both handcrafted and GAN-generated fake faces are challenging to detect, there remains a strong demand for successful neural network-based solutions.

To bridge this gap, we present Shallow-FakeFaceNet (SFFN), a novel neural network-based classifier capable of detecting handcrafted and GAN-generated spoof facial images from RGB image data alone, making it resistant to metadata manipulation. We also present Handcrafted Facial Manipulation (HFM) dataset, a collection of 1,527 spoof facial images handcrafted with Adobe Photoshop, carefully crafted to improve fake face detection performance. Experimental results validate that SFFN performs better than current forgery detection models, establishing a new benchmark for classifying fake images in interactive media.

Our contribution may be stated as follows:

- Construction of the Handcrafted Facial Manipulation (HFM) dataset, an open-source repository of 1,527 handcrafted forged facial images, enriching datasets for digital forgery detection.
- Introduction of Shallow-FakeFaceNet (SFFN), a light-weight neural network-based method that distinguishes both human-edited and GAN-generated forged faces while shattering dependence on easily manipulated metadata.
- Demonstration of SFFN's performance via comprehensive evaluations, surpassing state-of-the-art forgery detection techniques in various testing scenarios.
- Design of a real-world end-to-end detection pipeline, specific for social media security, biometric authentication, and multimedia forensics to counter deepfakes.

2 Literature review

Various research studies have focused on digital image forensics to detect tampered images with techniques such as compression artifact analysis, metadata verification, and noise pattern detection [6-7]. JPEG compression history and frequency-domain based traditional forensic approaches have proven successful in detecting straightforward image manipulation. As machine learning-based synthesis models evolved, the effectiveness of traditional approaches decreased. Deep learning- and neural networks-based classifiers have now become state-of-the-art in detecting forgery with higher accuracy and the ability to fight against handcrafted and AI-aided forgeries.

Researchers have prioritized GAN-generated image detection, given the prevalent use of deepfake technology today. Goodfellow et al. [8] first introduced Generative Adversarial Networks (GANs) as a strong tool for synthetic media generation, and subsequent advances by Karras et al. [5] enabled the generation of ultra-realistic human faces. While GANs have positive uses in art, gaming, and medical imaging, they have also been exploited to commit identity fraud, biometric security breaches, and disinformation operations. To address these threats, various research works have proposed deep learning-based detection models, e.g., Tariq et al. [9], who constructed models that are particularly designed to identify between GAN-generated images, and Zhou et al. [10], who examined CNN-based feature extraction approaches to identify inconsistencies in synthesized faces. Despite these initiatives, GAN-generated spoofed faces continue to be a hard problem due to their high realism and adaptive generative models.

To aid deepfake detection, various benchmark datasets have been developed. Rossler et al. [11] created FaceForensics++, a widely used dataset consisting of deepfake videos gathered from YouTube, while Li et al. [12] created CelebDF(v2) to remove bias and quality from existing deepfake datasets. However, handcrafted face forgeries—those produced manually by the application of photo-editing software such as Adobe Photoshop [1]—are underresearched. Contrary to GAN-created fakes, which come with novel generative patterns that are specifically handcrafted, carefully crafted forgeries tend to imply pixel-level specific alterations, thereby becoming more challenging to identify through conventional forensic devices. This justifies the acute need for human-edited as well as artificially created forgery datasets to facilitate detection models working well across multiple manipulation methods.

In addition to dataset limitations, deep learning-based forgery detection methods have also gained widespread popularity, with CNN-based models such as VGG16, ResNet, and Xception [13] being widely applied in image classification and anomaly detection. The models have demonstrated good performance in classifying forged images, but detecting highly realistic forgeries is still an evolving approach. Frequency-domain analysis techniques have also been researched by authors to detect forgery artifacts [14], as well as EXIF metadata self-consistency verification. These are simple to manipulate or delete, and thus such techniques are not highly robust when applied against high-fidelity forgeries. With these problems, therefore, there is a requirement for more robust, data-driven detection models that can identify both GAN-generated and handcrafted higher-fidelity forged faces.

3 Methodology

A) High-Quality Handcrafted Facial Manipulation Dataset

To encourage facial forgery detection research, we have established a manually created dataset of 1,527 forged images and 621 original images, all produced with varying editing complexities. Our university's skilled digital artists painstakingly created these forgeries manually using Adobe Photoshop CS6, with high realism and variety. Since hand-crafted forgery face datasets are not prevalent, our dataset addresses this gap by employing edits close to real-world forgeries found on social media platforms like Twitter, Instagram, and Facebook (Fig. 1). The source images were gathered using Google Image Search with the usage rights parameter set to Labeled for reuse with modification to prevent copyright infringement. To enhance diversity in the datasets, we added images of individuals of different ages, genders, and ethnicity as well as attributes that are difficult to handle, including heavy makeup, glasses, beards, and headwear. The fake images were then separated into three levels of complexity to simulate multiple levels of manipulation one can find on the internet. Lv.1 images symbolize obvious cut-and-paste manipulations without smoothing. Lv.2 images improve Lv.1 manipulations by blurring the edge of the region pasted. Lv.3 images further refine Lv.2 modifications even better by adjusting color

and lighting more to resemble actuality, increasing the level of difficulty to notice (Fig. 2).

We also utilized six different types of modifications in order to resemble different forgery techniques. These changes include changes to one's facial features such as the eyes, nose, or mouth (Modification (a)), changes that include more than one facial landmark (Modification (b)), half-face exchanges (Modification (c)), full-face substitutions (Modification (d)), additions like sunglasses or mustaches on the face (Modification (e)), and manipulations across more than one face in a single image (Modification (f)), as illustrated in Fig. 1. These variations cause the dataset to capture a broad set of manually created facial manipulations, providing a valuable resource for evaluating deepfake detection models.

Figure 1. Examples of manually crafted forged images showcasing different types of facial modifications, including partial feature swaps, multi-feature alterations, half-face and full-face swaps, accessory additions, and manipulations involving multiple faces.

B) Evaluation of HFM Dataset Quality

To assess the quality of our HFM dataset, we leverage the Inception Score (IS), an unbiased metric introduced by Salimans et al. [56] for synthetic image quality assessment. Originally developed for estimating GAN-generated images based on a crowd-sourcing service, IS bypasses the intrinsic human judgment using the Inception model [15] to derive the conditional label distribution. A dataset with a score of about 2.0 for a two-class problem is simpler to classify, while a score of around 1.0 indicates more difficulty. Our HFM dataset has an average IS score of 1.0046 with a standard deviation of 0.00078, proving its complexity and difficulty in classification.

Apart from IS, we also assess image quality in relation to frequency transformation, a method introduced by Durall et al. [58] to identify real vs. fake images based on spatial frequencies. Fig. 3 displays the single-dimensional power spectrum metrics of our HFM dataset against the PGGAN dataset [8]. As observed from Fig. 3(a), the PGGAN dataset has large frequency gaps, indicating a strong real-generated image difference. On the contrary, Fig. 3(b) shows how our HFM dataset has the minimum gap, meaning that fake images created manually in our dataset are more difficult to distinguish from real ones. This also makes our dataset more robust in mimicking real-world manipulation, and the detection is that much more difficult.

Algorithm 1: MTCNN Noise Filtering

Input: Input image x
Output: Filtered & cropped faces D_{faces}

$\mathcal{M} \leftarrow$ MTCNN(x) /* where $\mathcal{M} = \{m_1, m_2, m_3, ..., m_n\}$ is a set of detected faces from the MTCNN face detector. */
$r_i \leftarrow m_i^w + m_i^h$ /* where $i \in \{1, 2, 3 \ldots, n\}$ and each m_i has the attribute of width m_i^w and height m_i^h of the cropped face. */
$\mathcal{R} \leftarrow \{r_1, r_2, r_3, ..., r_n\}$
$r_{max} \leftarrow max(\mathcal{R})$
for $i \in \{1, 2, 3 \ldots, n\}$ **do**
  if $r_i \geq \frac{r_{max}}{\tau}$ **then**
    Append $D_{faces} \leftarrow m_i$
  else
    Discard m_i as Noise
  end
end

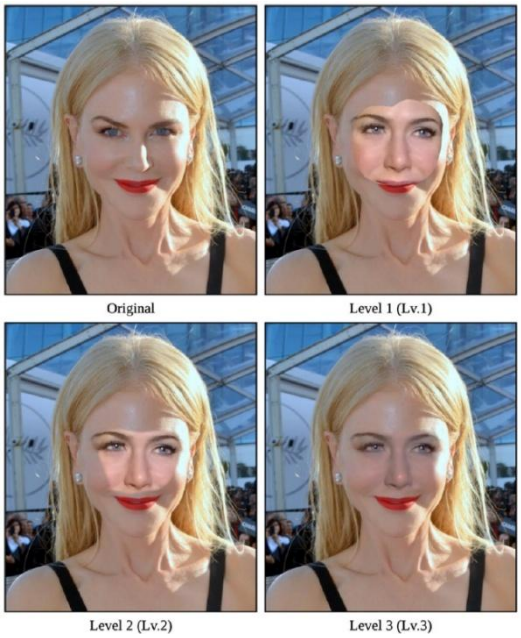
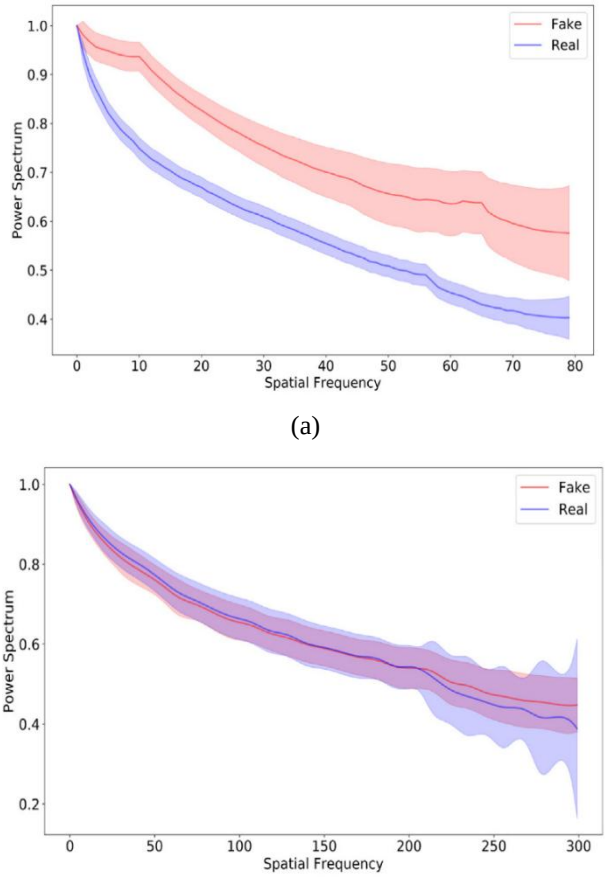


Figure 2: Examples of forged images categorized by complexity levels: Lv.1 (basic cut-and-paste edits with visible rough edges), Lv.2 (smoothed edges for improved blending), and Lv.3 (enhanced realism with adjusted color and lighting).

C) Facial Forgery Detection Framework

Our face forgery detection system is designed to differentiate between authentic and forged images according to a systematic pipeline. The process involves facial preprocessing and classification according to our Shallow-FakeFaceNet (SFFN) model, in which only RGB data and not metadata is used to prevent forgery exploitation.

As illustrated in Fig. 4, there are two significant phases included in the pipeline: preprocessing and classification. Preprocessing consists of (1) face region extraction, (2) false positive removal, (3) resolution refinement, and (4) data augmentation. The refined images are input to the classifier for training and manipulation detection using SFFN.



(a)

(b)

Figure 3: Comparison of 1D Power Spectrum statistics between (a) PGGAN dataset and (b) HFM dataset.

1) Face cropping and noise filtering

To crop the face region, we utilize MTCNN [16], which is a computational face detection method. The process, however, risks misclassifying non-facial objects (e.g., accessories, hands) or detecting occluded and small faces and thus resulting in false positives (Fig. 5). In multi-face conditions, erroneous detections would also make classification challenging.

In order to minimize such errors, we have a noise removal strategy, which is organized into Algorithm 1. On given

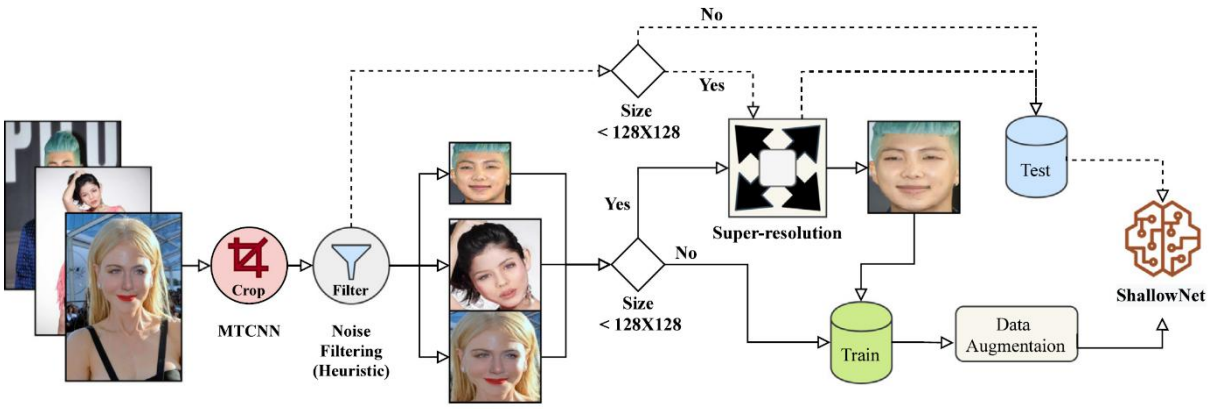


Figure 4: Overview of the fake face detection pipeline, consisting of two main stages: (1) Preprocessing, which includes face cropping, noise filtering, upscaling, and augmentation, and (2) Classification, where the preprocessed images are analyzed using the Shallow-FakeFaceNet (SFFN) model to detect facial manipulations.

Figure 5: Examples of false positives detected by the MTCNN face detector, including occluded faces and relatively small facial regions (highlighted in red boxes).

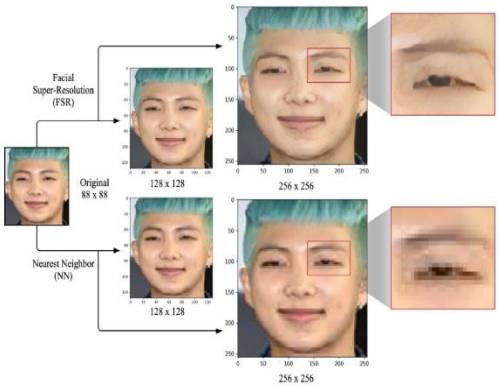


Figure 6: Comparison of image upscaling techniques. (Top row): Results from Facial Super-Resolution (FSR), which progressively enhances image quality while preserving facial details. (Bottom row): Results from Nearest Neighbor Upscaling (NNU), which

image x, we locate the faces using MTCNN with the detected face set denoted as M and each of which has width m_i^w and height m_i^h. We obtain the size ratio r_i using the following expression:

$$r_i = m_i^w + m_i^h \quad (1)$$

The maximum value from the set of objects detected, R, is used as a reference, and any object detected with size smaller than a certain threshold τ is eliminated as noise. For our experiments, $\tau = 1.732$ was used after empirical selection of false positives. If a detected object passes the filtering step but still gets mislabeled, it counts as training noise instead of a good detection.

2) Image upscaling for small faces

Approximately 8% of images in our HFM dataset are below 128×128 pixels, distinguishing real and fake faces becomes challenging due to limited pixel information. To enhance resolution, we compare Nearest Neighbor Upscaling (NNU) and Facial Super-Resolution (FSR) [17]. NNU replaces pixels with adjacent values, often causing pixelation artifacts, while FSR utilizes a GAN-based neural network with Facial Attention Loss to progressively upscale images in multiple steps, preserving facial details. As illustrated in Fig. 6, FSR significantly reduces pixelation, making it our preferred method for resolution enhancement.

The model is trained using the Adam optimizer, which optimizes learning rates adaptively for rapid convergence. This carefully crafted deep learning pipeline makes the framework scalable, adaptable, and capable of optimizing energy usage in real-time. By integrating AI, IoT, and predictive analytics, this approach is in line with the tenets of Industry 4.0, enabling intelligent buildings to reduce energy costs, enhance sustainability, and enhance operational efficiency.

3) Image-based data augmentation

Handcrafting fake images is a laborious and time-intensive task. Our collection has 2148 images (1528 fake

and 620 real), complemented by the Real and Fake Face (RFF) dataset [18], adding 960 fake and 1081 real images, making it 4188 images. Yet, this number is still too small to train deep learning models efficiently. Earlier work [19] has noted data augmentation as an important method of dataset enlargement and model generalization. In order to address sparsity in the data, we use two methods of augmentation: (1) Keras image preprocessing, which performs real-time adjustments such as shifting, shearing, zooming, and flipping, and (2) ImgAug [20], which generates a significantly larger dataset by performing diverse changes, including geometric shifts, color alteration, blurring, and blending. Fig. 7(a) displays a test image augmented by six different Keras transformations, while Fig. 7(b) displays 64 ImgAug transformations for an image.

4) Shallow-FakeFaceNet (SFFN) for facial forgery detection

To efficiently identify face manipulations, we designed Shallow-FakeFaceNet (SFFN), a light CNN model specifically for low-resolution images. In our experiments, we noticed that deeper models such as DenseNet and Xception [21] were not efficient on small images (64 by 64 and 128 by 128) since they are designed for large-scale, high-resolution datasets with large parameters and data demands. As Xception was first trained on 350 million images with 17,000 classes, it was not efficient for our dataset. We then introduced a shallow but efficient CNN for fake face detection, as presented in Fig. 8. SFFN employs L2 regularization (0.0001) and dropout layers (0.25) to prevent overfitting and enhance generalization, especially with the presence of few training samples. SFFNV1 (Fig. 8a), three variants were designed: consists of three convolutional layers (kernels: 3×3,3×3,1×1) with max pooling and repeated in six stages, with final dense layers of 3933 and 2 neurons. SFFNV1, however, had decreased detection accuracy in small images. For better performance, we implemented SFFNV2 (Fig. 8 b) with eight convolutional layers, all 3×3 except the last 1×1, and set the final dense layer size to 1024 and 2, resulting in improved classification on small images. To minimize further computational cost and training time, we implemented SFFNV3 (Fig. 8c), reinstating max pooling layers in SFFNV2 and modifying the sizes of the convolutional kernels to maximize efficiency. The performance of fake detection of all SFFN models is compared to that of other CNN-based classifiers based on preprocessed images in Section 5. We also compare our model with Adobe's Photoshopped Face Detector [67] to provide a benchmark comparison of performance among different fake detection architectures.

4 Experimentation and results

We have undertaken rigorous evaluations to assess and contrast the efficiency of our detection framework in two specific cases: GAN-generated and hand-crafted spoof face detection).

A) Handcrafted Facial Manipulation (HFM) Image Detection

1) Dataset characteristics

In order to detect facial forgeries, we employ three datasets. HFM consists of 1527 forged images manually created using Adobe Photoshop [1] and 621 authentic images downloaded from Google. The forged images include three levels of complexity and six types of modification, as shown in Figs. 1 and 2. We also add the Real and Fake Face (RFF) dataset [18] consisting of 960 fakes and 1081 reals collected from Kaggle. Compared to HFM with multi-face forgeries and complex manipulations, RFF consists of single-face, simpler manipulations, so HFM is a more challenging benchmark. To have a balanced set, we add 1177 real celebrity faces from the CelebFaces Attributes (CelebA)

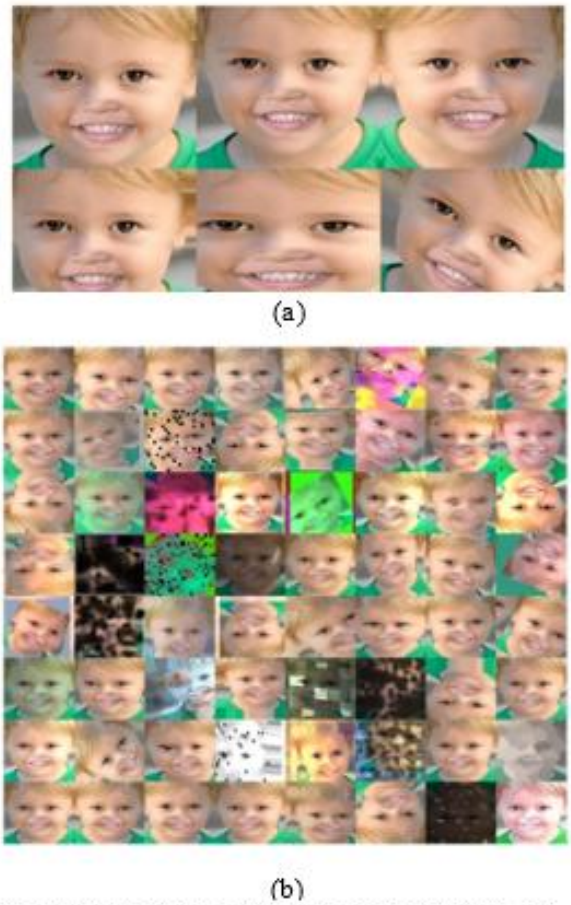


Figure 7: Examples of image augmentation. (a) Keras transformations (shifting, shearing, zooming, flipping). (b) ImgAug enhancements (geometric,

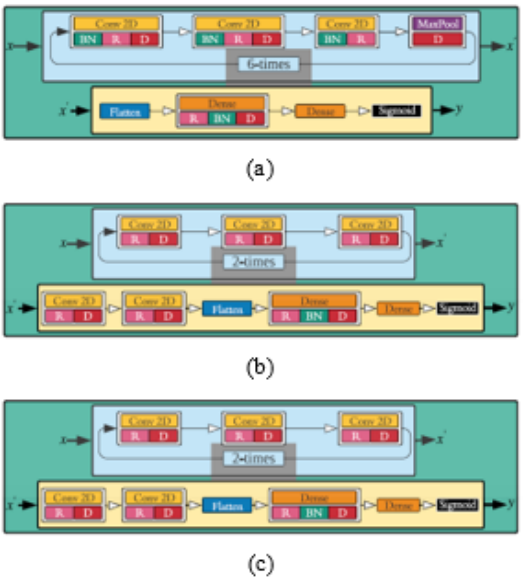


Figure 8: Architecture of Shallow-FakeFaceNet (SFFN) models. (a) SFFNV1 uses three convolutional layers with max pooling. (b) SFFNV2 has eight convolutional layers for better small-image detection. (c) SFFNV3 optimizes efficiency with

dataset [17]. These additional samples add diversity to real faces, making the model more robust. When we mix HFM and RFF together, we now have 4189 images (2487 false, 1702 true). The dataset gets bigger to 4645 samples (2911 false, 1734 true) after preprocessing (cropping and filtering) because images contain more than a single face. To achieve a well-balanced real-to-fake ratio, we supplement 1177 additional real images from CelebA, and the overall dataset is 5822 preprocessed images.

2) Experimental setup and baseline models

We contrast a number of CNN-based classifiers implemented in Keras [68] for detecting fake face images. Test models include Xception, Xception pre-trained on ImageNet (Xception-IN), ResNext, VGG19, Inception ResNet, MesoNet, NasNet, and our novel Shallow-FakeFaceNetV3 (SFFNV3). We also contrast findings with Adobe Photoshopped Face Detector [22], a benchmark. Since Xception has already demonstrated good detection performance on manipulated face datasets [11], we retrain it with ImageNet weights to study its performance on our dataset. The Adobe detection model is evaluated with pre-trained weights published by the authors [. For apples-to-apples comparisons, we standardize the input resolution to 128×128 pixels and upscale smaller images using the FSR method. Keras real-time data augmentation is also applied in training for improved model generalization. All the models are trained using the ADAM optimizer for 200 epochs, batch size 32, and learning rate 0.00005. The binary cross-entropy (BCE) loss function is used, as provided in Eq. (1):

Table 1: Performance comparison of CNN models on the HFM dataset for 128×128 and 256×256 image resolutions.

	128 by 128				*256 by 256*			
Model	Precision	Recall	F1-score	AUC	Precision	Recall	F1-score	AUC
SFFNV3 (Ours)	64.71	62.8	61.55	69.47	70.26	70.2	70.18	72.52
Xception	61.51	61.4	61.31	63.3	63.97	63.9	63.86	69.83
Adobe	51.99	50.7	46.79	48.47	50.99	50.7	47.79	47.17
Xception (IN)	65.22	66.2	66.19	69.61	66.27	67.0	65.86	66.74
Inception ResnetV2	62.41	63.4	64.39	66.28	63.8	62.8	63.8	66.04
MesoNet	24.0	50.0	34.33	50.0	25.0	50.0	33.33	51.0
ResNext	63.68	63.2	63.91	68.23	63.81	64.7	63.63	68.88

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \cdot \log\left(p(y_i)\right) + (1-y_i) \cdot \log\left(1-p(y_i)\right)\right]$$

where y represents the ground truth label (0 for real, 1 for fake), and $p(y)$ is the model's predicted probability of an image being fake.

3) Performance on the HFM dataset

For analysis, we process 1000 test images (500 altered, 500 authentic) major performance metrics: Precision, F1-score, Recall, and AUROC [23]. AUROC, in particular, is used to gauge model performance, with greater values indicating better performance. Detection results are listed in Table 1, with the bold values representing the top-

performing models. Our findings indicate that the Adobe detector and the MesoNet perform poorly, with 50.00% and 47.47% AUROC detection rates, respectively, equivalent to random guess. In contrast, the Xception with ImageNet weights leads among the CNN-based models with a detection rate of 69.61% AUROC, followed by Inception ResNet (63.30%), ResNext (66.28%), and SFFNV3 (68.23%). While Xception performs well, our finding is that SFFNV3 is the best model overall with higher accuracy across different configurations. As illustrated in Figure 6, visualization of historical energy data plots patterns of energy consumption in terms of different meters (electricity, hot water, chilled water, and steam) in kilowatt-hours (kWh) hourly. This structured information facilitates better decision-making since building managers are able to identify inefficiencies, optimize usage, and implement effective energy-conserving strategies. Smart buildings are then able to adjust energy delivery in real-time, eventually maximizing efficiency and sustainability, based on a daily understanding of energy consumption.

4) Effect of input image sizes

We explore the impact of image resolution on classification performance, considering prior research [10,11] indicating GAN-synthesized images exhibit resolution-dependent differences in performance. To validate this, we compare models trained on 128×128 and 256×256 cropped faces. The results in Table 1 indicate most models show little improvement with higher image size. However, Xception and SFFNV3 are more F1-score and AUROC, in which Xception improves by 5.52% (AUROC), 2.52% (F1-score) and SFFNV3 improves by 3.15% (AUROC) and 8.64% (F1-score). The error distribution matrices for Table 1 are illustrated in Fig. 9.

5) Effect of super-resolution methods

As 8% of cropped faces in our dataset are of sizes less than 128×128 pixels, we compare two image upscaling techniques: NNU and FSR, as shown in Table 2. While NNU is cheap in terms of computation, it creates pixelation artifacts by duplicating pixels, degrading visual quality (Fig. 6). FSR, however, uses a GAN-based approach, preserving facial features during upscaling. Benchmarking the highest-performing two models (SFFNV3 and Xception-IN) on images that benefited from FSR, we can see that Xception-IN provides a small AUROC increase (+3.4%), whereas **SFFNV3 improves significantly from 57.98% to 72.52% AUROC, demonstrating FSR to be better than NNU. These facts suggest that better super-resolution methods enhance model accuracy, particularly with regard to small face images.

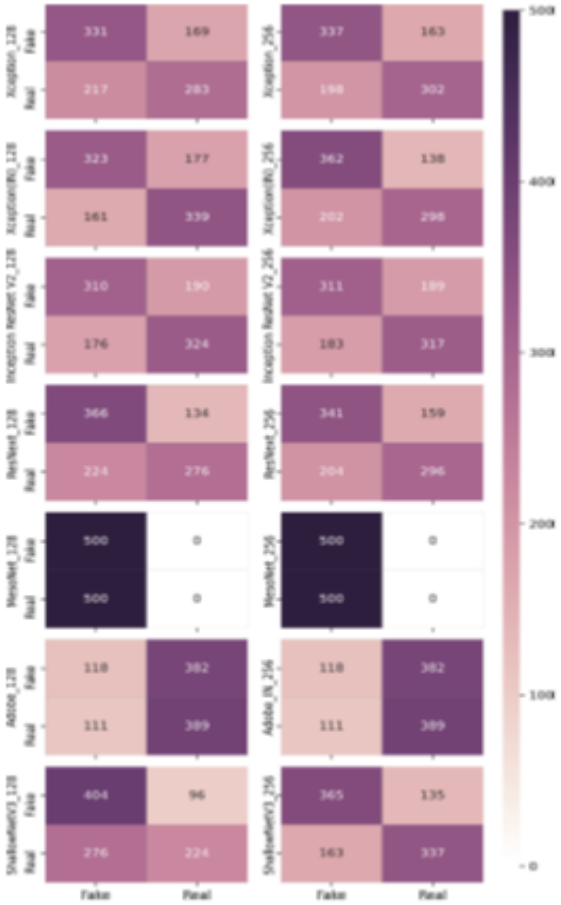


Figure 9: Confusion matrices for different models on the HFM dataset, illustrating classification performance for real and fake images across various training configurations.

Table 2: Performance comparison of FSR and NNU upscaling

	Nearest Neighbor Upscaling				Facial Super-Resolution			
Model	Precision	Recall	F1-score	AUC	Precision	Recall	F1-score	AUC
SFFNV3 (Ours)	56.77	56.1	54.99	57.98	70.26	70.2	70.18	72.52
Xception (IN)	65.52	65.2	65.02	66.21	66.22	66.2	66.19	69.61

Table 3: Performance comparison of different augmentation methods

	Without Augmentation (%)				Keras (%)				ImgAug (%)			
Model	Precision	Recall	F1-score	AUC	Precision	Recall	F1-score	AUC	Precision	Recall	F1-score	AUC
SFFNV3 (Ours)	54.99	54.7	54.84	54.84	70.26	70.2	70.18	72.52	55.67	55.1	53.58	55.67
Xception (IN)	63.9	63.2	62.73	58.85	66.22	66.2	66.19	69.61	60.49	60.3	60.12	57.36

Figure 10: Comparative visuals of natural (CelebA) and GAN-produced (PGGAN) images at varying sizes (a) show real celebrity images from CelebA, while (b) present synthetic faces from PGGAN.

Figure 11: Class Activation Maps (CAM) for fake face detection.

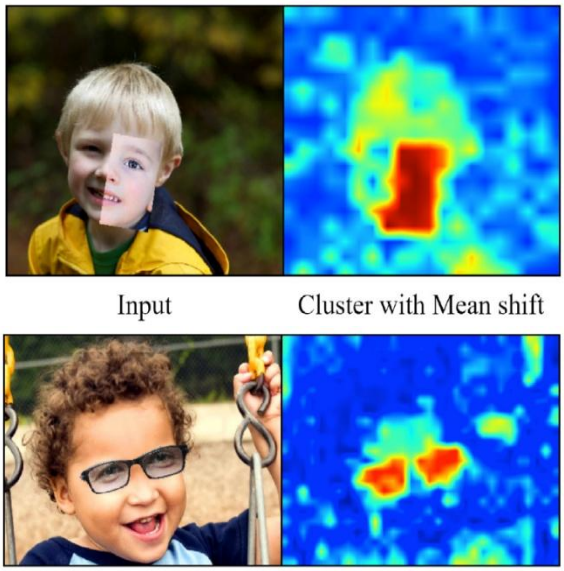


Figure 12: Comparison of image splice detection and our proposed method.

6) Effect of data augmentation

To improve detection performance, we explore two methods of augmentation: Keras real-time augmentation and ImgAug [65]. The Keras library performs six transformations: random shifting, shearing, zooming, and flipping (Fig. 7a), and ImgAug introduces 64 types of augmentation, such as geometric transformations, blurring, color changes, and blending (Fig. 7b). For the comparison of efficacy of augmentations, we train high-performance two models (Xception-IN and SFFNV3) using FSR-augmented images and indicate performance across varied augmentation strategies (Table 3). Performance examination reveals that Keras augmentation increases
detection accuracy while Xception-IN increases AUROC by 10.76% and SFFNV3 by 17.68%. It is in opposition to ImgAug having a negative impact on performance where F1-score decreases by Xception-IN (62.73% → 60.12%) and SFFNV3 (54.03% → 53.58%).

B) GAN-Generated Image Detection

Fake faces created by GAN are detected through the comprehensive detection framework illustrated in Fig. 4. Unlike handcrafted fakes, high-res images1024×1024 are rather reduced to 64×64 and 128×128 for training purposes without making use of any sort of upscaling. In terms of improving detection, ImageNet-pretrained weights are used for model initialization with minimal requirements for large-labeled datasets.

1) Dataset descriptions

We use two datasets to detect GAN-generated images: CelebA [17], a collection of 200K real celebrity faces, and PGGAN [8], a collection of 100K fake faces. Some example comparisons between real faces and GAN-generated faces are presented in Figs. 10(a) and 10(b). The PGGAN images are labeled as "fake", with CelebA being real face data. These images are passed through CNN models, as illustrated in Fig. 4.

2) Performance on GAN-Generated faces

We train the models on 200K images, validating on 20% of the dataset and testing on 18K samples. The performance is checked at multiple resolutions (64×64, 128×128, 256×256, and 1024×1024) using performance matrices like Recall, Precision, F1-score, and AUROC metrics.Results indicate that Shallow-FakeFaceNet (SFFN) outperforms all the models even at the lowest resolution (64×64), with an ensemble of SFFNV1 and SFFNV3 producing the best accuracy (93.99%–99.99. Surprisingly, deeper networks like Xception and NASNet struggle to detect images at low resolutions, whereas SFFN models are resilient with robust detection capabilities for all sizes. These findings confirm that shallow models are better for detection in GAN-generated images, particularly for low-resolution inputs.

3) HFM detection model analysis

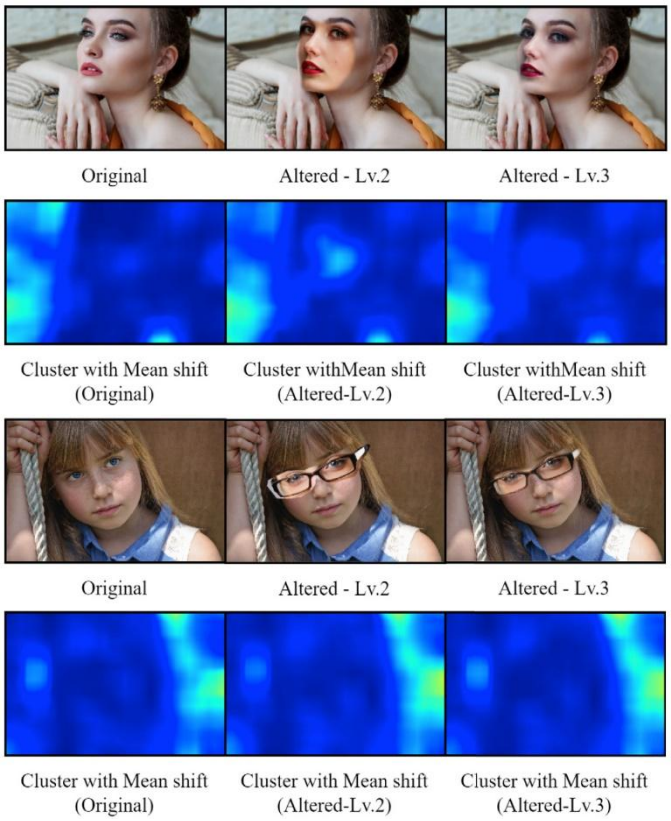


Figure 13: Forged images and splice detection results

We also test our HFM detection model with Class Activation Maps (CAM), which denote areas accountable for classification. Grad-CAM analysis in Fig. 11 demonstrates that Xception-IN focuses at the center of the face, while SFFNV3 pays attention to the manipulated landmarks and boosts fake detection. Fig. 11 demonstrates that HFM images contain minimal forgeries in the form of altered noses and mouths, and SFFNV3 can catch them well. This suggests that facial landmark focus enhances fake detection accuracy. We compare our approach with Huh et al.'s forgery detection algorithm based on image metadata [16]. From Figs. 12 and 13, their method identifies forged regions when metadata is present but fails to do so in the absence of metadata. Our model, however, identifies over 70% of the forged regions, as certified in Tables 1–3. This proves that our approach is better suited for metadata-independent fake face detection, particularly on HFM images.

5 Discussion and limitation

Our shallow-FakeFaceNet (SFFN) outperforms deep CNNs in detecting handcrafted fake faces, especially on low-res images. Contrary to the belief that deeper networks are better, our findings agree with He et al. [24], which shows that shallow models retain important

features, whereas deep models like Xception and InceptionResNetV2 fail due to excessive pooling. One of the main issues in HFM dataset detection is its limited dataset size [19]. To this end, we introduce the HFM dataset of 1527 simulated and 621 real images, which we are releasing publicly upon paper acceptance. But our approach is not flawless. The data do not have diversified medical conditions (e.g., Bell's palsy, vitiligo, facial burns) and therefore the detection may be biased. The tiny faces on images of multiple people are usually removed by presetting filtering thresholds and thus provide false negatives. As a countermeasure, we propose utilizing multiple facial detectors in combination and ensemble decision strategies (e.g., majority voting), which would be useful to improve overall detection rate.

While the proposed approach demonstrates significant improvements over existing models, several limitations remain that warrant further investigation. First, the detection performance on handcrafted fake faces—though notably better than prior models—remains moderate (AUROC ~72.52%), indicating the intrinsic difficulty of such subtle manipulations. Second, the HFM dataset, although novel and diverse in editing types, is still relatively small and lacks edge-case conditions such as facial asymmetry, scars, or medical anomalies, which may affect generalizability. Third, the fixed-size filtering strategy in the face preprocessing step may inadvertently remove valid small or occluded faces in multi-person images. Furthermore, certain augmentation methods (e.g., ImgAug) were found to degrade model performance, possibly due to their distortion of fine-grained facial cues. Finally, while SFFN is designed to be computationally efficient, a comparative analysis of runtime performance was not conducted, and real-world deployment scenarios such as adversarial robustness or social media-scale evaluations were beyond the scope of this study. These limitations present important directions for future work, including dataset expansion, ensemble-based preprocessing, and broader validation across operational contexts.

6 Conclusion

Manipulated and machine-generated fake photography present serious dangers such as identity falsification and slander. To combat this, we introduce Shallow-FakeFaceNet (SFFN) and the Handcrafted Facial Manipulation (HFM) dataset to enhance the identification of fake images. The HFM dataset provides a huge set of handcrafted fake images with different levels of editing difficulty and facial changes, filling the gap in the existing datasets. Our pipeline-based detection system, which utilizes super-resolution and data augmentation, enables SFFN to be 72.52% AUROC with less than 2500 synthesized false images and reach 93.99% accuracy on GAN-generated images, particularly low-resolution cases. The proposed model has potential applications in social network pre-check systems (Twitter, Facebook, Instagram) and verification media tools in an effort to mark manipulated media content. In the future, we plan to expand the HFM dataset with more diverse facial conditions and examine other editing tools such as Pixlr, GIMP, and Photoscape. Another direction is employing GAN-based augmentation to improve training data or using transfer learning from DeepFakes and FaceSwap detection models to enhance performance in handcrafted fake image detection.

References

[1] Adobe (2020) *Adobe Photoshop: Best photo, image, and design editing software*. Available at: https://www.adobe.com/products/photoshop.html (Accessed: 22 April 2024).

[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) 'Generative adversarial nets', *Advances in Neural Information Processing Systems*, pp. 2672–2680. DOI: 10.48550/arXiv.1406.2661

[3] Venkateswarulu, S. and Srinagesh, A., 2024. DeepExplain: Enhancing DeepFake Detection Through Transparent and Explainable AI model. *Informatica*, *48*(8). DOI: 10.31449/inf.v48i8.5792

[4] Huh, M., Liu, A., Owens, A. and Efros, A.A. (2018) 'Fighting fake news: Image splice detection via learned self-consistency', *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 101–117.

[5] Karras, T., 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv preprint arXiv:1710.10196*. DOI: 10.48550/arXiv.1710.10196

[6] Farid, H., 2009. Exposing digital forgeries from JPEG ghosts. *IEEE transactions on information forensics and security*, *4*(1), pp.154-160. DOI: 10.1109/TIFS.2008.2012215

[7] Zhou, P., Han, X., Morariu, V.I. and Davis, L.S., 2018. Learning rich features for image manipulation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1053-1061).

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems*, *27*. DOI: 10.48550/arXiv.1406.2661

[9] Tariq, S., Lee, S., Kim, H., Shin, Y. and Woo, S.S., 2019, April. Gan is a friend or foe? a framework to detect various fake face images. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (pp. 1296-1303).

[10] Zhou, P., Han, X., Morariu, V.I. and Davis, L.S., 2018. Learning rich features for image manipulation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1053-1061).

[11] Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J. and Nießner, M. (2018) 'FaceForensics: A large-scale video dataset for forgery detection in

human faces', *arXiv preprint*, arXiv:1803.09179. DOI: 10.48550/arXiv.1803.09179

[12] Li, Y., Yang, X., Sun, P., Qi, H. and Lyu, S. (2019) 'Celeb-DF: A new dataset for DeepFake forensics', *arXiv preprint*, arXiv:1909.12962. DOI: 10.48550/arXiv.1909.12962

[13] Simonyan, K. and Zisserman, A. (2014) 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint*, arXiv:1409.1556, pp. 1–14.

[14] Murali, S., Chittapur, G.B., Anami, B.S. *et al.* (2013) 'Comparison and analysis of photo image forgery detection techniques', *arXiv preprint*, arXiv:1302.3119. DOI: : 10.48550/arXiv.1302.3119

[15] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016) 'Rethinking the inception architecture for computer vision', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826.DOI: 10.1109/CVPR.2016.308

[16] de Paz Centeno, I. (2016) *MTCNN face detection implementation for TensorFlow*. Available at: https://github.com/ipazc/mtcnn (Accessed: 31 December 2020).

[17] Kim, D., Kim, M., Kwon, G. and Kim, D.-S. (2019) 'Progressive face super-resolution via attention to facial landmark', *arXiv preprint*, arXiv:1908.08239. DOI: 10.48550/arXiv.1908.08239

[18] Computational Intelligence and Photography Lab Yonsei University (2019) *Real and fake face detection*. Available at: https://www.kaggle.com/ciplab/real-and-fake-face-detection (Accessed: 31 December 2020).

[19] Hussain, Z., Gimenez, F., Yi, D. and Rubin, D. (2017) 'Differential data augmentation techniques for medical imaging classification tasks', *AMIA Annual Symposium Proceedings*, *American Medical Informatics Association*, p. 979.

[20] Jung, A.B., Wada, K., Crall, J., Tanaka, S., Graving, J., Reinders, C., Yadav, S., Banerjee, J., Vecsei, G., Kraft, A., Rui, Z., Borovec, J., Vallentin, C., Zhydenko, S., Pfeiffer, K., Cook, B., Fernández, I., De Rainville, F.C.-M., Weng, C.-H., Ayala-Acevedo, A., Meudec, R., Laporte, M. *et al.* (2020) *ImgAug*. Available at: https://github.com/aleju/imgaug (Accessed: 31 December 2020).

[21] Chollet, F. (2017) 'Xception: Deep learning with depthwise separable convolutions', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258. DOI: 10.1109/CVPR.2017.195

[22] Venkateswarulu, S. and Srinagesh, A., 2024. DeepExplain: Enhancing DeepFake Detection Through Transparent and Explainable AI model. *Informatica*, *48*(8). DOI: 10.31449/inf.v48i8.5792

[23] Chu, H. (2017) *GitHub: AUROC (Area Under the Receiver Operating Characteristic)*. Available at: https://github.com/hyoungseokchu/AUROC (Accessed: 31 December 2020).

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

Energy Consumption Prediction in Smart Homes Using QIO-Enhanced Regression Models

Yuanjun Zhang
College of Art and Design, Luoyang Vocational College of Science and Technology, Luoyang 471822, China
E-mail: yzdavid09162@163.com

Keywords: smart homes, machine learning schemes, quadratic interpolation optimization

Accurately forecasting household energy consumption remains a challenge due to the variability introduced by user behavior, appliance diversity, and environmental conditions. Smart homes have also become a key solution to the expanding global energy demands in the residential arena through the introduction of new strategies to maximize and control the use of energy. Smart systems comprise sophisticated sensors and networked appliances to provide accurate, affordable, as well as ecofriendly management of energy. Yet, forecasting the use of energy in such homes is challenging with the differences in end-user behavior, weather, and appliance efficiencies. In this work, the use of machine learning (ML) algorithms Extra Tree Regression (ETR), Naive Bayes Regression (NBR), and Elastic Net Regression (ENR) to predict the use of energy in a home is presented. These are also optimized with Quadratic Interpolation Optimization (QIO) to adjust their hyper parameters. Experiments were performed using the Kaggle Smart Home Energy Usage dataset, which provides comprehensive synthetic data across various features of occupancy, appliance usage, temperature, and timestamped consumption data. The developed hybrid schemes were tested using the criteria of R^2 and RMSE in the training, validation, and testing stages. Out of the varied modeled algorithms, the ETQI (ETR + QIO) model realized the highest accuracy of $R^2 = 0.985$ and RMSE = 0.245 while outperforming NBQI (NBR + QIO) with $R^2 = 0.974$ and RMSE = 0.233 and ENQI (ENR + QIO) with $R^2 = 0.952$ and RMSE = 0.319. These conclusions reflect the ability of optimized ML algorithms to drive more effective energy efficiency strategies in the smart home scenario.

Povzetek: Članek predstavi QIO-optimizirane regresijske modele (ETQI, NBQI, ENQI) za napoved porabe energije v pametnih domovih, kjer ETQI doseže najvišjo točnost in stabilnost.

1 Introduction

Due to the unending demand for energy consumption in various parts of the world, especially in the residential sector, efforts towards enhancing energy management have been considered imperative [1]. Integrated homes with multiple devices and sensors are considered a significant step in energy matters [2]. Such dwellings can manage, regulate, and optimize energy intake in unique and valuable ways that preserve the environment while saving the homeowner's money [3]. However, predicting energy use in smart homes has unique difficulties since several variables affect energy use, including appliance efficiency, weather, and user behavior [4]. A smart home consists of two interactive elements: an intelligent network and a smart load comprising sensors, actuators, middleware, and a network [5], [6]. A bright house can focus on three primary goals: maximizing the percentage of home automation, making energy regulation more straightforward, and reducing environmental releases [7]. Today, smart buildings have become a phenomenon as they are overwhelming the world by developing brighter built spaces through the application of leading-edge computational and communication technologies [8]. Public awareness will relate to building automation, where operations are controlled through instruments and microcontrollers in an information exchange process [9]. Secondly, smart buildings utilize integrated systems that forecast energy consumption, hence triggering energy conservation and cost minimization [10]. More significantly, this is crucial given that it is proven that, within developed countries, buildings utilize approximately 40% of primary energy [11].

Energy regulation in the smart home relies on efficient appliances and increasing consumer engagement in demand-side management, energy management systems, and appliance design [12]. Many studies have demonstrated that appliance design and efficiency are paramount; outdated appliances and consumer ignorance of their efficiency are also significant concerns [13]. The authors continue to advocate stricter regulations over appliances, but the responsibility for energy efficiency lies with the consumer. Indeed, several studies have shown a high correlation between awareness of the amount of energy used and the actual consumption. Real-time feedback, through smartphones or fixed-site monitoring equipment, is encouraging, but there is still much interface opacity and technological issues. Energy efficiency extends beyond smart appliances into other areas like space heating, despite possible efficiencies from thermostat positioning. Managing smart home systems requires the customer's awareness and experience. Intertek

defines a smart home as a home with network-connected appliances that are managed remotely [14].

The current research seeks to (1) contrast the accuracy of the three machine learning regression algorithms Extra Tree Regression (ETR), Naive Bayes Regression (NBR), and Elastic Net Regression (ENR) in modeling energy usage in the home, (2) assess the impact of Quadratic Interpolation Optimization (QIO) on the improvement of the accuracy of these models, and (3) ascertain the most significant features responsible for the energy usage trend from the feature importance analysis.

1.1 Related works

This section investigates various research articles on appliances and socio-economic factors concerning energy consumption. Pratt et al. [15] presented data collection of hourly end-use from 140 commercial buildings and 454 residential homes from the Pacific Northwest. They mainly research the development of techniques to predict energy use profiles, with a special interest in refrigeration and freezer loads. A data acquisition system is presented that monitors 12 to 16 channels for energy consumption. Data from UK households was analyzed using a Markov Chain Monte Carlo approach. At the same time, Guo et al. utilized a Hidden Markov model to identify and estimate the energy consumption of individual appliances from aggregated power signals. Other techniques from machine learning have been used in various works: Ling et al. [16], Veit et al. [17], and Arghira et al. [18] analyzed neural networks (NNs), forecasting methods, and multiple regression techniques. Bharati et al. applied KNN, SVM, stochastic gradient descent, random forests, and neural networks for the classification attributes of the energy dataset [19]. They also utilized neuro-fuzzy inference for superconductor dataset feature prediction. Scott et al. [20] deployed a NARX Gaussian Process regression scheme and a linear regression scheme to predict energy loads from UK data, attaining a low prediction error of 0.26% [1], [2].

Other research filtered data from 1,628 households for electricity consumption, using the following criteria as a basis: weather, location, age of building, ownership, energy-efficient fixtures, and even income class [3], [4]. The analysis determined that weather, location, and floor area were the most significant factors in energy consumption. Nevertheless, daytime occupancy of the home was related to reduced appliance efficiency [21]. Overall, all literature underlines these different elements as an understanding of patterns in energy usage. Recent studies have introduced advanced forecasting and control models for smart energy systems. Razghandi et al. proposed a Seq2Seq model integrated with Q-learning for predictive control in smart homes [5]. Pan applied tree-based models optimized with War SO for accurate solar energy forecasting [6]. Nakıp et al. developed an rTPNN-based framework for joint forecasting and appliance scheduling in smart environments, achieving near-optimal scheduling and high predictive performance [7].

Table 1 gives a comparative overview of the significant studies in the area of predicting smart home energy usage. Comparison of the studies is in terms of the datasets, the machine learning algorithms used, the adopted feature engineering techniques, and model performances as measured by RMSE and R^2. The comparison highlights the novelty and strength of the adopted approach in this research, Quadratic Interpolation Optimization (QIO), in its ability to improve model performances in energy forecasting tasks.

Table 1: Comparative summary of key studies

Study	Dataset	Algorithm	Feature Engineering	Results (RMSE, R^2)
Pratt et al. [15]	ELCAP, 140 buildings, 454 homes	Custom Regression Techniques	Manual Load Profiling	RMSE: N/A, R^2: N/A
Guo et al. [8]	UK Aggregated Load	Hidden Markov Model	Appliance-level Disaggregation	RMSE: ~0.32, R^2: ~0.91
Ling et al. [16]	Synthetic Short-term Load	Neural Networks	Fuzzy Feature Selection	RMSE: 0.29, R^2: 0.94
Bharati et al. [19]	FICTA Dataset	KNN, SVM, RF	Plot Analysis	RMSE: ~0.27, R^2: ~0.92
Scott et al. [20]	UK Smart Meter	NARX + Gaussian Process	None Specified	RMSE: 0.26, R^2: 0.95
Current Study	Kaggle Smart Home	ETR, NBR, ENR + QIO	Quadratic Interpolation Optimization	ETQI: RMSE: 0.245, R^2: 0.985

1.2 Objectives

Tree-based machine learning approaches were used in preference to deep learning solutions because of their reduced training time, improved interpretability, and good results on structured, table-like datasets such as the Smart Home Energy dataset. They use less data preprocessing and computational power as opposed to neural networks and are more apt for real-world, real-time applications in smart homes. This paper discusses a few Machines

Learning (ML) schemes that can be used for energy usage prediction in smart homes, mainly focusing on Extra Tree Regression (ETR), Naive Bayes Regression (NBR), and Elastic Net Regression (ENR). Each of these schemes has unique advantages when it comes to the estimation of energy consumption data. ETR has been widely

recognized for its efficiency in handling a large volume of data and its robustness to outliers, especially in the case of complex energy-use consumption patterns. In contrast to the above, NBR relies on a probability-based method that correctly addresses the uncertainty in the data, thereby leading to better prediction accuracy.

First, ENR embodies the characteristics of both Lasso and Ridge regressions and has stronger robustness for high correlations between features. Second, Quadratic Interpolation Optimization (QIO) is applied to improve the predictiveness of these schemes further. The QIO optimizer is essential in hyperparameter optimization, enabling it to find the best-fitted parameters for each model. The result is a holistic approach that leads to the development of predictive schemes with better performance, thus providing the most valuable insight into energy consumption efficiency in smart homes. This paper contributes to further understanding and optimally steering energy management within residential settings with the help of advanced ML techniques.

The remainder of this paper is organized as follows: Section 2 provides a detailed description of the dataset and preprocessing techniques. Section 3 explores the influence of occupancy status on energy consumption in smart homes. Section 4 presents the mathematical schemes, including the machine learning models and the QIO optimization process. Section 5 discusses the experimental results and model evaluations. Finally, Section 6 concludes the study and outlines future research directions.

2 Data description

2.1 Data preprocessing

Before model training, various steps in the preprocessing of the data were performed. Missing continuous variable values were replaced using mean substitution, but rows with missing categorical values were deleted as they were of low frequency. Continuous features like temperature and energy consumption were normalized with min–max scaling to have uniform contribution to the model during training. The day-of-week and season variables were encoded with one-hot encoding in order to maintain model consistency with non-numerical data. Timestamp data was also processed to its day, month, and hour components to enable more effective capturing of temporal trends in the energy behavior. The data was partitioned randomly to 70% for training, 15% for validation, and 15% for testing, with the aim of having a representative feature value distribution across phases. Five independent trials were conducted to ensure consistency of results.

Temporal characteristics were analyzed to augment their prediction value. The raw timestamp was analyzed to extract hour, day, and month numbers to identify periodic trends in energy use. The categorical characteristics "day of week" and "season" were encoded using one-hot encoding to preserve non-ordinal connections among the categories. This encoding enabled the models to leverage temporal context without creating erroneous hierarchies. No embedding layers were utilized, as the models are tree-based and proficiently manage one-hot encoded features.

2.2 Data description

All these details were obtained from the Smart Home Energy Usage Dataset from Kaggle, as noted at this link. The given dataset contains synthetic data, essential for further research on different scenarios related to the energy optimization subject within smart homes. Thus, the availability of data regarding electricity consumption, appliance usage, temperature settings, and occupancy status is critical for researchers and developers who want to enhance contemporary dwellings' energy efficiency and home automation. It is formatted in a manner that allows the detection of relevant consumption data, further facilitating the identification of potential energy-saving issues and optimization of home energy consumption to support automation. This dataset is useful in further research to analyze energy intensities to improve smart home operations and other automated processes that aim to improve home energy efficiency.

Timestamp: This column specifies the precise day and time the data was collected to allow comparison of customers' purchasing patterns at various time intervals.

Home-id: A unique number assigned to each house, allowing for comparisons and evaluations of the energy efficiency of various residences.

Energy-consumption-kWh: This column's first parameter, which allows for evaluating total energy consumption, is the energy usage measured in kilowatt-hours (kWh).

Temperature-setting-C: This feature displays the houses inside temperature in degrees Celsius, which is highly useful for estimating the energy a heater or air conditioner would use in a certain household.

Occupancy status: Indicates whether the house was occupied or unoccupied when the data was gathered, allowing the impact of presence on energy usage to be shown.

Appliance: It classifies the kind of appliance being used, including air conditioning units, washing machines, and dishwashers, helping to determine each appliance's contribution to the overall amount of energy used.

Usage-duration-minutes: The number of minutes an appliance was on helps distinguish between the quantity of energy used and the period the device was in use.

Season: The year when the data was collected since these aids in analyzing the patterns of energy consumption associated with each season, for instance, winter, spring, summer, or fall.

Day-of-week: This variable establishes the day of the week to differentiate daily energy use.

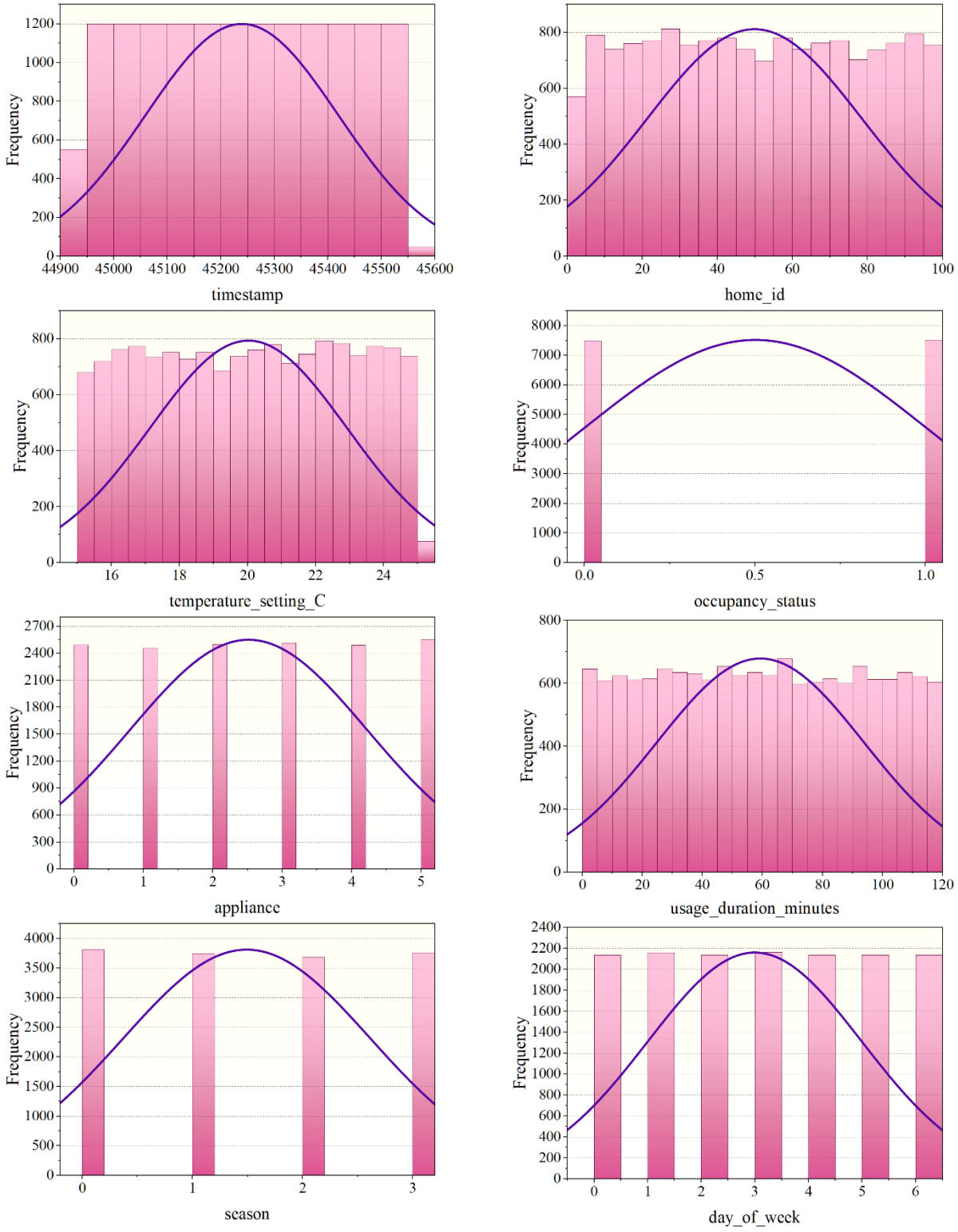
Holiday: A flag variable that, to filter out days that could vary due to the holiday, is equal to 1 if the day was a holiday and zero otherwise.

Therefore, it can be used in trend analysis, forecasting overall energy usage, enhancing intelligent automation at home, and, more importantly, developing automated energy-saving solutions. It allows for deeper and more

practical analysis of energy use in smart homes, thus making an environment even more effective and efficient.

The use of a synthetic dataset offers controlled conditions for model development and evaluation; however, it also presents limitations in terms of real-world applicability. Synthetic data may not capture the full complexity, variability, and noise present in actual household energy consumption patterns. Consequently, model performance observed in this study may not fully generalize to real-world deployments. Future work will incorporate real datasets, such as UK-DALE or REDD, to evaluate model robustness under realistic operational conditions and validate applicability in practical smart home environments.

Fig.1 presents the relationship between measured and predicted values, showing the model's capability to detect patterns and accuracy in prediction.



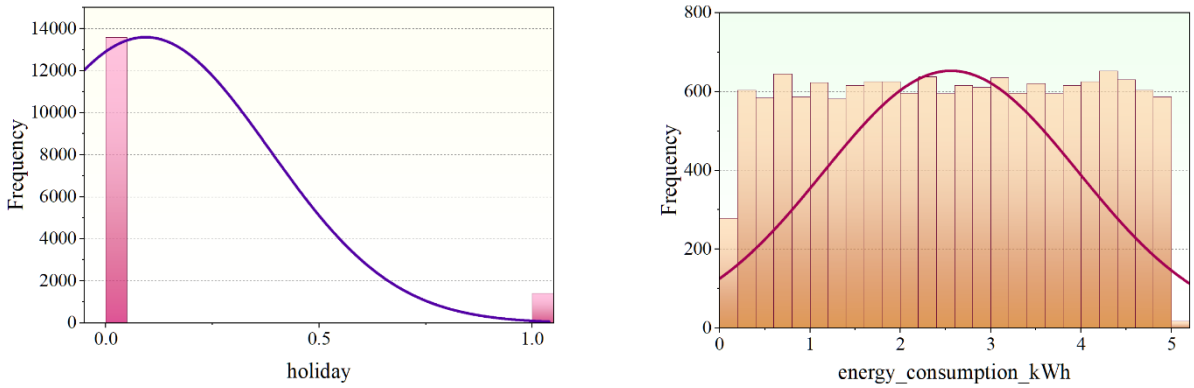


Figure 1: The links between input and output variables assessed utilizing a marginal histogram plot

3 Exploring the influence of occupancy status on energy consumption in smart homes

Energy consumption in smart homes is closely related to the occupancy status range, and it is primarily associated with HVAC. For instance, in the U.S., HVAC accounts for 43% of residential energy consumption, while it increases to 61% in colder regions like Canada and the U.K. Energy savings of up to 20-30% have also been estimated by switching these systems off. At the same time, the residents are asleep or away. However, these savings have often been hard to realize. Many residents do not usually change their thermostats, while programmable thermostats frequently result in higher energy use due to incorrect settings or user disinterest [22]. Occupancy-based control strategies have gained attention for improving energy efficiency by aligning HVAC operations with real-time user presence. Studies show that dynamic occupancy detection using sensors or predictive models can significantly reduce unnecessary energy consumption. Integrating occupancy data into energy prediction models enhances forecasting accuracy and enables proactive control, which is essential for smart home automation systems aiming to balance comfort, cost, and environmental sustainability.

4 Mathematical schemes

All experiments were implemented in Python version 3.13.3 using Visual Studio Code as the development environment. The machine learning models were developed using the scikit-learn library, which provided tools for model training, evaluation, and cross-validation.

The average runtime for base models was approximately 1.4 seconds for NBR, 3.1 seconds for ENR, and 4.2 seconds for ETR. When optimized with QIO, runtimes increased to 48.6 seconds for NBQI, 76.4 seconds for ENQI, and 101.8 seconds for ETQI. This added time reflects QIO's iterative tuning process, which enhances model performance and generalization.

4.1 Extra tree regression (ETR)

Geurts et al. [23] first introduced ETR, initially inspired by the Random Forest (RF) scheme. The ETR tactic generates regression trees, or sets of unpruned choices, in line with the conventional top-down methodology [23]. The RF scheme does the regression in two steps: bagging and bootstrapping. During the bootstrapping process, each tree develops to generate a collection of DTs using a sample from a stochastic training dataset. Upon arriving at the ensemble, the decision tree (DT) nodes undergo division by implementing a two-step bagging process. The first stage involves the selection of various random subsets of training data. The decision-making procedure is concluded by choosing the ideal subset and its value [24]. Breiman [25] thought of the RF scheme as being composed of several DTs where $G(x,\theta_r)$ displays the G^{th} the unified individual distribution pattern assigned before the tree grows, denoted by θ in the prediction tree. Breiman is used to average and combine all of the trees, resulting in an ensemble of trees (a forest) of G(x). The procedure is shown in Eq (1):

$$G(x,\theta_1,\dots,\theta_r)=\frac{1}{R}\sum_{r=1}^{R}G(x,\theta_r) \quad (1)$$

There are two basic differentiations between RF and ETR. By randomly choosing a subset of all the cutting locations, the ETR separates nodes in the first place. It also cultivates trees by utilizing all of the learning samples to lessen bias. Two variables in the ETR technique control the splitting process: k and n_{min} .where the n_{min} Variable indicates the least sample size needed to discriminate across nodes, and k r displays the number of characteristics the node randomly picks. Furthermore, k and n_{min}, respectively, determine the medium result noise intensity and the feature extraction strength. These two elements reduce overfitting and boost the accuracy of the ETR scheme [26], [27].

4.2 Naive Bayes Regression

The example below looks at the challenge of anticipating a numerical goal value Y. Qualities in E $(X_1, X_2,\dots,X_m)$. are m. Features may be actual, quantitative, nominal, or a collection of unordered values [28]. To lower the predicted prediction error, Y might be selected if the

probability density function $f(Y \mid E)$ of the target value was known. However, $f(Y \mid E)$ is usually unknown and necessitates data estimates. It is expected that the features. $X_1, X_2, \ldots, X_m$ are independent of the goal value Y when Naive Bayes uses the Bayes theorem.

$$f(Y \mid E) = \frac{f(E,Y)}{\int f(E,Y)dY} = \frac{f(E \mid Y)f(Y)}{\int f(E \mid Y)f(Y)dY} \quad (2)$$

The Probability Density Function (PDF) of example E for a particular target value Y is known as the like hood, or $f(E \mid Y)$, the pdf of the goal value before any examples are seen is known as the prior f(Y). Naive Bayes is represented in terms of attribute independence by Eq (2).

$$f(Y \mid E) = \frac{f(X_1 \mid Y)f(X_2 \mid Y) \cdots f(X_m \mid Y)f(Y)}{\int f(X_1 \mid Y)f(X_2 \mid Y) \cdots f(X_m \mid Y)f(Y)\,dY} \quad (3)$$

It is now possible to compute the individual PDFs $f(X_i \mid Y)$. instead of computing the overall pdf $f(E \mid Y)$. This reduction in dimensionality makes the learning task much more manageable. Since more data are needed to predict accurately $f(X_i \mid Y)$, this method is more trustworthy than estimating $f(E \mid Y)$.

4.3 Elastic Net Regression (ENR)

It is a high-performance linear regression technique that combines the best features of L1 (Lasso) and L2 (Ridge) regularization strategies. Its dual regularization strategy reduces multicollinearity and overfitting in high-dimensional datasets, encourages sparsity and stability in coefficient estimates, and improves the predictive accuracy and interpretability of the model [29], [30].

4.3.1 Direct representation

$P(y|\beta,\sigma^2) = N(y|X\beta,\sigma^2 I_n)$ This is the probability for this article, where β is a p-vector that contains the regression coefficients. Where X is a matrix of predictor variables with $n \times p$ dimensions. Since it is assumed that the vector y and the columns of X are demeaned, the model does not contain an intercept [31], [32]. In this approach, estimations of the linear regression parameters are usually

$$\hat{\beta} = \underset{\beta}{arg\,min}\,(y - X\beta)^T(y - X\beta) + \lambda J(\beta) \quad (4)$$

Taking into account a regularization parameter $\lambda > 0$ and a nonnegative punishment function J.

$$p(\beta|\lambda,\alpha) \propto exp[-\lambda\,\{\alpha|\beta|^2 + (1-\alpha)|\beta|_1\}], \quad (5)$$

This study extends the Bayesian connection to the ELR procedure by providing a properly normalized and explicated version of the prior.

$$p(\beta|\alpha,\lambda,\sigma^2) \propto exp\left[-\frac{\lambda}{2\sigma^2}\,\{\alpha|\beta|^2 + (1-\alpha)|\beta|_1\}\right], \quad (6)$$

As a result, the penalty λ now has a size of $2\sigma^2$. This formulation states that for given values of σ^2 and α, the posterior mode will be the naïve elastic net estimate with an overall penalty of λ. This prior is a double-exponential distribution when $\alpha = 0$. When $\alpha \approx 1$, it has the properties of a normal distribution. The integration of Eq. (6) displays that the normalizing constant is available in closed form until the univariate standard normal cdf is assessed. The appropriate reduced-scale density function from the previous

$$p(\beta|\lambda,\alpha,\sigma^2) = \prod_{j=1}^{p}\left\{(0.5).N^-\left(\beta_j|\frac{1-\alpha}{2\alpha},\frac{\sigma^2}{\lambda\alpha}\right) + (0.5).N^+\left(\beta_j|-\frac{1-\alpha}{2\alpha},\frac{\sigma^2}{\lambda\alpha}\right)\right\}, \quad (7)$$

In truncated normal distributions, $N^-$and N^+ are appropriately adjusted density functions.

$$N^+(t|m,s^2) \equiv \frac{N(t|m,s^2)}{\phi(m/s)}1(t \geq 0)\ And\ \ N^-(t|m,s^2) \equiv \frac{N(t|m,s^2)}{\phi(-m/s)}1(t < 0), \quad (8)$$

A normal distribution's tails will always contain the univariate standard normal CDF and $\phi.\beta_j$ since the location parameter for the positive component in Eq. (7) is always negative.

Here's another interpretation of the preceding: $et\ Z = \{-1,1\}^p$ be the set of all p-vectors that can have members ±1, and let $\mathcal{O}_z \subset \mathbb{R}^p$ be the orthant that corresponds to each vector z in $Z.\beta j \geq 0$ for $\mathcal{Z}_j = 1$ and $\beta j < 0$ for $\mathcal{Z}_j = -1$ if $\beta \in O_z$. The previous Eq. (6) may thus be rewritten as

$$p(\beta|\lambda,\alpha,\sigma^2) = 2^{-p}\phi\left(\frac{\alpha-1}{2\sigma\sqrt{\alpha/\lambda}}\right)^{-p} \times \sum_{z\in Z} N\left(\beta|\frac{\alpha-1}{2\alpha}z,\frac{\sigma^2}{\lambda\alpha}I_p\right)1(\beta \in Oz). \quad (9)$$

An "orthant normal" prior is produced by describing each piece over a distinct orthant, proving that the prior is piecewise normal. The results are as follows when the prior is stated in terms of λ_1 and λ_2.

$$p(\beta|\lambda_1,\lambda_2,\sigma^2) = 2^{-p}\phi\left(\frac{-\lambda_1}{2\sigma\sqrt{\lambda_2}}\right)^{-p} \times \sum_{z\in Z} N\left(\beta|-\frac{\lambda_1}{2\lambda_2}z,\frac{\sigma^2}{\lambda_2}I_p\right)1(\beta \in Oz). \quad (10)$$

From now on, the (λ_1,λ_2) unless otherwise noted, the formulation is used. The posterior distribution is obtained by multiplying the probability of the regression scheme by Eq. (10) and using the Bayes theorem.

$$p(\beta|y,\lambda_1,\lambda_2,\sigma^2) = \sum_{z\in Z}\omega_z N^{[z]}(\beta|\mu_z,\sigma^2 R), \quad (11)$$

Following the trimming of the 2^porthant, the weighted sum of the normal distributions

$$N^{[z]}(\beta|m,s) \equiv \frac{N(\beta|m,s)}{P(z,m,s)}1(\beta \in Oz), \quad where \quad P(z,m,s) = \int_{\mathcal{O}_z} N(t|m,s)dt, \tag{12}$$

Symbolizes a multivariate standard orthant integral. Since each component is specified on a different orthant, the posterior and the prior are multivariate piecewise normal. Its posterior distribution is its set of parameters.

$$R = \left(X^T X + \lambda_2 I_p\right)^{-1} \quad and \quad \mu_z = \hat{\beta}_R - ,\frac{\lambda_1}{2} Rz, \tag{13}$$

The ridge regression estimate in this instance is represented as $\hat{\beta}_R = RX^T y$, with a penalty of λ_2. The weights for every orthostat make up the last components of the posterior.

$$\omega_z = \omega^{-1}\frac{P(z.\mu_z,\sigma^2 R)}{N(0|\mu_z,\sigma^2 R)}, \quad where \ \omega = \sum_{z\in Z}\frac{P(z,.\mu_z,\sigma^2 R)}{N(0|\mu_z,\sigma^2 R)}. \tag{14}$$

4.3.2 Representation of mixtures

Several studies employing Bayesian modeling have used scale mixes of normal distributions. One can obtain the orthant normal distribution by merging several scale mixes of various normal distributions.

$$P(\beta|\sigma^2,\ \lambda_1\ ,\lambda_2) = \prod_{j=1}^{p}\int_0^1 N\left(\beta_j|0,\frac{\sigma^2}{\lambda_2}(1-\tau_j)\right) \times IG_{(0,1)}\left(\tau_j|\frac{1}{2},\frac{1}{2}\left(\frac{\lambda_1}{2\sigma\sqrt{\lambda_2}}\right)^2\right)d\tau_j, \tag{15}$$

In such case, the only viable place for the inverse gamma distribution is $(IG_{(0,1)})$. There is a proof in the appendix. By adding latent variables $\tau_1,\dots,\tau_p$ and deploying the notation $S_\tau = \mathrm{diag}(1-\tau_j)$, the prior may be stated hierarchically below:

$$p(\beta|\tau,\sigma^2,\ \lambda_2) = N\left(\beta|0,\frac{\sigma^2}{\lambda_2}S_\tau\right), p(\beta|\sigma^2,\lambda_1,\lambda_2) = \prod_{j=1}^{p} IG_{(0,1)}\left(\tau_j|\frac{1}{2},\frac{1}{2}\left(\frac{\lambda_1}{2\sigma\sqrt{\lambda_2}}\right)^2\right) = 2^{-P}\phi\left(\frac{-\lambda_1}{2\sigma\sqrt{\lambda_2}}\right)^{-P}\left(\frac{\lambda_2}{8\pi\sigma^2\lambda_2}\right)^{P/2} \times exp\left(\frac{\lambda_1^2\sum_{j=1}^{p}\tau_j^{-1}}{8\sigma^2\lambda_2}\right) \times \prod_{j=1}^{p}\tau_j^{-3/2}1(0<\tau j<1). \tag{16}$$

Because $P(\beta|\sigma^2,\ \lambda_1\ ,\lambda_2)$ is a product of separate, unconstrained normal dispersion, the hierarchical representation is advantageous. The multivariate normal conditional posterior dispersion is thus given by $p(\beta|y,\tau,\sigma^2,\lambda_1,\lambda_2) = N\left(\beta|\hat{\beta}_{R_\tau},\sigma^2 R_\tau\right)$, where $R_\tau = (X^T X + \lambda_2 S_\tau^{-1})^{-1}$, and the mean vector is the ridge-like estimate $\hat{\beta}_{R_\tau} = R_\tau X^T y$.

4.3.3 Relationship with the previous double-exponential

According to the research, the elastic net method is defined as follows: A continuum of penalties is produced by the penalty term $P(\beta|\sigma^2,\ \lambda_1\ ,\lambda_2)$when there is ridge regression $(\alpha = 1)$ and a lasso $(\alpha = 0)$ at both extremes. In other words, a continuum of priors between a double-exponential prior $(\alpha = 0)$ and a typical prior $(\alpha = 1)$ is indexed by the orthant normal prior. The relationship between regression schemeing and the lasso under the double-exponential prior has received more attention recently. Applying the scale mixing of regular representation, it makes sense to compare the double-exponential before the ELR prior. A scaled penalty term may be used to derive the double-exponential prior under the restriction that $-(\lambda_2/2)\beta_j^2 - (\lambda_1/2)|\beta_j|$. This matches a prior when $\sigma^2 = 1$.

$$P(\beta_j|\lambda_1) = \frac{\lambda_1/2}{2}e^{-\left(\lambda_1/2\right)|\beta_j|}. \tag{17}$$

Scale mixing may be used to depict this antecedent using the findings of Andrews and Mallows (1974).

$$\beta_j|\theta_j = \sim N(0,\theta_j^2), \quad P(\beta_j|\lambda_1) = \frac{\lambda_1^2}{4}\theta_j e^{-\lambda_1^2\theta_j^2/8}, \quad \theta_j > 0, \tag{18}$$

As an example, consider $\theta_j^2 \sim \mathrm{Exp}\left(\frac{\lambda_1^2}{8}\right)$ or, conversely, $\theta_j \sim \mathrm{Weibull}\left(2,\frac{\lambda_1^2}{8}\right)$. The proper transformation may also denote the scale-mixture representation of the orthant typical prior.

$$\beta_j|\theta_j = \sim N(0,\theta_j^2), \quad P(\theta_j|\lambda_1,\lambda_2) = \frac{\lambda_1\theta_j\left(\lambda_2^{-1}-\theta_j^2\right)^{-\frac{3}{2}}}{\lambda_2\sqrt{8\pi}\,\phi\left(-\frac{\lambda_1}{\left(2\sqrt{\lambda_2}\right)}\right)} \times exp\left\{-\frac{\lambda_1^2\left(\lambda_2^{-1}-\theta_j^2\right)^{-1}}{8\lambda_2^2}\right\}, \quad 0<\theta_j<1/\sqrt{\lambda_2}. \tag{19}$$

Significant shrinkage is correlated with small values of θ_j. Increasing the value of λ_1 causes more shrinkage in both (17) and (18) as the mixing distribution gradually shifts in favor of lower values of θ. Adding option λ_2 to (18) also results in more shrinking, but the final shrinkage is different from what would happen if λ_1 were to be lowered alone. Even while the distribution of θ tends to become more concentrated at the upper border of the support of θ at the nonzero value $\lambda_2^{-\frac{1}{2}}$, it still prefers smaller values of θ as $\lambda_2$grows.

4.4 Quadratic interpolation optimization (QIO)

QIO [9] was developed to handle problems related to continuous optimization. This process was modeled by the generalized quadratic interpolation (GQI) approach, which is used as a searching mechanism by the QIO algorithm to solve many types of optimization issues. The following two sections detail the algorithm's two phases, exploration and exploitation, comparable to those of the other metaheuristic algorithms.

4.4.1 Method of exploration

The QIO algorithm uses this strategy to preserve population diversity and avoid local minima. The following formulas show why this method updates each population solution using the GQI method.

$$\vec{v}_i^{t+1} = \vec{x}_I^*(t) + w_1 \cdot \left(\overrightarrow{x_{r3}} - \overrightarrow{x_{rI}^*}(t)\right) + round\ (0.5.\ (0.05 + r_1)) \cdot log\ \frac{r_2}{r_3} \quad (20)$$

Where the values in each of the three variables, r_1 , r_2, and r_3, are produced at random and range from 0 to 1. As the current population's randomly chosen solution, $\overrightarrow{x_{r3}}$, the GQI function predicts $(\overrightarrow{x_{rI}^*}(t)$ According to the following formula:

$$\vec{x}_I^*(t) = GOI(\vec{x}_I^t, x_{r1}, x_{r2}, f(\overrightarrow{x_{r1}})), f(\overrightarrow{x_{r2}})) \quad (21)$$

Here $\vec{x}_I^t$ is the ith solution, $f(\cdot)$ is the fitness function, and $\overrightarrow{x_{r1}}$ and $\overrightarrow{x_{r2}}$ Are random solutions selected from the current population. In this Eq, GQI is the GQI function. It may be quantitatively generated for w_1 with the following formula:

$$w_1 = 3_{n1}b \quad (22)$$

$$b = 0.7 \cdot a + 0.15 \cdot a \cdot \left(\cos\left(\frac{5\pi t}{T_{max}}\right) + 1\right) \quad (23)$$

$$a = cos\left(\frac{\pi t}{2T_{max}}\right) \quad (24)$$

Where n_1 is a random number drawn from a normal dispersion, t is the current function evaluation, and T_{max} is the maximum function evaluation.

4.4.2 Exploitation Strategy

By carrying out the exploitation operator using the GQI approach in the regions surrounding the best-so-far solution, the QIO algorithm quickens the convergence of the near-optimal solution. The exploitation capabilities of the QIO algorithm are applied using the following formula for each solution in the population:

$$\vec{v}_i^{t+1} = \overrightarrow{x_{best}^*}(t) + w_2 \cdot \left(\overrightarrow{x_{best}} - round(1 + r_4) \cdot \frac{(U - L)}{U_{rD} - L_{rD}} \cdot \vec{x}_{i,rD}^t\right) \quad (25)$$

The formula w_2 is produced as follows: r_4 displays a random number between 0 and 1, U displays the 1 upper limit, and L displays the lower limit. $\vec{x}_{i,rD}^t$ is a vector consisting of random dimensions picked 2 from the ith solution, where the best-so-far answer is denoted by $\overrightarrow{x_{best}}$.

$$w_2 = 3 \cdot \left(1 - \frac{t-1}{T_{max}}\right) n_2 \quad (26)$$

Where n_2 is a standard distribution-based random number. The GQI function calculates it for $\overrightarrow{x_{best}^*}(t)$, as shown by the following formula:

$$\overrightarrow{x_{best}^*}(t) = GOI\left(\overrightarrow{x_{best}}, \overrightarrow{x_{r1}}, \overrightarrow{x_{r2}}, f(\overrightarrow{x_{best}}), f(\overrightarrow{x_{r1}}), f(\overrightarrow{x_{r2}})\right) \quad (27)$$

QIO acts as a hyperparameter optimizer for every base regression model in the suggested hybrid models. Extra Tree Regression optimizes factors like minimum samples per leaf and the number of estimators. In Naive Bayes Regression, smoothing parameters are changed, but in Elastic Net Regression, the mixing ratio and penalty severity are tweaked. During cross-validation, QIO uses its exploration and exploitation algorithms to repeatedly adjust these hyperparameters with the goal of lowering RMSE. The hyperparameter set that is being tuned for each individual model is represented by the "solution."

In contrast to grid search and Bayesian optimization, QIO provides expedited convergence in low-dimensional hyperparameter spaces by effectively balancing exploration and exploitation phases. Whereas grid search systematically assesses all possibilities and Bayesian optimization necessitates statistical modeling, QIO employs fewer evaluations to attain near-optimal answers. This study demonstrated that QIO attained convergence within acceptable timeframes (60–120 seconds), rendering it appropriate for lightweight systems with constrained processing resources.

Fig. 2 represents the QIO flowchart.

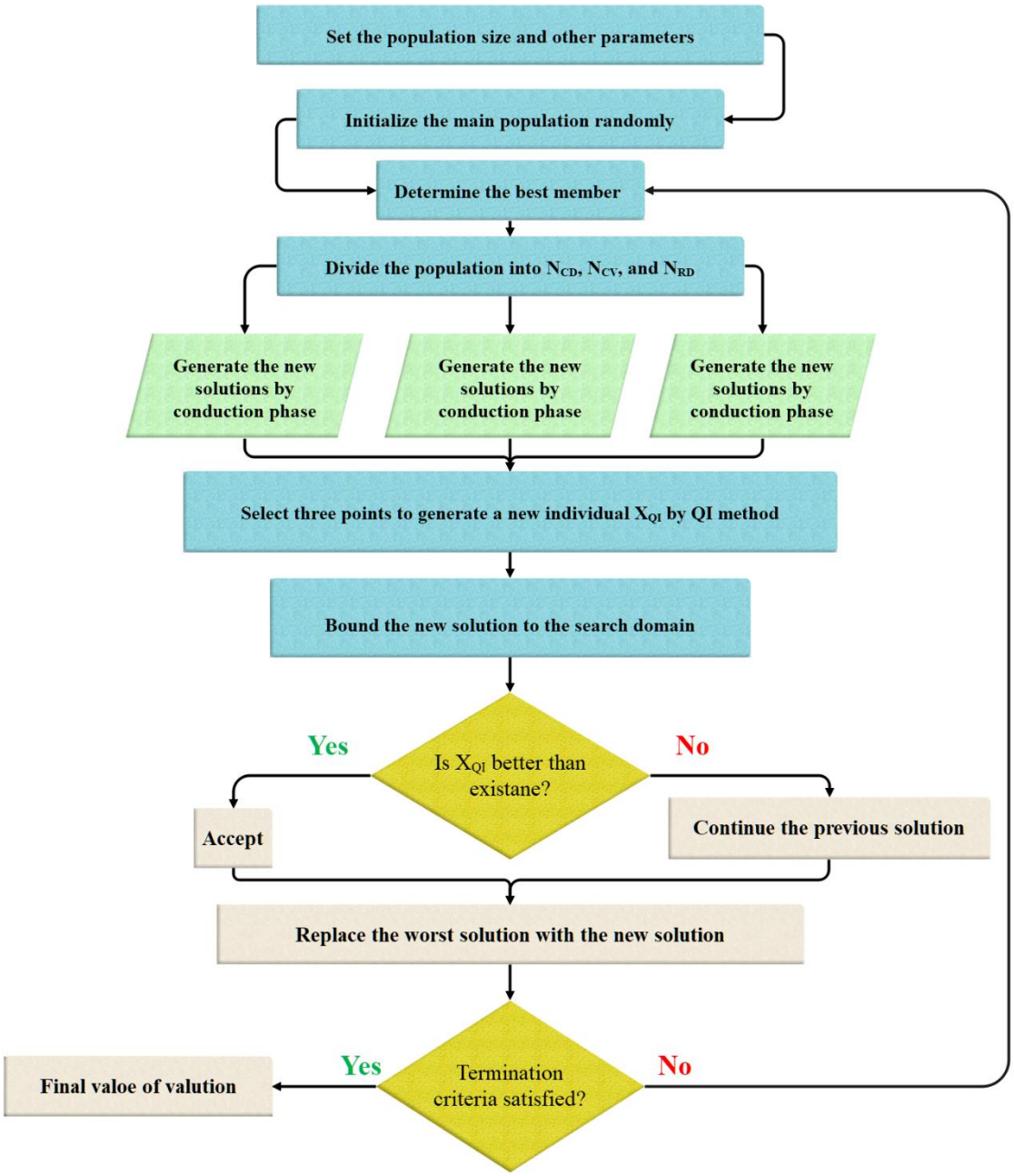


Figure 2: QIO flowchart.

4.5 Performance evaluators

This section represents a systematic framework for deploying hybrid schemes, along with an overview of the various assessment criteria concerning the accuracy and correlation of the schemes. The following mathematical formulae define the metrics for evaluation:

$$R^2 = \left(\frac{\sum_{i=1}^{n} (b_i - \bar{b})(m_i - \bar{m})}{\sqrt{\left[\sum_{i=1}^{n} (b_i - \bar{b})^2\right] \left[\sum_{i=1}^{n} (m_i - \bar{m})^2\right]}} \right)^2 \quad (28)$$

Coefficient of Determination

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (m_i - b_i)^2} \quad (29)$$

Root Mean Square Error

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (P_i - T_i)^2 \quad (30)$$

Mean Squared Error

$$MARE = \frac{1}{r} \sum_{i=1}^{r} \frac{|P_i - T_i|}{T_i} \quad (31)$$

Mean Absolute Relative Error

$$n20_index = \frac{m20}{M} \quad (32)$$

Normalized Mean Error Index

n signifies sample size in Eqs. (28-32); b_i signifies forecasted value, while p_i signifies every single measurement in the sample, and T_i Signifies actual value. m_i is the measured value, and mean predicted and measured values are represented as $\bar{b}$ and $\bar{m}$, respectively.

❖ **Hyper parameter**

Table 2 delineates the principal hyperparameters employed for the basic models (NBR, EN, ETR) and their respective QIO-optimized variants (NBQI, ENQI, ETQI). The number of restarts for the optimizer in the Naive Bayes models was substantially raised from 0 to 71, hence improving convergence stability. The Elastic Net models vary in their regularization settings. The optimization of QIO markedly decreased the L1 ratio, signifying a heightened focus on ridge-like regularization. In the Extra Tree models, QIO adjusted the number of estimators from 100 to 11 and raised the minimum number of samples per leaf from 1 to 4, enhancing generalization. These modifications demonstrate the function of QIO in optimizing hyperparameter settings to improve model efficacy in prediction tasks.

Table 2: Hyper parameters of hybrid and base models.

Models	n_restarts_optimizer	alpha	l1_ratio	n_estimators	min_samples_leaf
NBR	0				
NBQI	71				
EN		1	0.5		
ENQI		3	0.0108		
ETR				100	1
ETQI				11	4

5 Result and discussion

This section presents the results of the analyses done to assess smart home energy usage, focusing on the performance of different ML schemes in forecasting energy consumption.

- **Convergence**

Fig. 3 displays the performance comparison for each model across different iterations. As can be seen, for instance, the NBQI model had an initial value of RMSE of 0.45, which decreased to 0.201 after 163 iterations, showing that this model takes much time to increase its accuracy. Similarly, ENQI starts with an approximate value of RMSE 0.55, then decreases after 104 iterations to 0.296, showing the performance improvement advantage. It was initiated at an RMSE value of 0.40 and managed to reduce it to 0.173 after 156 iterations. Therefore, ETQI is the best for smart home energy usage prediction.

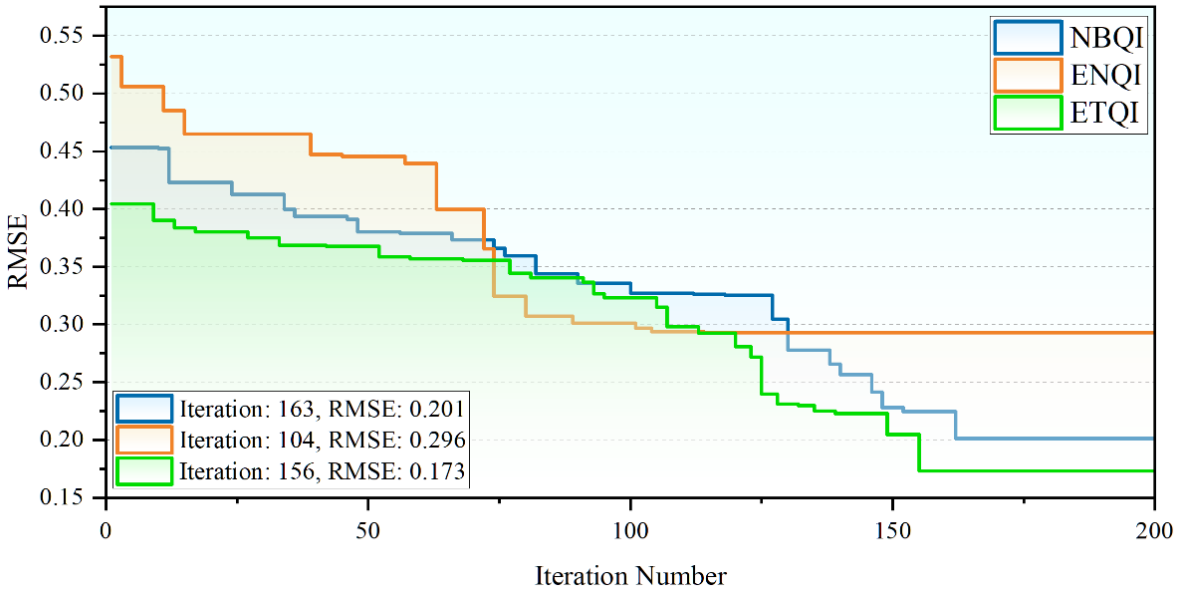


Figure 3: The convergence curve of the three hybrid schemes is presented

- **Comparison of predictive schemes for energy usage**

Table 3 displays the model performances based on five metrics during the training, validation, and testing steps. The R^2 values of the schemes NBR, NBQI, ENR, ENQI, ETR, and ETQI were estimated to be 0.968, 0.980, 0.949, 0.959, 0.976, and 0.985, respectively, in the modeling process. At this phase, ETQI performed better than any other schemes in sequence by NBQI schemes. Comparing the results of NBQI and NBR in the testing phase, NBQI outperforms with R^2 values 0.974 and 0.962, while their corresponding RMSE values are 0.233 and 0.280. Similarly, ENQI performs better from the two variants, ENR and ENQI. It achieves an R^2 of 0.952 and

an RMSE of 0.319 for ENQI, while the R^2 value for ENR is 0.942 with 0.349 RMSE. Among all, ETR yields an R^2 of 0.956 and an RMSE of 0.302, while ETQI reaches an R^2 value of 0.971 and an RMSE of 0.245 in the test phase. ETQI consistently achieved the lowest MAE and highest AUC in all phases, indicating superior precision and classification confidence. NBQI also performed competitively, especially in the validation and test phases. ENR and ENQI exhibited the weakest performance, with noticeably higher MAE and slightly lower AUC compared to tree-based schemes.

The experimental results clearly demonstrate the effectiveness of QIO-enhanced models in improving prediction accuracy. As shown in Table 3, ETQI consistently achieved the lowest RMSE and highest R^2 across all phases, indicating superior generalization. The performance gap between base and optimized models highlights the impact of QIO in fine-tuning hyperparameters. Figs. 4 to 6 further confirm the robustness and consistency of ETQI and NBQI, with lower dispersion and tighter clustering around the median error values.

Fig. 3 presents the performance of the schemes during all phases combined in a chart. It is seen from this figure that ETQI has performed the best; its R^2 is higher, and its RMSE is lower than those of the other schemes. NBQI has ranked second best in performance. Conversely, ENR has been ranked the poorest model.

Table 3: The result of developed schemes for NBR, ENR, and ETR

Phase	Model	Index values						
		R^2	RMSE	MSE	MARE	N20_index	MAE	AUC
Train	**NBR**	0.968	0.258	0.067	0.069	0.960	0.173	0.970
	NBQI	0.980	0.201	0.041	0.054	0.994	0.135	0.974
	ENR	0.949	0.326	0.106	0.087	0.906	0.219	0.961
	ENQI	0.959	0.293	0.086	0.079	0.931	0.197	0.965
	ETR	0.976	0.222	0.049	0.060	0.986	0.149	0.971
	ETQI	0.985	0.173	0.030	0.047	0.997	0.117	0.976
Validation	**NBR**	0.958	0.295	0.087	0.078	0.921	0.191	0.967
	NBQI	0.971	0.245	0.060	0.065	0.958	0.158	0.974
	ENR	0.918	0.420	0.176	0.114	0.799	0.275	0.961
	ENQI	0.947	0.335	0.113	0.089	0.875	0.217	0.965
	ETR	0.949	0.329	0.108	0.088	0.881	0.215	0.971
	ETQI	0.968	0.260	0.068	0.070	0.941	0.170	0.976
Test	**NBR**	0.962	0.280	0.078	0.087	0.909	0.204	0.963
	NBQI	0.974	0.233	0.054	0.072	0.949	0.170	0.974
	ENR	0.942	0.349	0.122	0.110	0.829	0.256	0.961
	ENQI	0.952	0.319	0.101	0.098	0.860	0.232	0.965
	VOPO	0.956	0.302	0.091	0.094	0.882	0.220	0.971
	VothingR	0.971	0.245	0.060	0.076	0.943	0.179	0.976

The stand-alone effect of QIO on model performance was investigated using ablation. Using RMSE and R^2 measures across training, validation, and test sets, each baseline model ETR, NBR, ENR was matched to its QIO-enhanced equivalent ETQI, NBQI, ENQI. Applying QIO regularly produced performance gains, according the results. In the test phase, for example, ETQI rose over ETR by lowering RMSE from 0.302 to 0.965 and raising R^2 from 0.956 to 0.961. Pairing t-tests on the RMSE values over five-fold cross-valuation divides helped one to evaluate statistical significance. All tests had p-values < 0.01, verifying that the noted gains from QIO were statistically significant and not resulting from random fluctuation. These results confirm that QIO is a strong optimizing tool for improving model correctness.

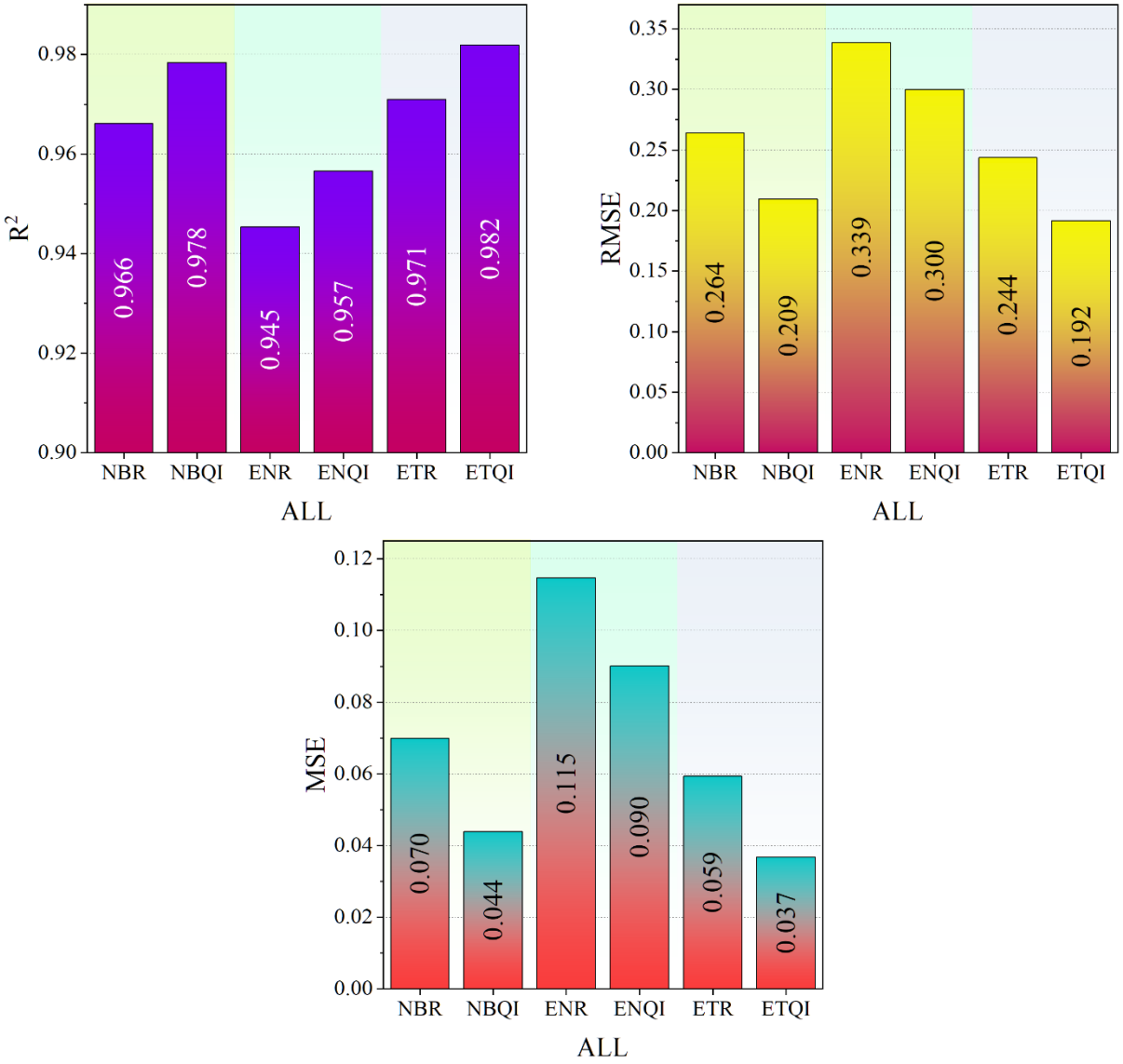
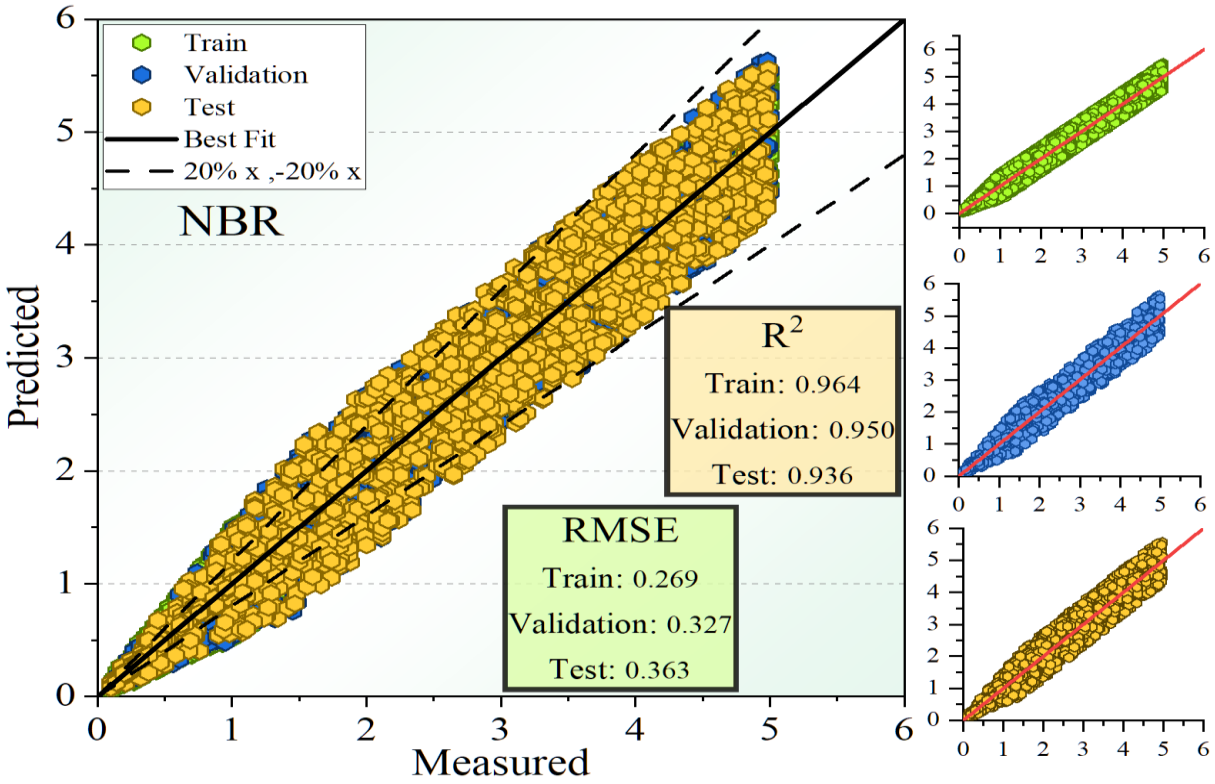
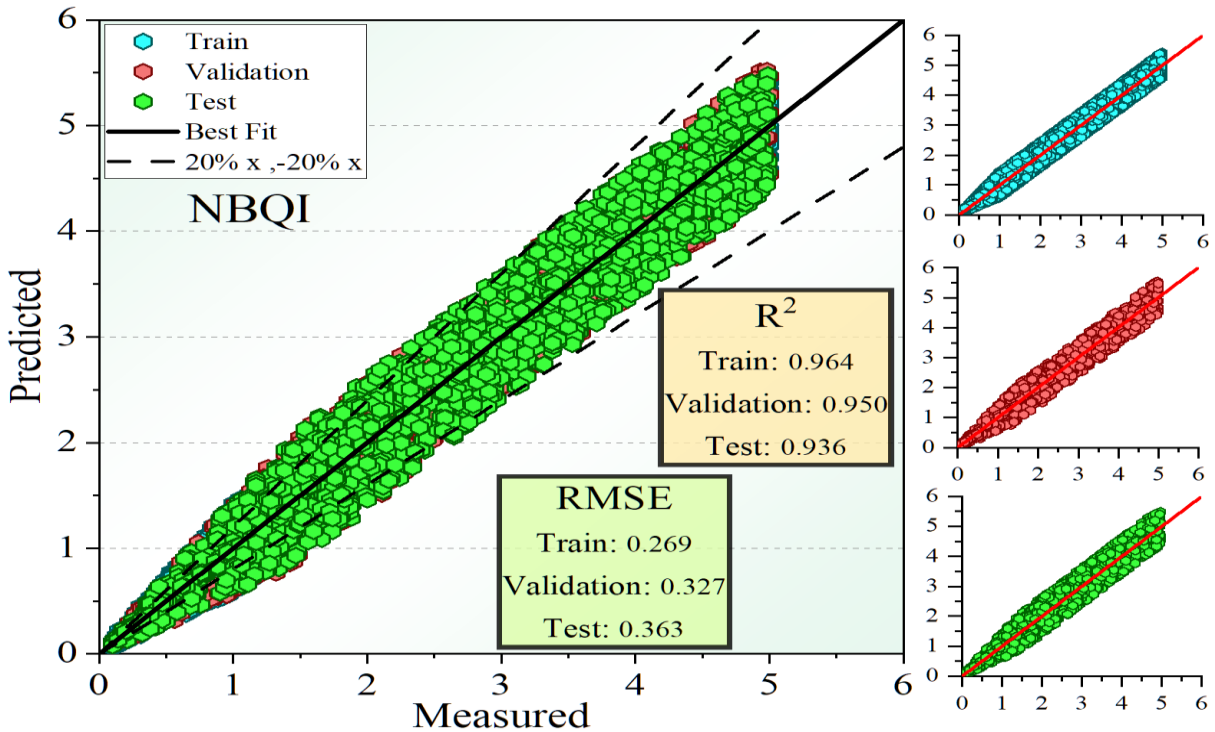


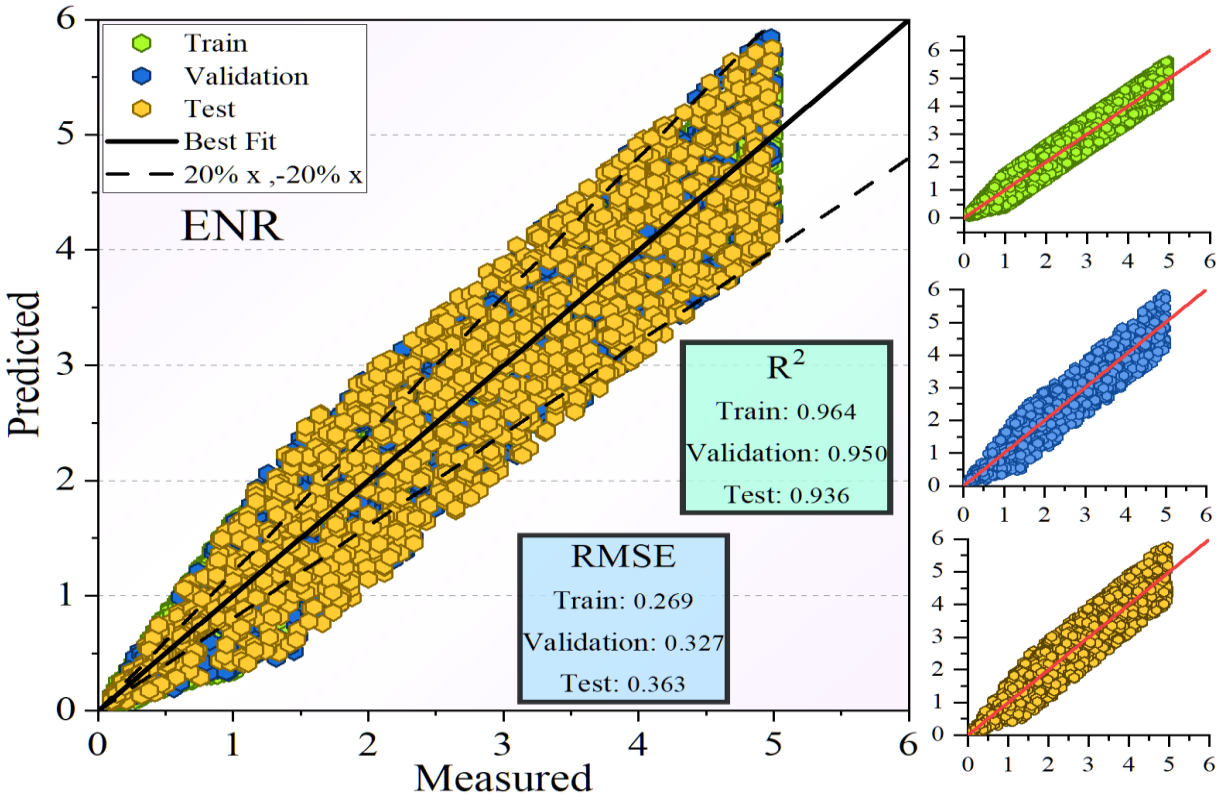
Figure 4: The metric of expected and gauged values

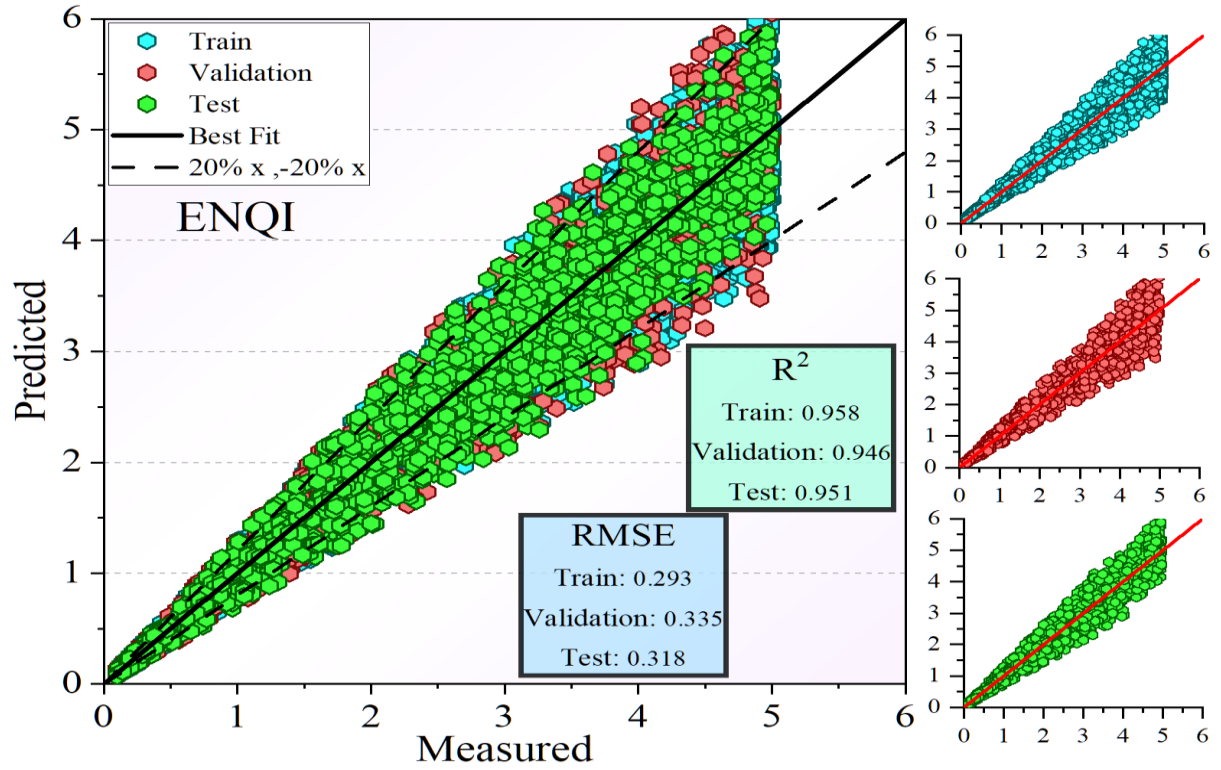
Fig.5 displays the maximum error of the schemes and their performance versus density. The stronger the performance of the schemes whose data points fall closer to the best-fitted line, the better. For example, the ETR scheme has more scattering around its best-fitted line, indicating that it performs less. In comparison, ETQI and NBQI have a higher density close to the best-fitted line, which means these two outperform the others. This is further evidenced by the fact that the distribution of their density would indicate both ETQI and NBQI performing very well during the training phase; in fact, it is toward this that the more compact points lying closer to the best-fit line are pointing to indicate excellent robustness and accuracy in the prediction of results for this model when compared with other schemes, including ETR, which even fail to maintain consistency across phases.

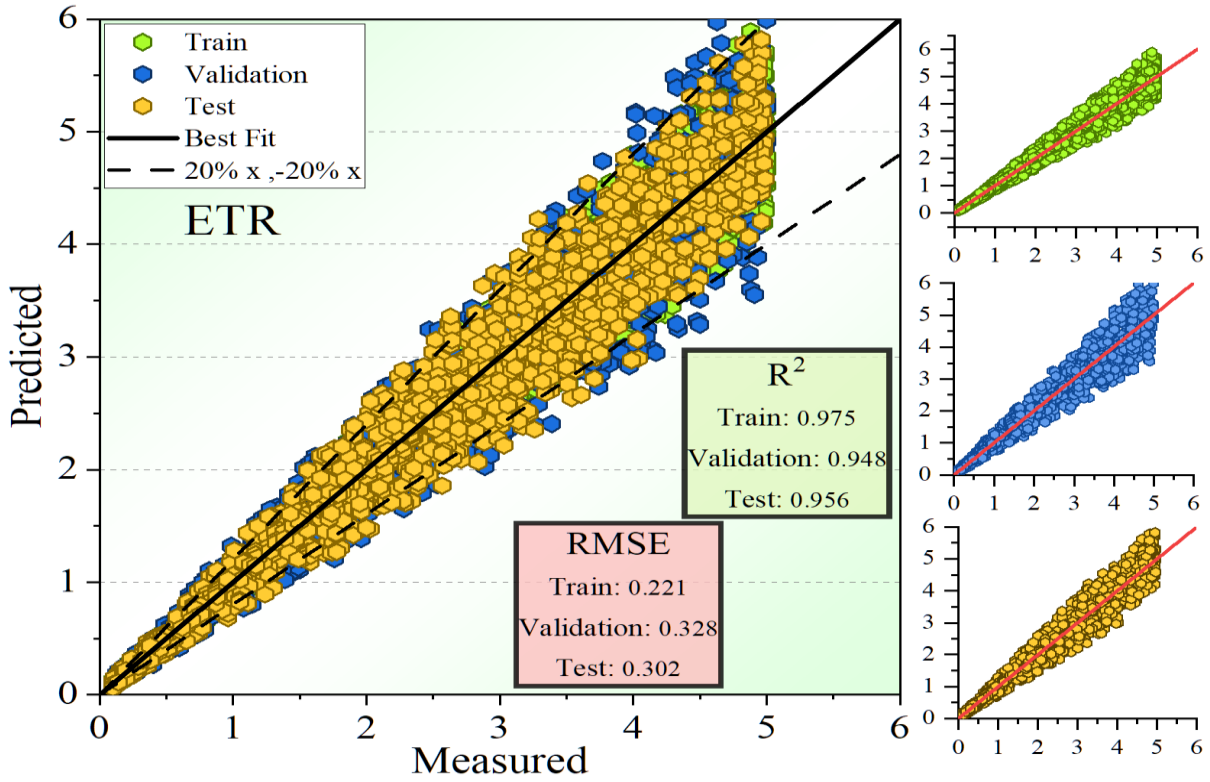
Fig. 6 compares the proposed schemes' error distribution in their training, validation, and testing phases: The x-axis represents the phases, and high density with clustering around the median line promises better performance. For instance, for NBR and NBQI, NBQI outperforms both in all three phases since it has a high density near the median line. More precisely, in comparing ENR versus ENQI, ENQI has more apparent clustering around the median, which means better accuracy. In the case of ETR and ETQI, ETR displays more significant scattering around the median line, which means that ETQI is much stronger than ETR because its distribution is tighter, which finally means better general performance of accuracy.

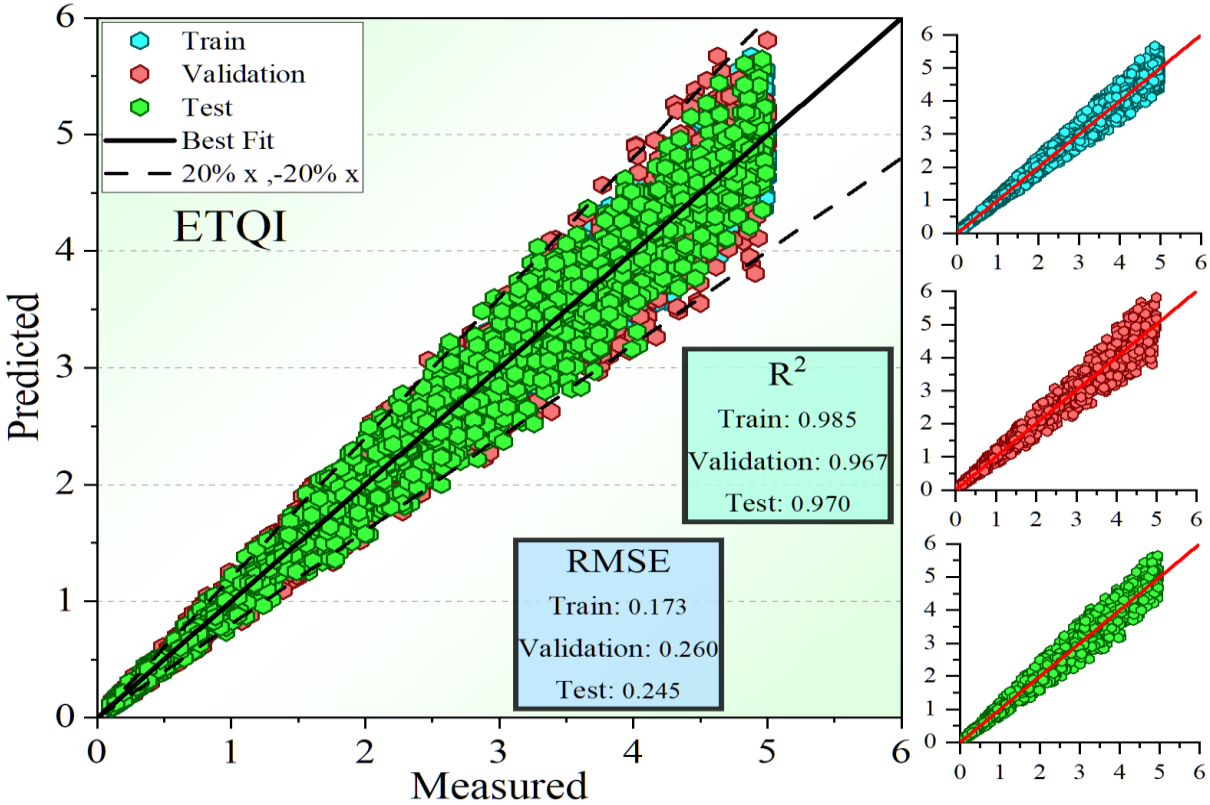

Train
Validation
Test
Best Fit
20% x ,-20% x
NBR
R2
Train: 0.964
Validation: 0.950
Test: 0.936
RMSE
Train: 0.269
Validation: 0.327
Test: 0.363
Predicted
Measured

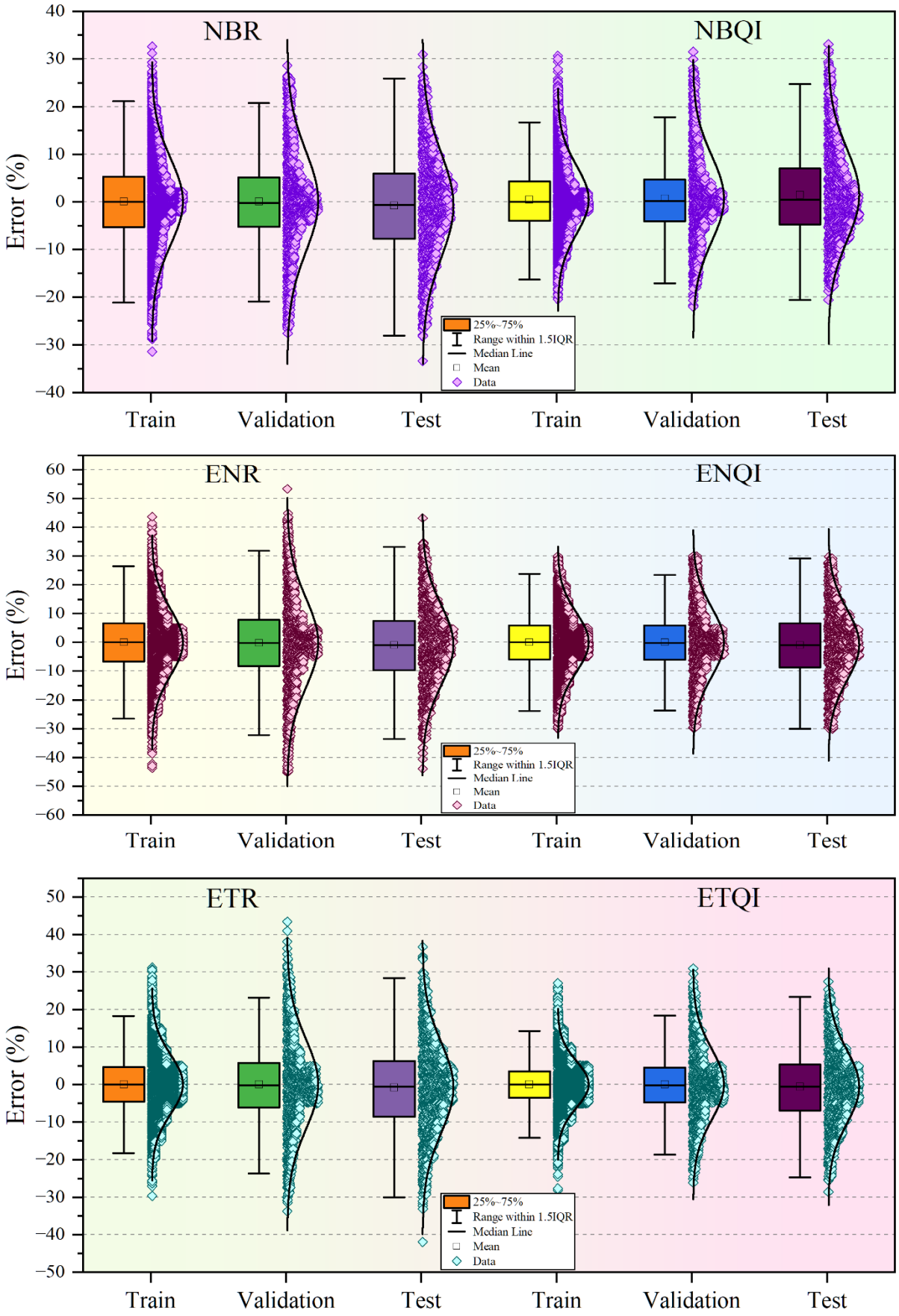

Train
Validation
Test
Best Fit
20% x ,-20% x
NBQI
R2
Train: 0.964
Validation: 0.950
Test: 0.936
RMSE
Train: 0.269
Validation: 0.327
Test: 0.363
Predicted
Measured


Train
Validation
Test
Best Fit
20% x ,-20% x
ENR
R2
Train: 0.964
Validation: 0.950
Test: 0.936
RMSE
Train: 0.269
Validation: 0.327
Test: 0.363
Predicted
Measured


Train
Validation
Test
Best Fit
20% x ,-20% x
ENQI
R2
Train: 0.958
Validation: 0.946
Test: 0.951
RMSE
Train: 0.293
Validation: 0.335
Test: 0.318
Predicted
Measured


Train
Validation
Test
Best Fit
20% x ,-20% x
ETR
R2
Train: 0.975
Validation: 0.948
Test: 0.956
RMSE
Train: 0.221
Validation: 0.328
Test: 0.302
Predicted
Measured


Train
Validation
Test
Best Fit
20% x ,-20% x
ETQI
R2
Train: 0.985
Validation: 0.967
Test: 0.970
RMSE
Train: 0.173
Validation: 0.260
Test: 0.245
Predicted
Measured



Figures 5 and 6: The dispersion of evolved hybrid schemes and error percentage on the box plot.

Fig. 7 presents the feature selection of the best model that yielded the highest performance in predicting smart home energy usage. The radar chart points out the features that are of more importance to the performance of the model with values: appliance and occupancy status are closer to the outer edge, which represents higher importance, at values of 0.022 and 0.012, respectively; this identifies their significant influence on the performance. In contrast, the features, including home-id and holiday, are of low importance, with values ranging between -0.006 and -0.003, hence having minimal effects on the model. Lastly, the season has a moderate influence at 0.007. The chart displays that the main drivers are the appliance and occupancy status, which are the significant features that improve model accuracy. The least contributing feature is the temporal ones like holidays.

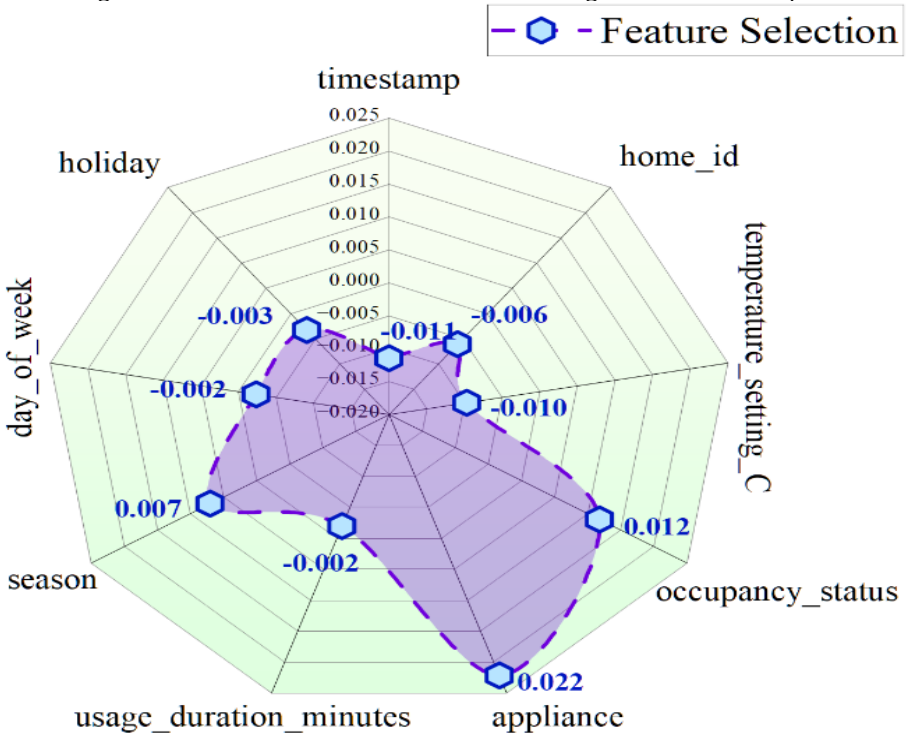


Figure 7: The feature selection method of the best-performed model

K-fold cross-validation was utilized to guarantee the robustness and generalization capacity of the suggested models, dividing the dataset into five equal subsets (K=5). In each iteration, one-fold functioned as the validation set, while the other four were utilized as training data. The procedure was executed five times, and the outcomes were averaged to reduce bias from any one data partition. Fig. 8 depicts the R^2 scores achieved over the five folds for each foundational model: EN, NB, and ET. The ET model consistently attained elevated R^2 values across all folds, signifying enhanced predictive stability. NB also exhibited commendable performance, but with marginally more variability. EN had the poorest performance overall, confirming the comparative efficacy of tree-based methodologies for smart home energy forecasting tasks.

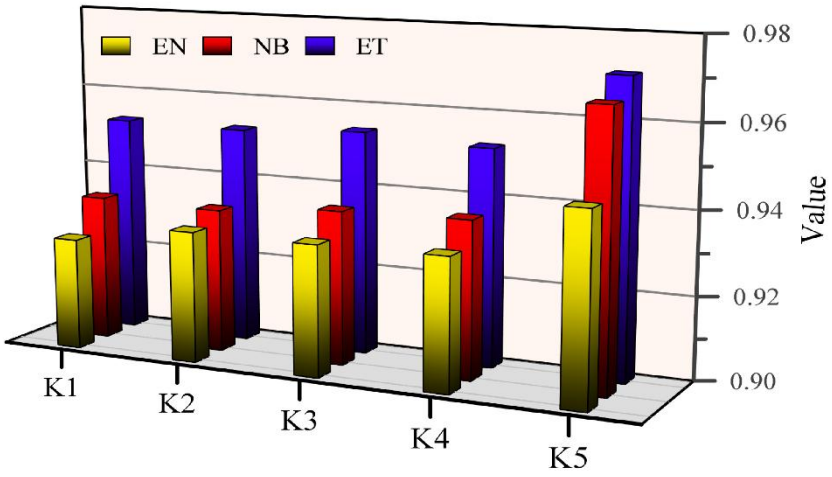


Figure 8: K-fold cross validation.

The Wilcoxon signed-rank test was utilized to evaluate the statistical significance of performance disparities between pairs of models. This non-parametric test is appropriate for comparing the distributions of paired samples, particularly when normality is not assumed. The assessment determines if variations in performance indicators (e.g., RMSE) are statistically significant over several trials. Table 4 displays the results of the Wilcoxon test. Comparisons incorporating ETQI typically produced elevated test results, signifying significant performance enhancements. The disparity between NBR and ETQI yielded a value of 0.497, indicating a significant improvement. Likewise, EN and ETQI achieved a score of 0.289, underscoring QIO's contribution to improving Elastic Net performance. A significant correlation of 0.787 was identified between ENQI and ETR, signifying considerable difference. Certain entries yielded "nan" because of identical matched values or inadequate variety in the data, rendering statistical comparison impracticable.

Table 4: Result of Wilcoxon test.

Difference of models	Parameter	
	p_value	statistic
Def. between NBR and NBQI	nan	nan
Def. between NBR and EN	55672394	0.279
Def. between NBR and ENQI	55854704	0.460
Def. between NBR and ETR	55782484	0.382
Def. between NBR and ETQI	55885794	0.497
Def. between NBQI and EN	nan	nan
Def. between NBQI and ENQI	nan	nan
Def. between NBQI and ETR	nan	nan
Def. between NBQI and ETQI	nan	nan
Def. between EN and ENQI	55655883	0.266
Def. between EN and ETR	55700118	0.303
Def. between EN and ETQI	55683398	0.289
Def. between ENQI and ETR	56102618	0.787
Def. between ENQI and ETQI	55848779	0.454
Def. between ETR and ETQI	55702510	0.305

Table 5 displays the confidence interval values corresponding to the predicted outcomes for three chosen models: NBR, EN, and ETR. Metrics encompass test accuracy (TEST_ACC), log-likelihood scores (TEST_LOSS), and classification confidence (TEST_ACC_CC and TEST_LOSS_CC). The values offer insight into the dependability and stability of the model. NBR and ETR demonstrated negligible test accuracy mistakes (0.000), but EN displayed more fluctuation (0.001–0.009), indicating increased inconsistency in predictions. All models had reasonably narrow loss intervals (about 9.2–11.6), signifying modest variation. ETR exhibited remarkable stability in classification confidence, corresponding with its robust overall performance. The confidence intervals affirm the durability of the ETR and NBR models, while

underscoring EN's somewhat less steady performance under test conditions.

Table 5: Confidence interval associated with the prediction results.

	TEST_ACC	TEST_LOSS	TEST_ACC	TEST_LOSS	TEST_ACC	TEST_LOSS
NBR	0.000	11.674	0.000	10.590	0.000	9.237
EN	0.001	11.660	0.006	10.237	0.009	9.256
ETR	0.000	11.639	0.002	10.237	0.009	9.257

6 Conclusion

Smart homes have now started to present a solution for the increasing demand for energy from the residential sector. They are offering new ways of curbing and managing energy consumption. These smart modern homes are equipped with advanced sensors and smart devices that can provide accurate energy management, reducing costs while minimizing environmental impact. However, several challenges complicate this process, including user behavior, varying weather conditions, and the efficiency of household appliances.

While dealing with these complexities, ML tactics can be deployed to predict energy usage in smart homes. Some of these schemes used in this paper include Extra Tree Regression (ETR), Naive Bayes Regression (NBR), and Elastic Net Regression (ENR). These schemes, enhanced by Quadratic Interpolation Optimization (QIO), deliver accurate predictions, optimizing energy efficiency in smart homes.

A fascinating insight is recorded with the performance analysis of the schemes. ETQI has shown the best performance among all analysis phases with an R^2 of 0.982. The second best is NBQI, which also held excellent predictive capabilities with an R^2 of 0.978. Elastic Net Regression is the weakest among the optimized schemes, with an R^2 of 0.945. The following comparative analysis underlines the importance of choosing appropriate machine learning schemes that will contribute to advanced energy management strategies in smart homes and create a pathway toward more sustainable living environments.

Despite its advantages, machine learning has certain limitations. The following are some limitations of smart home energy usage prediction using machine learning: it does not consider some exterior factors, including weather changes or incorrect appliance performances. This might be at the risk of overfitting if the schemes become too complicated and generalize poorly on new data. Finally, various devices and technologies for the realization of present ideas face severe tests in standardizing the input data, which may question the reliability of the prediction when implemented in real life.

The present work suffers a restriction in relying on a single synthetic dataset. Resources and scope limitations limited the study to the Kaggle Smart Home Energy Usage dataset. Additional real-world datasets including UK-DALE and REDD will be included in next study to evaluate the generalizability and robustness of the suggested models throughout several household energy usage scenarios. Particularly under dynamic household situations and data streams experienced in actual smart home scenarios, future research will investigate merging deep learning and online learning models to improve flexibility and accuracy in real-time energy prediction tasks. The present work offers high-accuracy energy consumption estimates, which form a fundamental component for energy efficiency improvement in smart homes even if it does not apply active control or scheduling algorithms. These predictive outputs can be included into future systems including demand-side management, load scheduling, or appliance-level control strategies, so enabling informed decision-making targeted at lowest energy consumption and maximum overall energy efficiency in home settings. Using smart home energy prediction models in the real world calls for handling various pragmatic issues. This cover addressing sensor noise, missing data management, and offering streaming or real-time prediction capability. Although the present work shows good performance in controlled circumstances, future research should concentrate on creating strong, adaptable systems competent of running under noisy, incomplete, and constantly changing data environments.

Acknowledgements

I wish to take this opportunity to acknowledge that no individuals or organizations require acknowledgment for their contributions to this investigation.

Competing of interests

The scholar claims no competing interests.

Authorship contribution statement

The scholar contributed to the study's conception and configuration. Yuanjun ZHANG executed data gathering, modeling, and inspection. Yuanjun ZHANG wrote the first draft of the manuscript and commented on previous versions of the manuscript.

Data availability

Data is available at request.

Declarations

The scholar claims no competing interests.

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.

Funding

This research did not receive a specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

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.

Research involving human participants and animals

The observational exploration on medical staff does not require an ethical code. Therefore, this investigation was not required to acquire a moral code.

Informed consent

This option is not necessary because the data were gathered from the references.

References

[1] S. O. Oyedepo, "On energy for sustainable development in Nigeria," *Renewable and sustainable energy reviews*, Elsevier, 16(5): 2583–2598, 2012. https://doi.org/10.1016/j.rser.2012.02.010.

[2] B. Lashkari, Y. Chen, and P. Musilek, "Energy management for smart homes—State of the art," *Applied Sciences*, MDPI, 9(17): 3459, 2019. https://doi.org/10.3390/app9173459.

[3] B. Zhou *et al.*, "Smart home energy management systems: Concept, configurations, and scheduling strategies," *Renewable and Sustainable Energy Reviews*, Elsevier, 61: 30–40, 2016. https://doi.org/10.1016/j.rser.2016.03.047.

[4] H. Xu, Y. He, X. Sun, J. He, and Q. Xu, "Prediction of thermal energy inside smart homes using IoT and classifier ensemble techniques," *Comput Commun*, Elsevier, 151: 581–589, 2020. https://doi.org/10.1016/j.comcom.2019.12.020.

[5] B. Li, P. Hathaipontaluk, and S. Luo, "Intelligent oven in smart home environment," in *2009 international conference on research challenges in computer science*, Shanghai, China, IEEE, 2009: 247–250. DOI: 10.1109/ICRCCS.2009.70.

[6] J. Choi, D. Shin, and D. Shin, "Research on design and implementation of the artificial intelligence agent for smart home based on support vector machine," in *Advances in Natural Computation: First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part I 1*, Berlin, Heidelberg, Springer, 2005: 1185–1188. https://doi.org/10.1007/11539087_157.

[7] A. Kailas, V. Cecchi, and A. Mukherjee, "A survey of communications and networking technologies for energy management in buildings and home automation," *Journal of Computer Networks and Communications*, Wiley Online Library, 2012(1): 932181, 2012. https://doi.org/10.1155/2012/932181.

[8] J. Cheng and T. Kunz, "A survey on smart home networking," *Carleton University, Systems and Computer Engineering, Technical Report SCE-09-10*, Elsevier, 2009. https://doi.org/10.1016/j.cose.2022.102677.

[9] B. Qolomany *et al.*, "Leveraging machine learning and big data for smart buildings: A comprehensive survey," *IEEE access*, IEEE, 7: 90316–90356, 2019. DOI: 10.1109/ACCESS.2019.2926642.

[10] D. Xu *et al.*, "A classified identification deep-belief network for predicting electric-power load," in *2018 2nd IEEE conference on energy internet and energy system integration (EI2)*, Beijing, China, IEEE, 2018: 1–6. DOI: 10.1109/EI2.2018.8582314.

[11] U. Berardi, "Building energy consumption in US, EU, and BRIC countries," *Procedia Eng*, Elsevier, vol. 118: 128–136, 2015. https://doi.org/10.1016/j.proeng.2015.08.411.

[12] E. D. Williams and H. S. Matthews, "Scoping the potential of monitoring and control technologies to reduce energy use in homes," in *Proceedings of the 2007 IEEE International Symposium on Electronics and the Environment*, Orlando, FL, USA, IEEE, 2007: 239–244. DOI: 10.1109/ISEE.2007.369401.

[13] J. G. Koomey, S. A. Mahler, C. A. Webber, and J. E. McMahon, "Projected regional impacts of appliance efficiency standards for the US residential sector," *Energy*, Elsevier, 24(1): 69–84, 1999. https://doi.org/10.1016/S0360-5442(98)00065-6.

[14] N. Gudi, L. Wang, V. Devabhaktuni, and S. S. S. R. Depuru, "A demand-side management simulation platform incorporating optimal management of distributed renewable resources," in *2011 IEEE/PES Power Systems Conference and Exposition*, Phoenix, AZ, USA, IEEE, 2011: 1–7. DOI: 10.1109/PSCE.2011.5772450.

[15] R. G. Pratt, C. C. Conner, B. A. Cooke, and E. E. Richman, "Metered end-use consumption and load shapes from the ELCAP residential sample of existing homes in the Pacific Northwest," *Energy Build*, Elsevier, 19(3): 179–193, 1993. https://doi.org/10.1016/0378-7788(93)90026-Q.

[16] S.-H. Ling, F. H.-F. Leung, H.-K. Lam, and P. K.-S. Tam, "Short-term electric load forecasting based on a neural fuzzy network," *IEEE Transactions on industrial electronics*, IEEE,

50(6): 1305–1316, 2003. DOI: 10.1109/TIE.2003.819572.

[17] A. Veit, C. Goebel, R. Tidke, C. Doblander, and H.-A. Jacobsen, "Household electricity demand forecasting: benchmarking state-of-the-art methods," in *Proceedings of the 5th international conference on Future energy systems*, ACM, 2014: 233–234. https://doi.org/10.1145/2602044.2602082.

[18] N. Arghira, L. Hawarah, S. Ploix, and M. Jacomino, "Prediction of appliances energy use in smart homes," *Energy*, Elsevier, 48(1): 128–134, 2012. https://doi.org/10.1016/j.energy.2012.04.010.

[19] S. Bharati, M. A. Rahman, R. Mondal, P. Podder, A. A. Alvi, and A. Mahmood, "Prediction of energy consumed by home appliances with the visualization of plot analysis applying different classification algorithm," in *Frontiers in Intelligent Computing: Theory and Applications: Proceedings of the 7th International Conference on FICTA (2018), Volume 2*, Singapore, Springer, 2020: 246–257. https://doi.org/10.1007/978-981-13-9920-6_25.

[20] D. Scott, T. Simpson, N. Dervilis, T. Rogers, and K. Worden, "Machine learning for energy load forecasting," in *Journal of Physics: Conference Series*, IOP Publishing, 2018: 12005. DOI: 10.1088/1742-6596/1106/1/012005.

[21] A. Kavousian, R. Rajagopal, and M. Fischer, "Ranking appliance energy efficiency in households: Utilizing smart meter data and energy efficiency frontiers to estimate and identify the determinants of appliance energy efficiency in residential buildings," *Energy Build*, Elsevier, 99: 220–230, 2015. https://doi.org/10.1016/j.enbuild.2015.03.052.

[22] J. Lu *et al.*, "The smart thermostat: using occupancy sensors to save energy in homes," in *Proceedings of the 8th ACM conference on embedded networked sensor systems*, 2010: 211–224. https://doi.org/10.1145/1869983.1870005.

[23] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach Learn*, Springer, 63: 3–42, 2006. https://doi.org/10.1007/s10994-006-6226-1.

[24] A. Sharafati, S. B. H. S. Asadollah, and M. Hosseinzadeh, "The potential of new ensemble machine learning models for effluent quality parameters prediction and related uncertainty," *Process Safety and Environmental Protection*, Elsevier, 140: 68–78, 2020. https://doi.org/10.1016/j.psep.2020.04.045.

[25] L. Breiman, "Random forests," *Mach Learn*, Springer, 45: 5–32, 2001. https://doi.org/10.1023/A:1010933404324.

[26] V. John, Z. Liu, C. Guo, S. Mita, and K. Kidono, "Real-time Lane estimation using deep features and extra trees regression," in *Image and Video Technology: 7th Pacific-Rim Symposium, PSIVT 2015, Auckland, New Zealand, November 25-27, 2015, Revised Selected Papers 7*, Cham, Springer, 2016, : 721–733. https://doi.org/10.1007/978-3-319-29451-3_57.

[27] G. Mishra, D. Sehgal, and J. K. Valadi, "Quantitative structure activity relationship study of the anti-hepatitis peptides employing random forests and extra-trees regressors," *Bioinformation*, National Library of Machine, 13(3): 60, 2017. DOI: 10.6026/97320630013060.

[28] E. Frank, L. Trigg, G. Holmes, and I. H. Witten, "Naive Bayes for regression," *Mach Learn*, Springer, 41: 5–25, 2000. https://doi.org/10.1023/A:1007670802811.

[29] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J R Stat Soc Series B Stat Methodol*, Oxford Academic, 67(2): 301–320, 2005. https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[30] C. Hans, "Elastic net regression modeling with the orthant normal prior," *J Am Stat Assoc*, Taylor & Francis, 106(496):1383–1393, 2011. https://doi.org/10.1198/jasa.2011.tm09241.

[31] Q. F. Xu, X. H. Ding, C. X. Jiang, K. M. Yu, and L. Shi, "An elastic-net penalized expectile regression with applications," *J Appl Stat*, Taylor & Francis, 48(12): 2205–2230, 2021. https://doi.org/10.1080/02664763.2020.1787355.

[32] Z. Zhang, Z. Lai, Y. Xu, L. Shao, J. Wu, and G.-S. Xie, "Discriminative elastic-net regularized linear regression," *IEEE Transactions on Image Processing*, IEEE, 26(3): 1466–1481, 2017. DOI: 10.1109/TIP.2017.2651396.

Multi-Objective Optimization of Energy Retrofit for Existing Buildings Using an Enhanced Chimp Optimization Algorithm

Hualong Zhang
Zhengzhou Urban Construction Vocational College, Zhengzhou, China
E-mail: hooloong271523429@163.com

Keywords: existing buildings, retrofit, energy consumption, multi-objective optimization, chimp optimization algorithm

Received: February 26, 2025

This study proposes a modified multi-objective chimp optimization algorithm (MO-ChOA) to optimize energy consumption retrofits for existing buildings to reduce operational energy consumption. First, a back-propagation neural network was used to predict carbon emission factors, and a mathematical model for multi-objective optimization of building energy consumption was established. Subsequently, Hammersley sequence and somersault foraging heuristic strategy were introduced to improve MO-ChOA, and the improved MO-ChOA was used to solve the multi-objective optimization model of building energy consumption. The experimental results showed that the proposed improved MO-ChOA had a reverse generation distance of 0.113 and a super volume of 0.973, which was superior to the traditional MO-ChOA. This study proposed a modified MO-ChOA to optimize energy consumption retrofits for existing buildings, aiming to reduce operational energy consumption. First, a back-propagation neural network was used to predict carbon emission factors, and a mathematical model for multi-objective optimization of building energy consumption was established. In the optimization of building energy consumption, the proposed method had a solution interval of only 0.110, an average uniformity evaluation index of more than 0.8, a carbon emission saving rate of 0.7-0.95, a cost saving rate of 0.79, and an investment return rate of 0.685, which could effectively reduce carbon emissions and operating costs. The study encourages the development and innovation of energy-saving retrofitting technologies and offers new technical tools and solutions for retrofitting existing buildings with lower energy use.

Povzetek: Predlagan je izboljšan večciljni metahevristični algoritem optimizacije šimpanzov z Hammersleyjevim vzorčenjem in strategijo, ki učinkovito zmanjša porabo energije ter emisije pri energetski sanaciji stavb.

1 Introduction

The building industry is a major contributor to energy consumption (EC) and CO2 emissions. A large number of old residential buildings have poor envelope structures, old and inefficient equipment, and a large proportion of non-energy efficient buildings. Moreover, the lack of operation and maintenance management leads to a high percentage of building whole-life EC in the total EC [1-2]. In conclusion, achieving carbon peak, carbon neutrality, and advancing green, low-carbon, and high-quality development all depend heavily on accelerating the promotion of energy saving (ES) and carbon reduction (CR) in the building industry. In the meanwhile, building CR initiatives can raise the energy efficiency level of buildings by utilizing ES materials, optimizing building design, and taking other steps. Buildings must conduct ES and CR since this can give occupants a more comfortable and healthful living environment [3]. Building operations contribute significantly to carbon emissions (CEs). Reducing EC directly lowers emissions while improving energy efficiency. Retrofitting existing buildings (EBs) for energy efficiency is still fraught with difficulties, though. On the one hand, the current energy efficiency retrofit schemes are less generalizable due to the complexity of the EB stock and the range of building kinds, ages, and purposes. For example, building types can be divided into residential buildings, public buildings, industrial buildings, and agricultural buildings according to their functions. According to the structural form, it can be divided into wall load-bearing structure, skeleton structure, etc. It also includes special types of buildings such as historical buildings and memorial buildings. On the other hand, the integration of multiple technologies such as renewable energy integration and smart building technologies for the energy efficiency retrofit design of existing EBs increases the complexity of technology implementation [4-5]. Multi-objective optimization algorithm (MOOA) refers to an algorithm that searches for a set of non-inferior solutions in optimization problems where there are conflicts or contradictions among multiple objective functions (OFs). In recent years, MOOA has been widely applied in fields such as engineering design, supply chain management, and logistics optimization due to its unique advantages in handling multiple conflicting

or contradictory OFs. The energy-saving renovation of EBs faces challenges such as complex types and poor universality of schemes. This study aims to optimize the EC and cost of EBs through an improved MOOA to reduce operational EC, reduce CEs, and improve energy use efficiency. The study first determines the CE indexes of EB operations based on the ES objectives and CE factors, and constructs a mathematical multi-objective optimization (MOO) model. Then, the Hammersley sequence and somersault foraging heuristic strategy are introduced to improve the Chimp optimization algorithm (ChOA), which aims to enhance the multi-objective optimization performance of ChOA and achieve energy and cost optimization of EBs.

The study is mainly composed of four sections. The first section reviews the relevant studies on building ES and emission reduction optimization at home and abroad. The second section describes the construction of MOO model for EC of EBs and the design process of the improved multi-objective chimp optimization algorithm (MO-ChOA) solution algorithm. In the third section, the improved MO-ChOA solution algorithm is tested for performance and the EC optimization analysis is carried out. The fourth section summarizes the experimental results.

2 Related work

Renovating buildings contributes to a decrease in EC and CEs, which has broad social and practical implications as well as significant theoretical importance. Research on ES and emission reduction in buildings has been conducted by numerous academics and researchers. To properly analyze the energy performance of buildings, Fallah et al. used feed-forward neural network combined with electrostatic discharge algorithm to optimize and design an annual thermal energy demand prediction model for residential buildings. Experimental results indicated that the method demonstrated higher prediction accuracy compared to atomic search optimization, future search algorithm and satin gardener bird optimization [6]. To control building EC towards sustainable buildings, Tahmasebinia et al. explored the potential of building information modeling (BIM) and digital twin technology in building energy efficiency and management, supported by citing case studies [7]. An energy-efficient building management information system based on multi-intelligent body topology was proposed by Verma et al. to address the conflict between occupant comfort and EC in residential structures. To find the best way to reduce EC and increase comfort, the system first optimized the environmental parameters using a restricted nonlinear optimization algorithm. Artificial intelligence (AI) and deep learning techniques were then used to further train and validate the system. The findings proved that the system successfully decreased EC while maintaining high levels of occupant satisfaction with regard to air quality, thermal comfort, and visual comfort [8]. To optimize the design of commercial buildings in the tropical climate of India to enhance thermal performance and reduce EC, Sana et al. used building energy simulation optimization (BESO) in conjunction with grasshopper optimization algorithm (GOA). The experimental results indicated that the method performed better in reducing the annual thermal load and was more computationally efficient with a reduction in computation time by 4.18%-37.11% [9]. To solve the problems of low efficiency and high cost of traditional building construction and to promote the green and sustainable development of buildings, Wu constructed a multi-objective decomposition building energy efficiency optimization model based on BIM. It also introduced the intelligent body-assisted multi-objective particle swarm optimization (MOPSO) to enhance the optimization effect. According to the experimental findings, the model exhibited good convergence, adaptability, and optimization time performance. In particular, it achieved the shortest search time, the smallest total EC and the highest oversize volume index in the ES optimization design of urban single-room office buildings [10]. Khan et al. proposed a bacterial foraging ant colony optimization algorithm to schedule electricity consumption for building energy optimization problems. The results showed that the proposed algorithm could reduce electricity cost and improve user comfort [11]. Xu et al. proposed a building EC optimization model based on convolutional neural networks and BIM. The results showed that the proposed method could reduce building EC by 24.53% and increase natural lighting by 18.98% [12]. Gheoany et al. proposed an energy management system based on branch and bound algorithm and particle swarm optimization algorithm for building energy management problems. The results showed that the proposed method could reduce the electricity bills and standard number of poles by 28% and 49.32%, respectively [13].

To optimize the building envelope and photovoltaic modules to enhance the performance of energy-efficient buildings, Zhao et al. constructed an efficient hybrid algorithm using PSO, support vector machines, non-dominated sorting genetic algorithms (NSGA) II, and sequential selection of optimization techniques. The method significantly reduced building EC by 41%, CE by 34% and retrofit operation cost by 20% [14]. Building envelopes can be retrofitted to adapt to the future climate using a MOO design process that Ding et al. designed to address the impact of global climate change on building EC. The method utilized back-propagation neural network (BPNN) to establish the correlation between design parameters and performance metrics and NSGA-III to optimize the retrofit strategy. The results revealed that the method was more efficient in terms of ES solutions, emphasizing the importance of considering climate change factors [15]. To address the problem of high dependence on expert knowledge and subjective decision making in traditional green building design, Khan et al. proposed an innovative framework integrating BIM, interpretable AI and MOO. The framework used Bayesian optimization for energy prediction and MOO with the help of multi-objective evolutionary algorithm (MOEA) based on decomposition. The results revealed that the framework achieved highly accurate prediction and significant optimization results, improved energy

efficiency and thermal comfort, and reduced CO_2 emissions [16]. To solve the MOO problem in the renovation of historic buildings, Wei et al. proposed a renovation method that combines the building envelope with systematic standard measures. Using energy simulation technology and NSGA-II algorithm, multiple retrofit solutions were generated by logarithmic additive decomposition, and then the weighted sum method was applied to find the optimal solution. Taking the Nanjing Courtyard as an example, EC was reduced by 63.62% after the retrofit, the net present value increased by 151.84%, and the CR rate was up to 60.48%, which realized the balance of ES, CR and economy [17]. The summary table of the above-mentioned related work is shown in Table 1.

Table 1: Summary table of related work

Literature	Method	Optimization objectives	Performance indicators	Limitation
[6]	Feed-forward neural network, electrostatic discharge algorithm	Forecast of annual thermal energy demand for residential buildings	Prediction accuracy	High dependence on data quality and integrity
[7]	BIM, digital twin technology	Building energy efficiency and management	Energy saving effect and management efficiency	The technical implementation is complex and the initial cost is high
[8]	Multi agent topology, constrained nonlinear optimization, deep learning	Minimizing energy consumption and maximizing living comfort	Satisfaction with energy consumption, thermal comfort, visual comfort, and air quality	The system requires high real-time performance, and there is a significant demand for computing resources during large-scale deployment
[9]	BESO, GOA	Optimization of thermal performance of commercial buildings in tropical climate	Annual heat load, calculation time	Specific to tropical climates, limited generalization ability to other climate types
[10]	BIM, Multi-objective decomposition model and MOPSO	Energy efficiency optimization of urban single room office buildings	Search time, total energy consumption, super large volume index	Sensitive to building types and requiring customized adjustments
[11]	Bacterial foraging ant colony optimization algorithm	Optimization of building electricity dispatch	Electricity cost, user comfort	The algorithm has high complexity and requires frequent parameter adjustments in practical applications
[12]	Convolutional neural networks, BIM	Optimization of building energy consumption and natural lighting	Energy consumption and natural lighting	Relying on large-scale training data, model training takes a long time
[13]	Branch and bound algorithm, PSO	Home energy management	Electricity cost, standard pole count	Insufficient scalability of large-scale residential areas
[14]	PSO, SVM, and NSGA-II	Building envelope structure and optimization of photovoltaic modules	Energy consumption, carbon emissions, and operating costs	High complexity of multi algorithm integration

[15]	BPNN and NSGA-III	Future climate adaptive building envelope renovation	Efficiency	The uncertainty of climate change prediction affects the reliability of optimization results
[16]	BIM, explainable AI and MOEA	Energy performance analysis of green buildings	Energy efficiency, carbon dioxide emissions	Explainable AI implementation is complex, requiring a balance between model accuracy and interpretability
[17]	NSGA-II, logarithmic additive decomposition	Energy saving renovation of historical buildings	Energy consumption, net present value, carbon reduction rate	There are many restrictions on the renovation of historical buildings, which require coordination among multiple stakeholders

In summary, in the area of ES and emission reduction in building retrofitting, which successfully increases building energy efficiency and comfort by adding cutting-edge algorithms and technologies, current research has produced noteworthy outcomes. Nonetheless, the majority of current studies concentrate on optimizing the structure and performance of a particular building.

To reduce the operational EC of EBs, the study innovatively constructs a mathematical MOO model and introduces intelligent optimization algorithms to improve the optimization, in order to achieve further optimization of EC.

3 optimization study of operational energy retrofit in EBS

The study unfolds the optimal design of operational energy retrofit of EBs in the context of carbon peak ES and the MOO study of ES with the help of multi-objective intelligent algorithms.

3.1 Construction of MOO model based on building ES and emission reduction

The determination of CE indexes is the basis for the MOO study of EC in the operation of EBs. By defining the CE index, it can quantify the CE of EBs in the process of operation, and provide a clear target and evaluation standard for the subsequent optimization study. To establish the CE index of EBs in the operation stage to realize the ES standard, the study firstly launches the prediction of building carbon peak CE factor. The prediction process is shown in Figure 1.

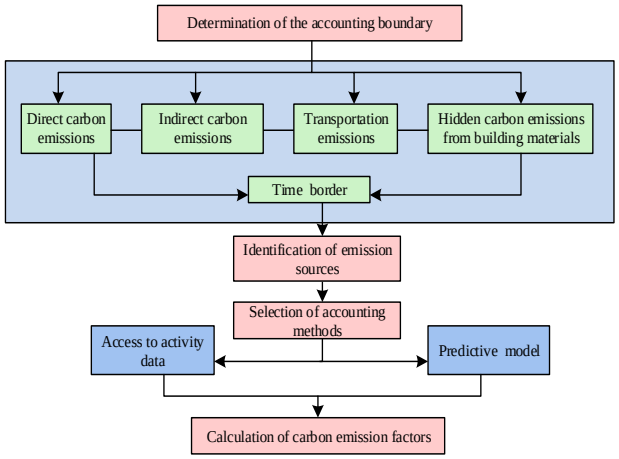


Figure 1: CE accounting process of EBs

As shown in Figure 1, the spatial boundary of the CE calculation is the area of the building and its annexes, and the time boundary is the life cycle of the building. The direct source of CE is the combustion of fuel in buildings, and the calculation factors include fuel type, fuel consumption, and fuel CE factor. Indirect CE refers to the use of electricity and heat, with calculation factors including electricity consumption, CE factors from electricity sources, and CE factors from heat consumption and heat. Transportation emissions refer to the transportation of building materials (BMs) and construction processes, with calculation factors including transportation distance, mode, and CE factors of transportation vehicles. Hidden CE refer to the CE during the production, transportation, and construction processes of BMs. The calculation factors include the type of BMs and the CE factors during the production and transportation stages of the materials. In addition, included are CEs from building-related transportation operations as well as CEs from the choice and application of BMs. With reference to the *Standard for Calculating CEs from Buildings (GB/T51366-2019)*, Equation (1) illustrates how CEs G are calculated during a building's whole life cycle.

$$G = G_s + G_j + G_y + G_c \tag{1}$$

In Equation (1), G_s, G_j, G_y, and G_c represent the emissions ($kgCO_2e/m^2$) from the production of BMs, building construction, operation, and demolition stages,

respectively. The CEs in the building production stage mainly come from the mining, production, transportation, and construction of BMs. Carbon accounting is shown in Equation (2).

$$G_s = \sum AD_i \times EF_i \tag{2}$$

In Equation (2), AD_i denotes the activity volume of the first BM. EF_i is the CE factor of the i th BM, i.e. the CE corresponding to the unit activity volume. Carbon accounting in the operation phase of the building is shown in Equation (3).

$$G_y = \sum P_i \times PF_i \tag{3}$$

In Equation (3), P_i denotes the amount of energy consumed by Type i energy sources. PF_i represents the CE factor of the i th type of energy. The EB retrofit process needs to consider the CEs from operation and retrofit at the same time. The calculation of CE G_g of building retrofit is shown in Equation (4).

$$G_g = G_{gs} + G_{gc} \tag{4}$$

In Equation (4), G_{gs} and G_{gc} represent the CEs caused by the production of BMs and construction in the remodeling process, respectively. Based on the CE accounting of EBs at different stages, the study builds a basic CE factor prediction framework with the help of classical BPNN. BPNN consists of input layer, hidden layer, and output layer (OL). The work signal and the error signal propagate in the BPNN simultaneously. Until the actual output is produced at the output, the work signal continues to travel forward. Layer by layer, the erroneous signal travels backwards from the output. The expression of the error E is shown in Equation (5).

$$E = \frac{1}{2}\sum_{s=1}^{N}\sum_{z=1}^{m}\left(d_{zs}\text{-}y_{zs}\right)^2 \tag{5}$$

In Equation (5), s denotes the data sample. z denotes the input. d and y denote the expectation and prediction of the OL nodes, respectively. To make the output as close to the intended target as possible, the learning objective of BPNN is to adjust its weights based on the error between the network's actual output and the target vector. The updating process of the weight factor SS is shown in Equation (6).

$$\Delta w = w - \alpha\frac{\partial E}{\partial w} \tag{6}$$

In Equation (6), α denotes the hyper-parameter learning rate, which takes the value of 0.01. In addition, BPNN utilizes the chain rule to decompose the gradient of the loss function into local gradients. The simplified computational procedure is shown in Equation (7).

$$\delta_k = -\frac{\partial E}{\partial I_k} = -\frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial I_k} = (d_k - y_k)f'(I_k) = y_k(1-y_k)(d_k - y_k) \tag{7}$$

In Equation (7), δ_k denotes the error signal. I_k denotes the input of the k th node of the OL. Finally, the study predicts the CE factor when the carbon peaks according to BPNN. The input layer of BPNN receives key feature data that affect the CE factor, including BM parameters, EC data, building structure parameters, and environmental parameters. The hidden layer is used for nonlinear mapping and feature extraction, adopting a single hidden layer structure to avoid overfitting and using the ReLU function to alleviate the gradient vanishing problem. The prediction target of the OL is the CE factor during the building operation phase, with a node count of 1. The activation function is selected as a linear function to directly output continuous values. The core module of the CE prediction framework, known as BPNN, possesses the capability to dynamically modify the CE factor. This modification is contingent upon real-time input EC data and building parameters. BPNN facilitates the iterative solution of multi-objective optimization models. Subsequently, the spatial model of the building is constructed using computer-aided design software according to the construction standards of the EB, and the building model constructed by the study is oriented in the north-south direction with reference to *GB50176-2016 Thermal Design Code for Civil Buildings*. Moreover, it comprehensively considers the building structure and environmental characteristics to determine the transformation parameters, including the transformation of lighting, thermal environment, and envelope structure. Finally, the study synthesizes the CE factor and ES demand at the time of carbon peaking, and determines the CE index of EBs with reference to the *Energy Conservation Design Standard for Public Buildings (GB50189-2015)*.

Energy retrofit of EBs mainly involves windows, lighting, envelope, cooling and heating systems. Window retrofit is an important part of EC retrofit, and the study mainly takes the window-to-ground ratio (WGR) as the retrofit standard. The lighting schematic of the building rooms is shown in Figure 2.

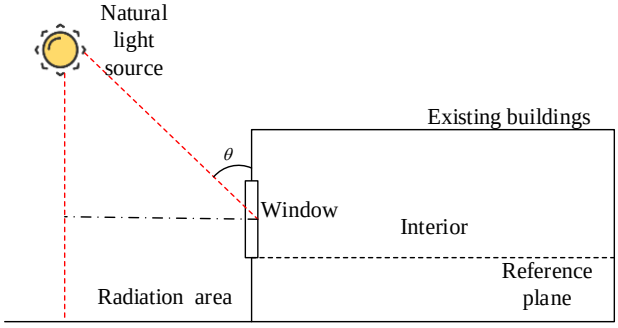


Figure 2: Schematic diagram of room lighting in the EB

In Figure 2, the WGR affects the lighting of the building. The average lighting coefficient C_{av} of a room is calculated in Equation (8).

$$\begin{cases} WGR = \dfrac{A_w}{A_{sum}} \\ C_{av} = \dfrac{A_w \tau\theta}{A_{sum}(1-\rho^2)} \end{cases} \tag{8}$$

In Equation (8), A_w and A_s denote window and room areas, respectively. θ denotes the elevation angle between the window center and the sun. τ denotes the

light transmission ratio. ρ denotes the reflectance ratio. The study uses Ecotect Analysis building simulation software to simulate the change of WGR and lighting coefficient. Supplemental artificial lighting is also required when natural lighting is insufficient. The total indoor illuminance value is composed of the natural lighting illuminance value and the lamp illuminance value. Finally, the final optimal WGR and illuminance values are determined according to the *Building Lighting Design Standard GB50033-2013*, and the number and power of luminaires are then determined. The heat balance of the envelope involves a number of aspects such as building EC, indoor thermal environment, and ES design of the building, and is mainly divided into the envelope modification of non-transparent and transparent envelopes [18-19]. The heat balance process of the two structures is shown in Figure 3.

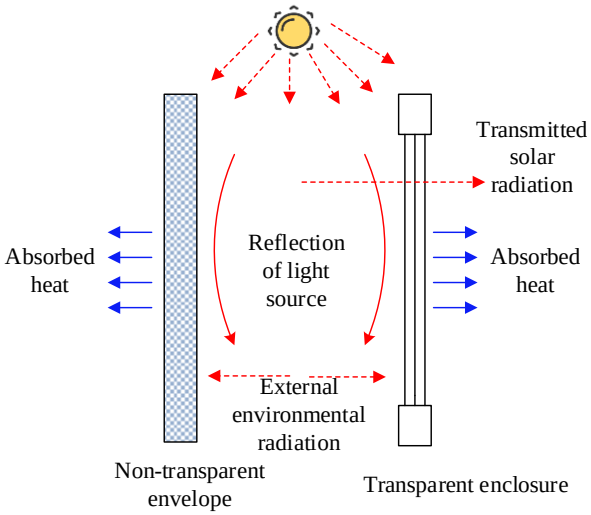


Figure 3: Schematic diagram of the thermal equilibrium of the different structures

In Figure 3, the difference between the two structures is mainly reflected in the fact that the transparent envelope enters a portion of the radiation affecting the heat load of the building. The heat transfer coefficient of the non-transparent enclosure and the indoor-outdoor temperature differential are the primary determinants of the thermal balance. In contrast, the heat balance of the transparent envelope is not only affected by the heat transfer coefficient, but also by the solar heat gain coefficient. In addition, the study introduces the ground source heat pump system to complete the renovation of the cooling and heating system. In summary, the study realizes that the multi-objective of building energy retrofit (BER) is mainly composed of EC and cost. The OF is shown in Equation (9).

$$\begin{cases} f_{\min} = f_{sum}\left(E_{ac}, E_i\right) = f_{ac}(x,K) + f_i(x) \\ g_{\min} = g(x) + g(k) \end{cases} \tag{9}$$

In Equation (9), $f_{\min}$ represents the minimum EC. E_{ac} represents the EC of the air conditioner. E_i represents the EC of lighting. $f_{ac}(x,K)$ represents the OF for optimizing air conditioning EC. $f_i(x)$ represents the OF for optimizing lighting EC. $g_{\min}$ represents the lowest renovation cost. K represents the heat transfer coefficient. x represents the window to ground ratio coefficient. $g(x)$ represents the cost of window renovation. $g(k)$ represents the renovation cost of the enclosure structure. To ensure the feasibility and rationality of the optimization plan, some additional constraints need to be met, as shown in Equation (10).

$$\begin{cases} K_{\min} \le K \le K_{\max} \\ g(x) + g(k) \le C_{budget} \end{cases} \tag{10}$$

In Equation (10), $K_{\min}$ and $K_{\max}$ represent the minimum and maximum values of the heat transfer coefficient. C_{budget} represents the budget.

3.2 Design of EC optimization solution algorithm based on improved MO-ChOA

Following the completion of the EC optimization model design for EBs, the study presents MOOA as a solution to the challenge of balancing several energies retrofit design goals. The EC MOO process proposed by the study is shown in Figure 4.

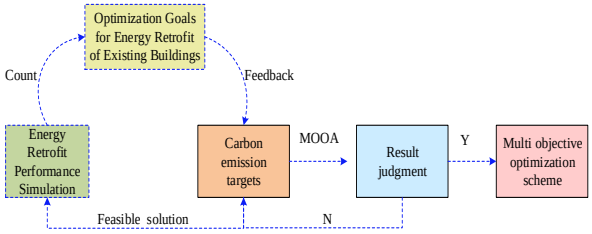


Figure 4: EC MOO process diagram

In Figure 4, the EC optimization process mainly includes three modules: determining the CE index, energy reform performance simulation, and optimization of CE index. The CE index and EC transformation OF are input into MOOA, and the search of the optimal solution is completed by using the CE index to search the solution space. The results of the energy reform performance simulation can be fed back into MOOA again, and the parameters of the algorithm can be adjusted according to the simulation results to complete the optimization of energy reform. The intelligent optimization algorithm used in the study is ChOA. ChOA is an intelligent optimization algorithm based on chimpanzee population behavior, which is characterized by fast convergence speed and high computational accuracy. ChOA mimics the strategies and behaviors of chimpanzee populations in terms of hunting, communication, and decision-making. It categorizes the population into attackers, interceptors, repellers and chasers. Attacking chimpanzees are responsible for the final attack behavior, usually representing the optimal solution, and their position vectors (PVs) are used to guide other chimpanzees to update their positions. The pursuers are responsible for collaborating with the repelling chimpanzees to bring them into attack range, guiding the search process towards the area of potential optimal solutions. Interceptor chimpanzees are responsible for blocking the escape route of prey and assisting other chimpanzees to surround the

prey during the hunting process [20-21]. In the ChOA, the chimpanzee's PV represents the solution vector, and the update process of the PV is also the optimization process of the solution vector. ChOA first randomly generates a certain number of solution vectors to model the initial state of the chimpanzee population and randomly updates the population position. As demonstrated by Equation (11), during the exploration phase, population members adjust their positions in accordance with the prey's position.

$$\begin{cases} X_{chimp}^{t+1} = X_{prey}^{t} - a \cdot d \\ d = \left| c \cdot X_{prey}^{t} - m \cdot X_{chimp}^{t} \right| \end{cases} \tag{11}$$

In Equation (11), X_{prey}^{t} represents the PV of the prey at time t. X_{chimp}^{t} and X_{chimp}^{t+1} represent the PV of individuals in the population at time t and time $t+1$, respectively. a denotes the coefficient vector, which is controlled by the convergence factor. d denotes the distance between the prey and the population. c denotes the influence factor of obstacles on the prey of the population. m denotes the chaotic mapping vector. During the exploitation phase, the chimpanzee population approaches the optimal solution by collaboratively updating their positions, which are shown in Equation (12).

$$X^{t+1} = \frac{X_1^{t+1} + X_2^{t+1} + X_3^{t+1} + X_4^{t+1}}{4} \tag{12}$$

In Equation (12), X_1^{t+1}, X_2^{t+1}, X_3^{t+1}, and X_4^{t+1} denote the updated PVs of the attacker, encircler, repeller, and pursuer, respectively. However, the initial population generation of traditional ChOA is too random, which leads to poor diversity and uniformity of the population distribution of the algorithm and affects the algorithm's traversal of the solution space [22-23]. In this regard, the study introduces the Hammersley sequence to help ChOA generate the initial population. The Hammersley sequence can generate a uniformly distributed point set in the multidimensional space. The improved calculation of the initial position of the population is shown in Equation (13).

$$X_k = l_b + H_k (u_b - l_b) \tag{13}$$

In Equation (13), u_b and l_b denote the upper and lower bounds of the positional variable interval. H_k denotes the Hammersley sequence. The Hammersley sequence is a low variance sequence that can generate uniformly distributed sample points. In ChOA, the Hammersley sequence first determines the number and dimension of the sample points. Second, the first d-1 primes are selected to generate Halton sequences in other dimensions. Then, Hammersley sequences are generated and the sequences for each dimension are combined to form complete Hammersley sequential sampling points. Finally, the sampling points are normalized to map them to actual variable ranges. Moreover, the generated sample points are used as the initial totality for the optimization of the algorithm. In addition, the traditional ChOA is prone to gradual convergence of population individuals to the local optimal solution as the number of iterations increases, resulting in the algorithm's difficulty in jumping out of this local optimal region. In this regard, the study introduces the somersault foraging heuristic strategy, which simulates the behavior of an animal looking for more food by somersaulting during foraging. The working mechanism is shown in Figure 5.

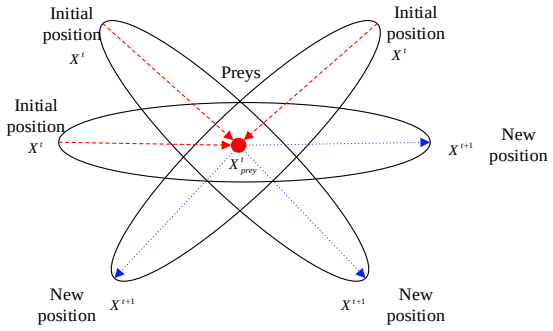


Figure 5: Mechanism of the somersault foraging heuristic working mechanism

As demonstrated in Figure 5, the somersault foraging strategy emulates the foraging behavior of organisms and updates the current solution symmetrically to the opposite side of the fulcrum. This allows the algorithm to explore the symmetric region of the current position, thereby increasing the diversity of the search and avoiding premature convergence of the population to the local optimal solution. In addition, the random number of the somersault foraging strategy also introduces randomness into the update process, making it possible to generate different new solutions with each update. The mathematical expression is given in Equation (14).

$$X^{t+1} = X^{t} + S(r_1 \cdot X_{prey}^{t} - r_2 \cdot X^{t}), if \quad i/N > r, r \in [0,1] \tag{14}$$

In Equation (14), S is the blanking factor. r denotes the random number. N is the maximum iterations' quantity. Finally, the study draws on NSGA II to design a multi-objective search strategy for improving ChOA, which mainly includes fast non-dominated sorting, congestion comparison operator, and elite strategy. The working mechanism is shown in Figure 6.

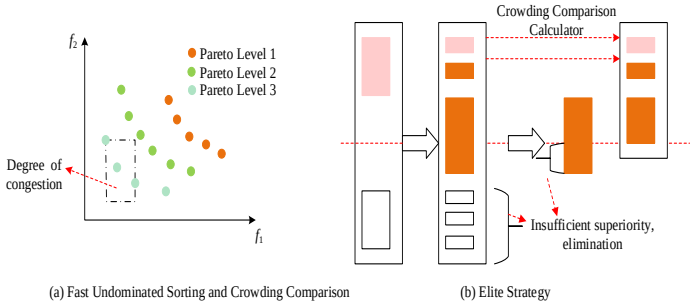


Figure 6: Multi-objective search strategy of improved MO-ChOA

In Figure 6, based on the population's fitness and each person's Pareto dominance connection, the enhanced MO-ChOA first completes the population hierarchy. Then, the comparison of individuals between the same tiers is performed. Finally, the optimal solution of the previous generation is retained and the multi-objective search is performed again for the new population of the next generation. The complexity analysis of the improved MO-CHOA is performed. It is assumed that the population size is N and the dimensionality of the decision variables is D. The computational complexity of the improved MO-CHOA to generate the initial population and the somersault foraging strategy is O(N·D). It is assumed that the computational complexity of the fitness function is O(F), and the complexity of each fitness computation is 0(N·F). The computational complexity of fast non-dominated sorting is $O(M \cdot N^2)$, where M represents the number of targets. The computational complexity of the crowding comparison operator is O(N·logN), and the computational complexity of the elite strategy is O(N·D). It is assumed the algorithm runs T iterations, the overall complexity of the improved MO-CHOA is $O[T(M \cdot N^2+N \cdot F+N \cdot \log N+N \cdot D)$.

4 Building energy retrofit MOOA performance testing and application effect analysis

To verify the effectiveness of the research-designed MOO model for BER and the improved MO-ChOA, the research launched performance testing and application analysis experiments.

4.1 MOOA performance testing

The comparative analysis of MOOA is unfolded first. The experiment is completed based on Windows 10 operating system; the central processor is Intel(R) Core (TM) i5-6300HQ with 16G of RAM. The image processor is GTX 3060 12G. The algorithm implementation language is Python3.8. Using hyper volume (HV) and inverse generation distance (IGD) as evaluation metrics. HV is employed to assess the extent to which the target space is encompassed by an approximate set. A higher value of HV signifies a more comprehensive coverage of the target space by the solution set, thereby indicating a higher quality of the solution. IGD measures the average distance from a reference point set to a solution set, with smaller values indicating that the solution set is closer to the true Pareto front and more evenly distributed. Sensitivity analysis is performed on the ZDT3 function by setting the maximum number of iterations to 100. The population size is set to 50, 100, and 150. The flip factor is set to 1, 3, and 5. The mutation probability is set to 0.01, 0.05, and 0.1, respectively. The results of the hyper-parameter sensitivity analysis of the improved MO-ChOA are shown in Table 2. In this table, a population size too small (50) leads to insufficient diversity and low HV, while a population size too large (150) increases the computational cost and slows down the convergence speed. When the population size is 100, convergence and diversity are balanced. If the flip factor is too small (1), the local search ability of the algorithm is weak, and if it is too large (5), it can lead to oscillations. If the mutation probability is too low (0.01), the algorithm will tend to get stuck in local optima. Furthermore, if it is too high (0.1), it will lead to a decrease in the convergence performance of the algorithm. When the population size is 100, the flip factor is 3, and the mutation probability is 0.05. Moreover, the improved MO-ChOA performs best in HV and IGD indicators, with values of 0.973 and 0.113, respectively. Therefore, in the study, the population size is set to 100, the flip factor is set to 3, and the mutation probability is set to 0.05.

Table 2: Hyper-parameter sensitivity analysis results

Hyper-parameters		HV	IGD
Population size	50	0.952	0.125
	100	0.973	0.113
	150	0.962	0.119
Flip factor	1	0.931	0.142
	3	0.973	0.113
	5	0.958	0.121
Mutation probability	0.01	0.962	0.129
	0.05	0.973	0.113
	0.10	0.955	0.124

The test functions include the dual-OF ZDT3, and the triple-OFs are DTLZ1 and DTLZ3. ZDT3 is a dual objective optimization test function with one global optimal solution and multiple local optimal solutions, which is commonly used to test the global convergence ability and local optimal solution avoidance ability of algorithms. DTLZ1 and DTLZ3 are multi-objective optimization test functions commonly used to test the performance of algorithms in multi-objective optimization problems. The performance of MO-CHOA on the test function before and after improvement is shown in Figure 7. Comparing Figure 7 (a) and (b), the improved MO-ChOA has a better solution on the ZDT3 function, and the solution value is more uniformly fit to the real frontier. The traditional MO-ChOA has poorer convergence of the solved values with more breakpoints. Comparing Figures 7(c) and (d), the improved MO-ChOA has better convergence on the DTLZ1 function, and the solved values are uniformly distributed on the frontiers, which can cover the true frontiers better. The solution deviation of the traditional MO-ChOA is obviously increased. Comparing with Figures 7(e) and (f), the improved MO-ChOA has better distributability on the DTLZ3 function, the overlap of the solution set gets reduced, the blank region of the true frontier is reduced, and the solution set distributability is superior. The results indicate that the improved MO-ChOA proposed by the study has a better multi-objective solution performance compared to the MO-ChOA. This is because the improved MO-ChOA uses a uniform initial population generated by Hammersley sequences, which can explore the search space more widely and avoid getting stuck in local optima. Meanwhile, the somersault foraging strategy further enhances the global search capability of the algorithm,

enabling it to more accurately find the global optimal solution. In addition, the multi-objective search strategy of NSGA II ensures the diversity and convergence of the solution, so that the solution values of the algorithm more closely match the real frontier. However, traditional MO-ChOA lacks these enhancements, resulting in poor convergence, multiple breakpoints, and poor distribution of the solution set.

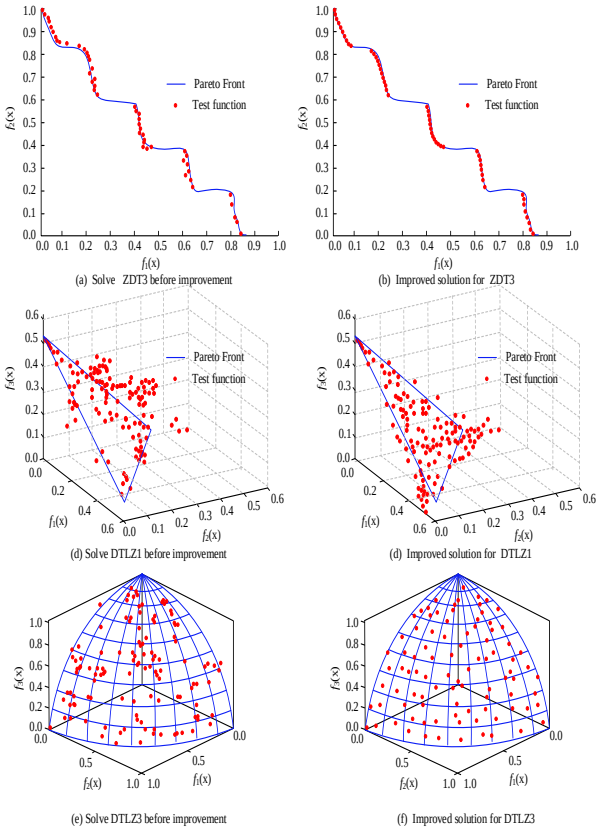


Figure 7: Performance of MO-ChOA on test functions before and after improvement

To verify the effectiveness of the proposed improvement strategy, ablation experiments are performed on the ZDT3 function. The original MO-ChOA, MO-ChOA with Hammersley sequences only, MO-ChOA with somersault foraging heuristic strategy only, MO-ChOA with NSGA II multi-objective search strategy only, and the complete algorithm are compared. The results of the ablation experiment are shown in Table 3. The full model performs best in terms of HV and IGD indicators. Next is the MO-ChOA, which only introduces the NSGA II multi-objective search strategy, with HV and IGD indicators of 0.765 ± 0.020 and 0.121 ± 0.008, respectively. This indicates that the NSGA II multi-objective search strategy contributes the most to improving the performance of MO-CHOA.

Table 3: Results of ablation experiment

Algorithm	HV	IGD
MO-ChOA	0.674±0.012	0.150±0.014
Hammersley+MO-ChOA	0.742±0.018	0.129±0.009
Heuristic strategy of somersault for foraging+MO-ChOA	0.721±0.015	0.134±0.012
NSGA II+MO-ChOA	0.765±0.020	0.121±0.008
Improved MO-CHOA	0.973±0.009	0.113±0.005

Compared with traditional MO-CHOA, MOPSO, non-dominated sorting genetic algorithm III (NSGA-III), and MOEA in reference [16]. The MOPSO algorithm simulates the foraging behavior of flocks of birds, finds the optimal solution through cooperation and competition among particles. Moreover, it introduces diversity search strategy and convergence search strategy to enhance the diversity and convergence of solutions. NSGA-III is a multi-objective algorithm based on evolutionary optimization, which uses genetic operations to optimize multiple conflicting objectives. Furthermore, it employs non-dominant sorting and crowding distance mechanisms to maintain population diversity and quality. MOEA is an algorithm that solves multi-objective optimization problems by simulating the process of biological evolution to simultaneously optimize multiple conflicting objectives. It uses dominance-based relationships and diversity preservation mechanisms to generate Pareto optimal solution sets. The comparison results of HV and IGD values for different algorithms are shown in Figure 8. In Figure 8(a), the IGD curve of the improved MO-ChOA is always at the lowest level, and the minimum value converges to 0.113, which is the closest distance to the real frontier, in line with the solution distribution in Figure 7. In contrast, the IGD values of the conventional MO-ChOA, MOPSO, NSGA-III, and MOEA converge at 0.353, 0.341, 0.301, and 0.295, respectively. Hammersley sequences with heuristic strategies to improve the convergence of the solution set. In Figure 8(b), the HV curve of the improved MO-ChOA grows the fastest and quickly converges to 0.973 at the late stage of iteration. The IGD values of the traditional MO-ChOA, MOPSO, NSGA-III, and MOEA converge to 0.826, 0.768, 0.751, and 0.820, respectively. The improved approach raises the convergence and diversity of the solution set while also improving the algorithm's overall performance.

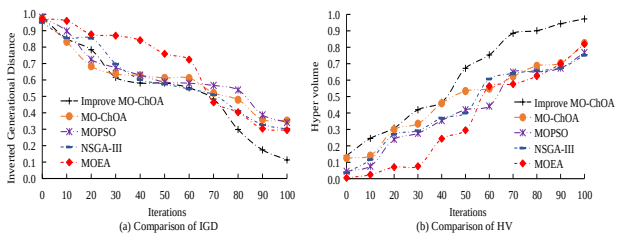


Figure 8: Comparison of HV and IGD values for different algorithms

The study used binary metrics (coverage metric, C-Metric) and knee driven dissimilarity (KD) as evaluation indicators. C-Metric is a performance metric used to measure the dominance relationship between two solution sets. It can evaluate which algorithm generates a solution set with better dominance relationship. KD is mainly used to measure the difference between the solution set and the

true knee point. The smaller its value, the closer the solution set is to the true knee point, and the better the algorithm performance. Adding more test functions to compare the different algorithms for solution set solving quality, Figure 9 displays the outcomes of the experiment. In Figure 9(a) and (b), the improved MO-ChOA designed by the study achieves the maximum C-metric on the test function ZDT3, which takes the value of 0.991. The C-metric is used to measure the degree of superiority of one solution set with respect to another solution set. That is, what percentage of solutions in one solution set is dominated by a particular solution in another solution set. Meanwhile, the minimum value of KD of 0.118 is obtained on the SCH test function. In Figure 9(c) and (d), the C-metric and KD values of other MOOA are worse than those of the improved MO-ChOA. The solution set of the algorithm can cover the real solution set more widely.

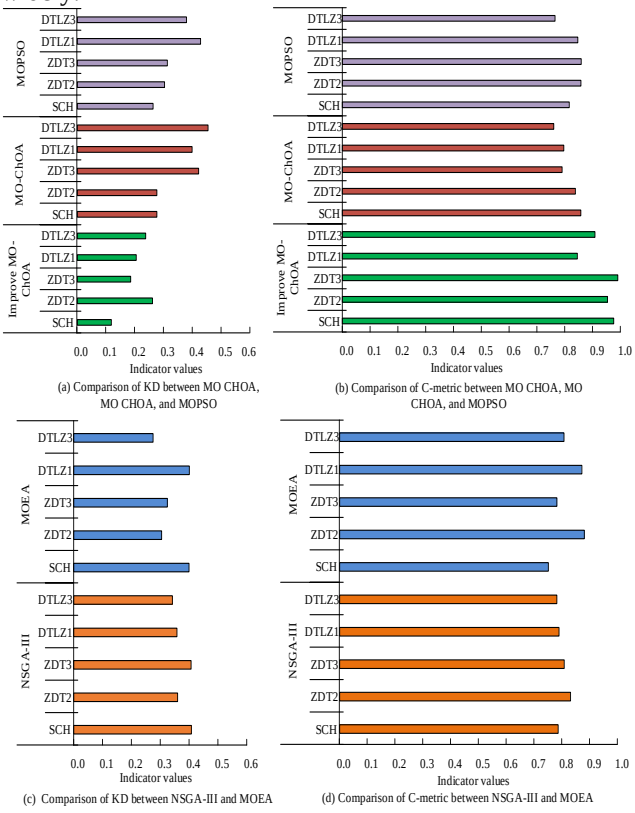


Figure 9: Comparison of C-metric and KD values for different algorithms

4.2 Effect of the application of building EC optimization

To verify the practical application effect of the proposed improved MO-ChOA, this study takes an EB in a city in China as an example to carry out EC optimization and renovation. The building is constructed in the 1990s, with a total area of about 5000 square meters and an annual CE target of 2.81 kg CO_2e/m^2. The energy utilization efficiency is relatively low, mainly relying on traditional central heating and air conditioning systems. The annual EC for cooling and heating is 27.81 kW·h/m^2, lighting EC is 6.95 kW·h/m^2, and other EC is 3.86 kW·h/m^2. The heat transfer coefficients of the interior and exterior walls of the building are 1.52W/(m^2 · k) and 0.92W/(m^2 · k), respectively. The heat transfer coefficient of the windows is 2.9W/(m^2·k), and the heat transfer coefficient of the doors is 1.35W/(m^2·k). A comprehensive optimization of the building's heating, air conditioning, lighting, and building envelope is planned. The solution quality of different algorithms in energy optimization is shown in Figure 10. In Figure 10(a), there is a significant difference in the solution spacing performance (SP) of different algorithms. Improved MO-ChOA has the smallest range of SP distribution. During 120 iterations, the minimum SP value is only 0.110. It can be concluded that the solutions inside the Pareto solution set of Improved MO-ChOA are denser and more diverse. While the minimum SP values of the other algorithms are on the level of 0.20 values. In Figure 10(b), the improved MO-ChOA performs well in uniformity performance (UP), with the average UP values all above the 0.80 taking level. Taken together, the studied design achieves the most widely distributed solution set in energy optimization.

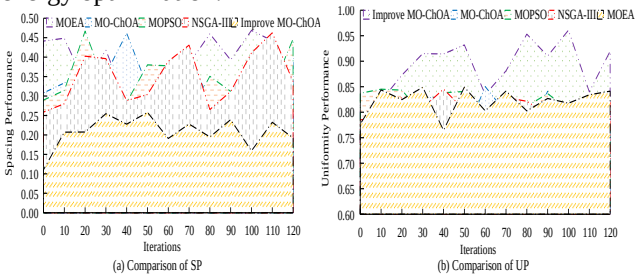


Figure 10: Comparison of SP and UP for different algorithms

The ESs rate (ESR) and CE savings ratio (CESR) of the building are compared after optimization with different algorithms. Figure 11 displays the outcomes of the experiment. Following energy optimization, the optimization scheme produced by the upgraded MO-ChOA in Figure 11(a) produces the maximum ESR fetch while the building is in operation. It is substantially superior to the solution schemes of other algorithms, fluctuating between 0.8 and 1.0. Following the building's refurbishment, the higher ESR readings indicate a notable decrease in EC and an increase in energy use efficiency. The retrofit effect of the other solutions is not significant and the reduction of EC is limited. In Figure 11(b), the optimization scheme obtained by the improved MO-ChOA obtained the highest CESR fetch value, fluctuating within the interval of 0.7-0.95. The method effectively reduces the carbon footprint of the building with less negative impact on the environment. China's *Energy Consumption Standards for Civil Buildings* has set binding and guiding indicators for building EC, and the energy-saving rate of the *Energy Efficiency Design Standards for Public Buildings* has increased from 65% to 72%. The improved MO-ChOA proposed in this study optimized the ESR of the building within the range of 0.8-1.0, which is much higher than the energy-saving rate

level of the energy-saving benchmark building. This indicates that the EC of the renovated building has been significantly reduced and the energy utilization efficiency has been improved, reaching the leading energy-saving level in China.

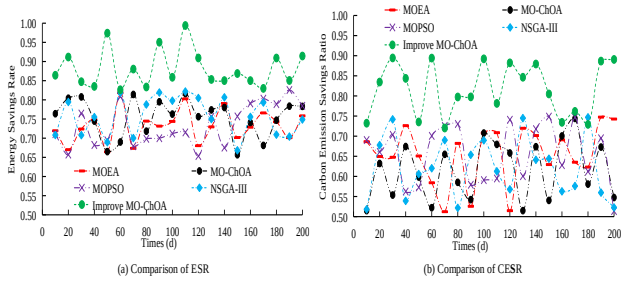


Figure 11: Comparison of ESR and CESR for different algorithms

The results of cost savings ratio (CSR) and return on investment (ROI) obtained by different algorithmic optimization schemes are shown in Figure 12. In Figure 12(a), the CSR curve of the improved MO-ChOA is the highest throughout the iterative optimization process, converging to the highest value of 0.790. It demonstrates that by increasing energy efficiency and lowering EC, the energy optimization plan lowers energy bills and other associated expenses. In Figure 12(b), the ROI of the Improved MO-ChOA is the highest, taking the value of 0.685. In summary, the scheme has significantly reduced the operating costs of the retrofitted building, with a high ROI and a high economic feasibility of the project.

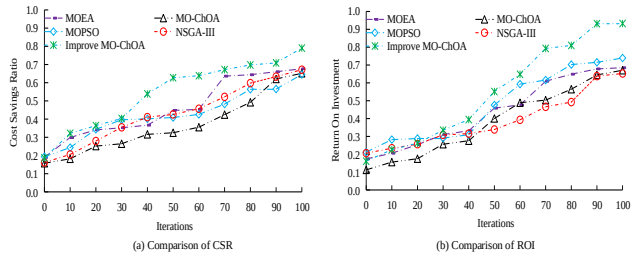


Figure 12: Comparison of CSR and ROI of different algorithms

5 Discussion

To optimize the operational EC of EBs, an improved MO-ChOA was proposed. The results showed that the convergence of the solution value of the improved MO-ChOA was better and could cover the real frontier better. The minimum value of IGD converged to 0.113 and the maximum value of HV converged to 0.973. IGD reflected the degree of closeness between the solution set and the true frontier, while HV was used to measure the coverage and overall quality of the solution set. The improved MO-ChOA focused more on the convergence of the solution set rather than the coverage area. Therefore, IGD had been significantly improved, but HV remained similar. The C-metric value of the improved MO-ChOA could reach 0.991 on the ZDT3 test function, and the KD value was only 0.118 on the SCH test function. Meanwhile, the minimum value of SP was only 0.110, and the UP values were all at the 0.80 fetch level. The investigated design achieved the most widely distributed solution set in energy optimization. During the operation of the building after energy optimization, the optimized solution obtained by improving MO-ChOA achieved the best ES rate and economic feasibility. The ESR took values fluctuating in the range of 0.8-1.0, the CESR took values fluctuating in the range of 0.7-0.95, the CSR converged to the highest value of 0.790, and the ROI took values as high as 0.685. The ESR, CESR, CSR, and ROI were all higher than the MOEA algorithm, indicating that the application effect was better than the MOEA algorithm proposed by Khan et al. [16]. In addition, the HV index of the method proposed by Zhao et al. [14] was 0.891, and the IGD index was 0.201. The HV index of the method proposed by Wei et al. [17] was 0.902 and the IGD index was 0.148, both of which performed worse than the algorithm in this study. It can be concluded that the improved MO-ChOA proposed by the research had a good multi-objective optimization performance. This was because the Hammersley sequence and the somersault strategy introduced in the study could generate an initial population by multi-dimensional uniform distribution. Moreover, the step size was controlled by the flip factor to make the algorithm jump out of local optima. The NSGA-II multi-objective search strategy could balance convergence and distribution. Compared to traditional MO-ChOA and NSGA-II algorithms, it was closer to the real frontier.

The proposed method can significantly reduce the EC of buildings during operation, thereby increasing the economic value and market competitiveness of buildings. Although in the case analysis, this study focuses on a specific building in a Chinese city and may not directly reflect the actual operation of other buildings. However, the proposed improved MO-ChOA has universality in principle and good scalability under different building types and climatic conditions. It can be combined with the automation control system of industrial buildings to optimize and schedule EC in the production process. To further improve the adaptability of the proposed method in different building types and climates, the optimization objectives and constraints can be redefined or adjusted.

In addition, the optimization and renovation of building EC also face various limitations in practical situations. In terms of data reliability, the collection of building EC data relies on various sensors and metering devices that may experience malfunctions or errors, resulting in inaccurate or incomplete data collection. In building renovation, the fact that the building is already in use and the construction space is limited, some large construction equipment cannot enter the site, which limits the choice of construction techniques and methods. Moreover, some EBs may have structural safety hazards due to their long construction history. To address the above issues, cross-validation and correction of anomalous data can be achieved by combining information from multiple data sources, such as fusing meter data with power sensor data to improve data accuracy and reliability. Prior to the initiation of construction renovations, a comprehensive survey of the building site should be conducted to formulate a

scientifically sound construction plan, methodically arrange the construction process, and develop appropriate reinforcement and repair strategies for addressing structural safety hazards.

6 Conclusion

To find the optimal retrofit solution through the optimization algorithm in the retrofit of EBs by simultaneously considering multiple objectives such as ES, economy, and comfort, the study implemented the solution of EC MOOA based on the improved MO-ChOA. The study provides effective technical means for the ES retrofit of EBs and promotes the development of the construction industry in the direction of low-carbon and sustainable development. However, the design of the study does not take into account the differences in structure, function, and usage patterns of different buildings, and the applicability of improving MO-ChOA needs to be further explored.

References

[1] Xue Q, Wang Z, Chen Q. Multi-objective optimization of building design for life cycle cost and CO2 emissions: A case study of a low-energy residential building in a severe cold climate. Building simulation. Building simulation, 2022, 15(1): 83-98, doi: 10.31449/inf.v49i16.7705.

[2] Khafaga D S, El-kenawy E S M, Alhussan A A, Eid M M. Forecasting energy consumption using a novel hybrid dipper throated optimization and stochastic fractal search algorithm. Intelligent Automation & Soft Computing, 2023, 37(2): 2117-2132, doi: 10.32604/iasc.2023.038811.

[3] Erebor EM, Ibem EO, Ezema IC, Sholanke A B. Building Professional Practices Knowledge and Integration Levels of Energy Efficiency Design Features in Selected Office Buildings in Abuja, Nigeria. Environment and Ecology Research, 2022, 10(3): 414-426, doi: DOI:10.13189/eer.2022.100309.

[4] Zhang T, Yang X. Energy-Saving Design of Smart City Buildings Based on Deep Learning Algorithms and Remote Sensing Image Scenes. Informatica, 2024, 48(19): 103-118, doi: 10.31449/inf.v48i19.6049.

[5] Fang Q, Zhang J, Xu Z, Lv Y, Pi J. Construction of Sustainable Building Performance Optimization Design Model Based on Sensitive Multi-Objective Decision Making. Informatica, 2024, 48(13): 81-96, doi: 10.31449/inf.v48i13.6013.

[6] Fallah A M, Ghafourian E, Shahzamani Sichani L, Ghafourian H, Arandian B, Nehdi M L. Novel neural network optimized by electrostatic discharge algorithm for modification of buildings energy performance. Sustainability, 2023, 15(4): 2884-2898, doi: 10.3390/su15042884.

[7] Tahmasebinia F, Lin L, Wu S, et al. Exploring the benefits and limitations of digital twin technology in building energy. Applied Sciences, 2023, 13(15): 8814-8851, doi: 10.3390/app13158814.

[8] Verma A, Prakash S, Kumar A. AI-based building management and information system with multi-agent topology for an energy-efficient building: towards occupant's comfort. IETE Journal of Research, 2023, 69(2): 1033-1044, doi: 10.1080/03772063.2020.1847701.

[9] Sana A F, Dibakar R, Bishwajit B. Genetic Algorithm and Grasshopper Optimization Algorithm with Metaoptimization and RL-Based Parameter Fine-Tuning and Their Comparison for Optimal Thermal Performance Analysis of Buildings in Tropical Climate. Journal of Computing in Civil Engineering, 2025, 39(2): 164-186, doi: 0.1061/JCCEE5.CPENG-6159.

[10] Wu Y. Design and analysis of energy efficient urban buildings based on BIM model. Journal of Computational Methods in Sciences and Engineering, 2024, 24(6): 3785-3798, doi: 10.1177/14727978241296348

[11] Khan F A, Ullah K, ur Rahman A, Anwar S. Energy optimization in smart urban buildings using bio-inspired ant colony optimization. Soft Computing, 2023, 27(2): 973-989, doi: 10.1007/s00500-022-07537-3.

[12] Xu F, Liu Q. Building energy consumption optimization method based on convolutional neural network and BIM. Alexandria Engineering Journal, 2023, 77(6): 407-417, doi: 10.1016/j.aej.2023.06.084.

[13] Gheouany S, Ouadi H, El Bakali S. Hybrid-integer algorithm for a multi-objective optimal home energy management system. Clean Energy, 2023, 7(2): 375-388, doi: 10.1093/ce/zkac082.

[14] Zhao N, Zhang J, Dong Y, Ding C. Multi-Objective Optimization and Sensitivity Analysis of Building Envelopes and Solar Panels Using Intelligent Algorithms. Buildings, 2024, 14(10): 3134-3151, doi: 10.3390/buildings14103134.

[15] Ding Z, Li J, Wang Z, Xiong Z. Multi-Objective Optimization of Building Envelope Retrofits Considering Future Climate Scenarios: An Integrated Approach Using Machine Learning and Climate Models Sustainability, 2024, 16(18): 8217-8233, doi: 10.3390/su16188217.

[16] Khan A M, Tariq M A, Rehman S K U, Saeed T, Alqahtani F K, Sherif M. BIM integration with XAI using LIME and MOO for automated green building energy performance analysis. Energies, 2024, 17(13): 3295-3316, doi: 10.3390/en17133295.

[17] Wei H, Jiao Y, Wang Z, Wang W, Zhang T. Optimal retrofitting scenarios of multi-objective energy-efficient historic building under different national goals integrating energy simulation, reduced order modelling and NSGA-II algorithm. Building Simulation. Tsinghua University Press, 2024, 17(6): 933-954, doi: 10.1007/s12273-024-1122-9.

[18] Teng J, Yin H. Carbon emission reduction in public buildings of extreme cold regions: A study on enclosure structure and HVAC system parameter optimization. Energy Science & Engineering, 2024, 12(6): 2676-2686, doi: 10.1002/ese3.1779.

[19] Alelwani R, Ahmad W M, Rezgui Y, Alshammari K. Optimising Energy Efficiency and Daylighting Performance for Designing Vernacular Architecture—A Case Study of Rawshan. Sustainability, 2025, 17(1): 315-315, doi: 10.3390/su17010315.

[20] Daoud M S, Shehab M, Abualigah L, Alshinwan, M., Elaziz M A, Shambour M K Y, Zitar R. A. Recent Advances of Chimp Optimization Algorithm: Variants and Applications. Journal of Bionic Engineering, 2023, 20(6): 2840-2862, doi: 10.1007/s42235-023-00427-w.

[21] Khishe M. Greedy opposition-based learning for chimp optimization algorithm. Artificial Intelligence Review, 2023, 56(8): 7633-7663, doi: 10.1007/s10462-022-10343-w.

[22] Qian L, Khishe M, Huang Y, Mirjalili, S. SEB-ChOA: an improved chimp optimization algorithm using spiral exploitation behavior. Neural Computing and Applications, 2024, 36(9): 4763-4786, doi: 10.1007/s00521-023-09236-y.

[23]Usman A M, Abdullah M K. An Assessment of Building Energy Consumption Characteristics Using Analytical Energy and Carbon Footprint Assessment Model. Green and Low-Carbon Economy, 2023, 1(1): 28-40, doi: 10.47852/bonviewGLCE3202545.

Real-Time Pore Blockage Quantification in Hydropower Drainage Pump Inlets via Canny Edge Detection and HSV Segmentation

Xiangqian Fu[1, 2], Linghua Hou[3, *], Qiqi Ding[4] and Long Xu[4]
[1] Hubei Technology Innovation Center for Smart Hydropower, Wuhan 430000, China
[2] School of Power and Mechanical Engineering, Wuhan University, Wuhan 430000, China
[3] China Yangtze Power Co., Ltd., Beijing 100032, China
[4] Three Gorges Jinshajiang Chuanyun Hydropower Development Co., LTD. Yongshan Xi Luodu Power Plant, Yongshan 657300, China
E-mail: houlinghua123@126.com
[*]Corresponding author

Keywords: drainage pump unit, edge extraction, HSV color detection, scale, small pore filter

Received: April 25, 2025

As a core component of the diversion tunnel in hydropower stations, the drainage pump is prone to blockage in its inlet filter due to severe water calcification inside the tunnel. In this study, the present study proposes a computer-vision-based detection method to assess the blockage degree of the inlet filter's pores. Specifically, the outline of the small pores in the inlet screen of the drainage pump unit was captured using the Canny edge detection algorithm. Then the pore and non-pore areas of the filter were distinguished with the HSV color model. Finally, the degree of blockage was determined by calculating the proportion of pore areas within the filter screen. When applied to a real hydropower station using 12 images sampled at 15-day intervals, this developed computer vision technology achieved 13.6% average error against manual annotations and demonstrated real-time processing (<1s/image on 4-core/8GB edge devices) effective detection for the pores blockage degree of the drainage pump's inlet filter screen, by quantitatively assessing the blockage degree, it provides critical metrics that enable predictive maintenance scheduling and performance evaluation, thereby ensuring reliable hydraulic performance and minimizing downtime in the hydropower station's drainage system.

Povzetek: Predstavljena je hibridno računalniško-vidna metoda s Canny robnim zaznavanjem in HSV segmentacijo za sprotno kvantifikacijo zamašitve por na črpalkah, ki omogoča prediktivno vzdrževanje hidroenergetskih sistemov.

1 Introduction

As one of the most essential parts of drainage systems, drainage pump unit is widely used in hydropower stations, mines, rivers and other fields for water management [1]. Because of the special landform and the low terrain height in southwest region of China, the powerhouse for large hydropower station is mostly located in the riverside mountains. To deal with the accidents such as water seepage from the mountain and heavy rainfall, etc., which can lead to flooding of the powerhouse, the hydropower station is often equipped with a comprehensive drainage system to discharge the accumulated water. Usually, the seepage water from the mountain is diverted to the catchment area through the open channel, and then the water is discharged through the drainage pump unit. But drainage pump unit may be damaged during this process because of inhalation of debris from the mountain or water calcification. Water calcification stems from calcium carbonate (CaCO3) precipitation in groundwater, driven by temperature shifts and pH changes. Southwest China's karst geology exacerbates mineral dissolution, while debris like silt combines with scales to form dense fouling layers, accelerating pore blockage. Such blockages significantly increase hydraulic resistance, forcing pumps to overwork, elevating energy consumption and overheating risks. Prolonged operation may cause motor burnout, while partial drainage raises flood risks during storms. Nowadays, people add a filter screen to the inlet channel of the drainage pump unit to avoid the damage but the small pores of the filter screen are still easily blocked by calcified debris, resulting in reduced hydraulic performance of the drainage pump unit and even leading to motor damage. Although the electrical submersible pump has been in use for many years, there are still many problems which can be summarized into three main categories: electrical problems, mechanical problems and operational problems. For operational problems, fluid with different nature would cause different problems with multi-phase fluids (liquids, solids and gases) causing more serious problems than others [2].

In drainage systems, traditional blockage detection methods such as pressure differential monitoring, vibration analysis, and acoustic emission sensing often lack spatial resolution, making it challenging to quantify pore-level obstructions. Recent advancements in computer vision have enabled non-invasive inspection techniques. For instance, an AIoT-based real-time visual detection solution

for culvert blockages integrates artificial intelligence, computer vision, and edge computing to effectively assess blockage status. However, these methods often rely on extensive training data or perform inadequately under dynamic environmental conditions, such as fluctuating illumination in enclosed hydropower tunnels. Notably, research addressing calcification-induced pore blockages in hydropower drainage systems is limited. One study applied impedance spectroscopy and tomography to monitor calcite deposits in drainage pipes, providing insights into the formation and detection of such blockages [3]. However, integrating real-time visual monitoring with predictive models remains underexplored. Traditional RGB-based vision systems often misclassify calcified deposits due to color similarities with filter materials. In contrast, utilizing alternative color spaces and edge detection algorithms has shown potential in distinguishing mineral deposits from metallic surfaces, yet their application in hydropower pump filters has not been fully investigated [4].

Moreover, existing methods-including manual inspections, predictive models, and traditional computer vision-face three key limitations: (1) labor-intensive periodic cleaning cannot address sudden blockages; (2) data-driven models require extensive datasets and calibration, hindering deployment in variable environments; and (3) conventional vision systems struggle with lighting fluctuations and color ambiguities between scale deposits and pores. To overcome these gaps, we propose a hybrid computer vision framework combining Canny edge detection and HSV segmentation. This approach bypasses the dependency on training data by directly quantifying pore geometry and color contrast, ensuring robustness in enclosed tunnels with dynamic conditions. Real-time porosity monitoring enables predictive maintenance, surpassing the reactive nature of traditional manual strategies. To contextualize our approach, Table 1 compares mainstream blockage detection methods across five critical dimensions:

Table 1: Comparative analysis of blockage detection methods

Method	Required Data	Resolution	Robustness	Cost	Deployment Context
Pressure differential	Sensor readings	Low	Flow-dependent	Medium	Pipe systems
Impedance tomography	Electrochemical signals	Medium	Contact-required	High	Lab/small pipes
AIoT vision	Labeled image datasets	High	Lighting-sensitive	High	Open culverts
RGB analysis	RGB images	Medium	Color-ambiguous	Low	Controlled lighting
Our method	Single unlabeled image	High	Lighting-adaptive	Low	Enclosed tunnels

Currently, different preventive and control measures are adopted to solve the problem of flow channel blockage for actual working places. For example, for irrigation canals, pumping stations, inlet sluice, and other hydraulic structures, Singal et al. [5] added garbage racks in front of the water intakes to prevent floating and sinking objects from causing damage or running problems to the turbines and other components. Hribernik [6] used hydropower station flow and waste rack head losses to build rack clogging models to distinguish head losses caused by debris and rack structures. Cui et al. [7] found that groundwater along rivers and lakes showed a periodic change and then did relevant experiments to evaluate the effect of water level changes on the migratory depositional characteristics of particles in porous media. Tao et al. [8] designed a pre-pumped micro pressure filter to filter sediment particles through a stainless-steel filter and developed a prediction model for filter head loss. Liu et al. [9] developed a novel simulation model which coupled hydrodynamics and hydrochemistry for calculating the deposition rate of CaCO3 fouling in the pipeline surface. Ogie et al. [10] proposed a method for measuring and ranking the vulnerability of pumping stations to trash blockage, which makes it possible to point out the pumping stations that are most vulnerable to trash blockage. For instance, Rana et al. [11] found in the ultrafiltration analysis and comparison of produced water in the upstream oil and gas operations that the best membrane for treating produced water was zirconium oxide. Fujita et al. [12] tested the Fe removal capacity of three green filtration materials in deployed closed- and open-loop thermal systems using groundwater geothermal energy (GGE) to reduce clogging problems in open-loop ground source heat pump systems. Nallakukkala et al. [13] proposed three techniques for controlling scale formation: removal or decrease of Scale-forming species, water treatment for acidification of feed and degassing CO2 generated from carbonates, and chemical treatment for descaling using acids or chelating agents.

The above research focuses on analyzing the flow pattern, the clogging mechanism and flow channel structure design to cope with flow channel blockage. Although few studies have constructed predictive models for the blockage degree of flow channel, these methods are all built based on being able to accurately extract blockage-related operation and environment variables, as well as

sufficient high-quality positive and negative blockage training sample data sets, which makes the modal hard to be applied in special work places such as the diversion tunnel's catchment area of hydropower station, where the water calcification is unavoidable and the optimization of the inlet filter structure of the pump is limited. Therefore, it is necessary to detect the blockage in these work places in advance. In this study, the present study proposes a computer-vision-based detection method for the blockage degree in the inlet filter's pores. High-definition cameras were installed to collect images of drainage pump units periodically, and then Canny edge detection algorithm was used to extract the pores outline of the inlet filter screen in this image. Finally, the HSV color model was used to separate the pore space from the non-pores space (including limescale and filter screen body). The proportion of pore areas in the filter screen was calculated, and the remaining porosity ratio threshold alarm was set to notify the relevant personnel in advance to carry out the filter screen descaling treatment.

To reliably quantify pore blockage in hydropower drainage pumps under dynamic tunnel conditions, our approach combines:

(1) Canny edge detection for robust pore localization;

(2) HSV segmentation to distinguish calcified deposits;

(3) Pore-area calculation for blockage quantification;

Compared with the manually annotated results, the accuracy of the algorithm model was verified through error comparison.

Main Contributions:

Hybrid vision framework: Combined Canny edge detection and HSV segmentation to quantify pore blockage, overcoming limitations of manual inspections and data-driven models.

Training-free real-time monitoring: Enabled robust pore detection in dynamic tunnel environments without requiring large datasets.

Practical validation in hydropower stations: Tracked pore-area reduction over time, supporting predictive maintenance and downtime reduction.

Enhanced robustness via HSV: Outperformed RGB methods in distinguishing pores from calcified deposits under variable lighting.

1 Methods

2.1 Canny edge detection

As a general edge detection algorithm [14], the Canny algorithm can detect the edge in a very robust way. It can extract the image edges without disturbing the image features and separate the noise from the input image. It is one of the most practical and functional detection operators available. The basic idea of the Canny algorithm is to use a two-dimensional Gaussian filter template to smooth the image, then use the derivative algorithm to find the grayscale gradient of the pixel in the x and y direction and further calculates the gradient's magnitude. Then it carries out the 'non-maximum suppression' to find out the maximum value of the pixel in the gradient direction and determine whether the central pixel is tentatively set as an edge point. Finally, it conducted the double-threshold selection to make a final judgment on the tentative edge points obtained in the previous step and select the final edge.

2.1.1 Gaussian filter

In the data acquisition stage, the digital image acquired by the sensor is usually affected by noise, and image denoising can effectively preserve the image edge and get better expansion in the flat area [15]. Gaussian filters are good at removing normally distributed noise, which is usually found in cameras images produced through its digitization process. It is a natural phenomenon caused by the light-reflecting nature of the camera itself and the sensitivity of the optical sensor [16].

First, the image noise is removed by a filter through convolving with the image. The noise in the image corresponds to the high-frequency part of the image so that the noise can be removed in the frequency domain. However, edge detection and filtering usually conflict with each other because the filtered image is pixel-weighted averaged according to the template, which reduces noise but blurs the image edge simultaneously, resulting in increasing uncertainty of edge position. Hence, compromise between the two is necessary. The kernel determined by Gaussian function can resist noise interference while not affect the sensitivity of edge detection too much. That's the reason why we choose Gaussian filter to remove the noise in the digital image. The 2D Gaussian kernel is defined as follows:

$$K = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (1)$$

2.1.2 Intensity and direction of pixel gradient

The image edge can be in any direction. The gradient (G) magnitude and the direction (θ) of each pixel point can be obtained by calculating the first derivative of the horizontal and vertical directions of each pixel point. The G and θ can be calculated as follows:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2)$$

$$\theta = arctan\left(\frac{Gy}{Gx}\right) \quad (3)$$

The first derivative of the horizontal and vertical gradients can be calculated by operator filtering such as Roberts, Prewitt, and Sobel. Among them, the Sobel operator is particularly useful for image processing and edge computing [17]. Therefore, this paper used the Sobel operator instead of the traditional 2*2 template to reduce the generation of isolated edge points and pseudo-edge points. The Sobel operator gave more weight on the center coefficient, which can provide smoother images. The gradient operator of Sobel along the x and y directions are calculated as follows:

$$g_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (4)$$

$$g_y = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (5)$$

2.1.3 Non-Maximum Suppression

The Non-maximum Suppression (NMS) was first created in edge detection field to improve a rough edge response to a thin line [18]. This oriented NMS operates in a one-dimensional (1-D) range perpendicular to the edge. Kitchen and Rosenfeld [19] extended this concept to isotropic NMS to locate two-dimensional (2-D) characteristic points of an image, which are selected as local maxima of an "Angle" image over a neighborhood.

Non-maximum suppression searches for local gradient maxima of pixel points along the direction of the pixel gradient and sets the gray value of pixel points corresponding to local non-maxima to 0. Its function is to remove false edges and refine and precisely locate the image edges.

2.1.4 Dual Threshold Selection

By setting two thresholds TH1 and TH2 (TH1 > TH2), double threshold selection filters the local maximum pixel x generated in non-maximum suppression to realize noise reduction and edge continuity. In this process, the pixel points are filtered into three sets: the strong edge set S_s, the weak edge set S_w, and the abandoned edge set S_d, defined as follows:

$$\{S_w | TH1 > x > TH2\} \quad (6)$$

$$\{S_s | x > TH1\} \quad (7)$$

$$\{S_d | TH2 > x\} \quad (8)$$

When its gradient is larger than TH1, the pixel point is classified as "definite edge" pixel and are retained. Pixel points with gradients smaller than TH2 are regarded not to be edges and are discarded. For those Pixel points whose gradient is between TH1 and TH2, they are considered to be part of the edge only if they are connected to a "definite edge" pixel point, otherwise they are discarded.

2.2 HSV color space

HSV (Hue, Saturation, Value) is created by graphic designers mimicking the process of artists creating colors [20-21]. H (hue) represents the hue of a pixel point, and is measured by Angle with value ranges from 0° to 360°. S (saturation) represents the saturation of the pixel point, and its saturation value ranges from 0 to 255. V (value) represents the brightness of the pixel point, reflecting the colorless concept of intensity. The value of V ranges between 0.0~1.0, with 0.0 representing black and 1.0 denoting white.

HSV aggregations are closer to human vision (as they reached highest F1 values) than the RGB aggregations, and HSV aggregations have lower standard deviations than RGBs, suggesting that the HSV methods are more robust [22]. As the HSV model is evolved from the RGB cube model, so the HSV color space can be converted from the RGB color space. Based on the principle of its conversion, the HSV color space can be calculated through the following formula:

$$H = \begin{cases} arccos\left(\dfrac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}}\right), B \le G \\ 2\pi - arccos\left(\dfrac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}}\right), B > G \end{cases} \quad (9)$$

$$S = \frac{max(R,G,B) - min(R,G,B)}{max(R,G,B)} \quad (10)$$

$$V = \frac{max(R,G,B)}{255} \quad (11)$$

where, *R*, *G*, and *B* are the three components of the RGB model, respectively.

2 Results

Applied this developed computer vision technology to the drainage pump of a real hydropower station. The pump was installed vertically with a filter screen at the inlet for filtering debris and calcified matter. A camera was installed on the wall of the catchment area closest to the pump. The pore blockage detection of the inlet filter included three steps: filter profile extraction, HSV separation of filter's small pores and pore area calculation.

2.1 Inlet filter profile extraction

The extraction steps of the inlet filter are as follows:

a. Marking the outline of inlet filter. The edge outline of the inlet filter was manually marked with Anylabeling tool to get the (x, y) coordinate of these pixel points. The image taken from the camera is shown in Fig. 1a.

b. Transforming into mask area. Read the profile coordinates of the inlet filter in json file and transform them into masked regions based on OpenCV as shown in Fig. 1b.

c. Filling the mask area. Actual pictures of the filter were used to fill the mask area obtained, and the area out of the profile was further filled with white

color. Finally, the complete outline picture of the filter screen was obtained, as shown in Fig. 1c.

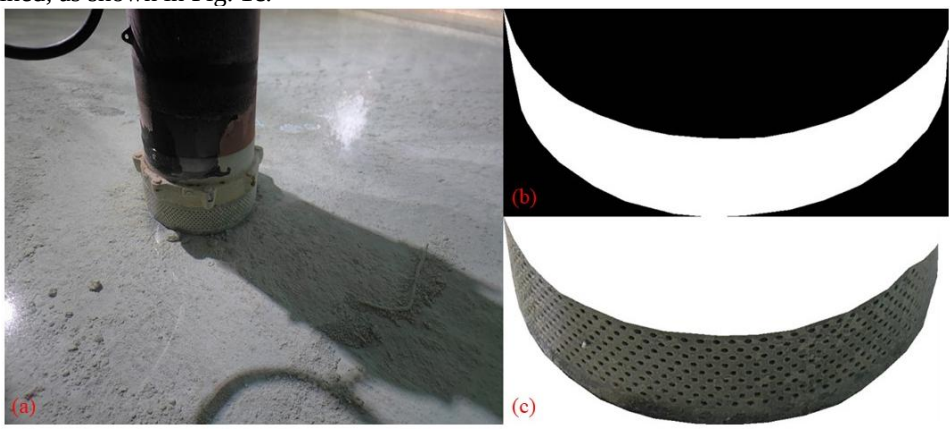


Figure 1: Extraction process of intake filter

3.2 Identification of filter's pores with HSV

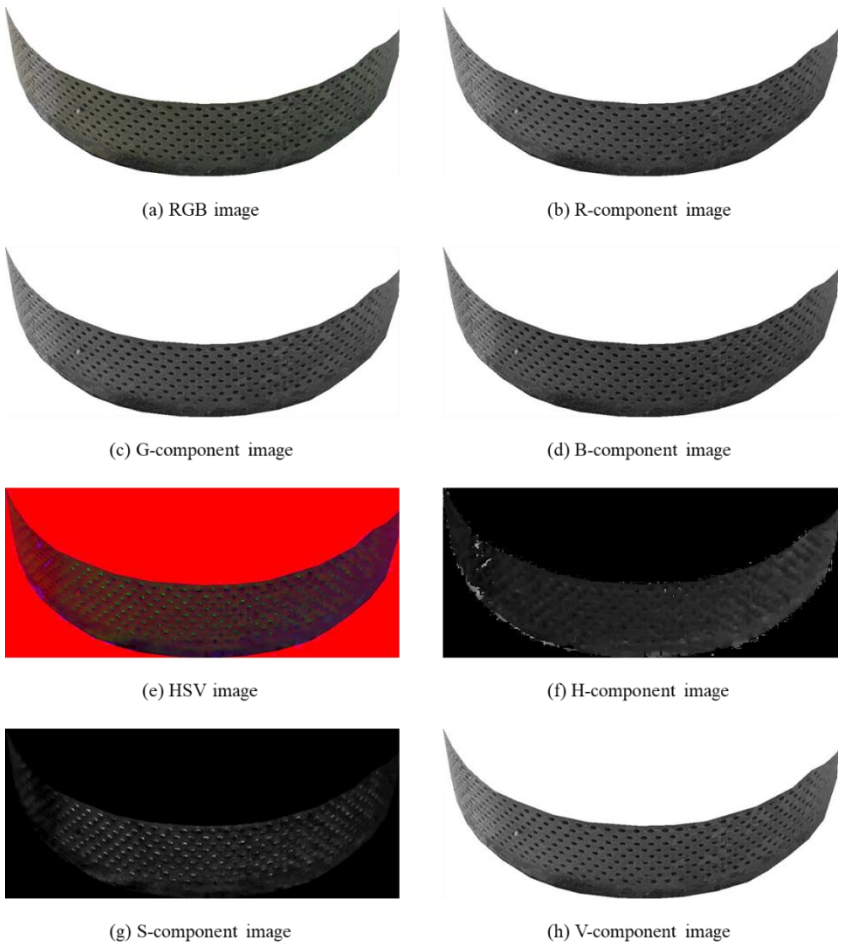


Figure 2: HSV result of RGB conversion of intake filter

The RGB component of the inlet filter' image was obtained first, and then they were converted into HSV component according to the Eq. (9-10). During the conversion process, the thresholds of hue, saturation, and brightness needed to be adjusted to enhance the filter screen pores' characteristics. The conversion results are shown in Fig. 2.

As can be seen from Fig. 2, the RGB component diagram cannot clearly distinguish the porous area of the inlet screen from other areas while the S-component map of HSV showed clearer pore areas. Based on the HSV color, the porous regions of the filter screen were extracted, and the outline of filter's pores was further obtained through color region mask. The results are shown in Fig. 3.

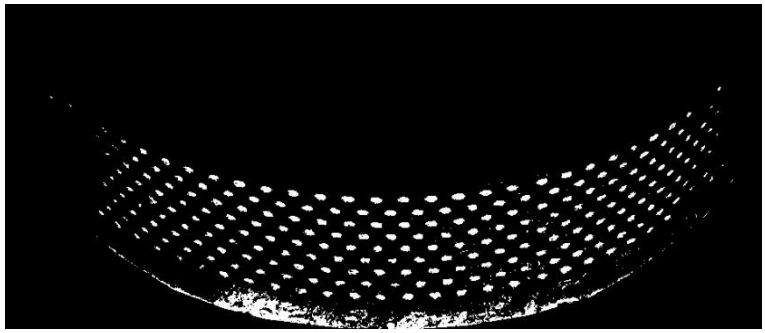
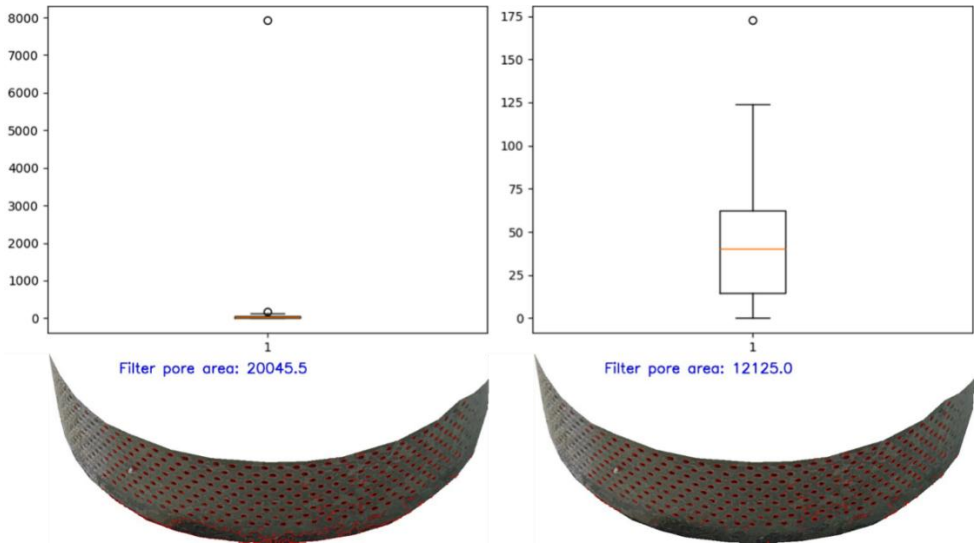


Figure 3: Mask image of filter pores

3.3 Pore area calculation

Based on the pore mask of the filter screen obtained through HSV color model, the edge of the pores was further detected by Canny algorithm. After grayscale treatment of the image, Canny operator was used to perform Gaussian filtering, gradient calculation, non-maximum suppression processing and double threshold selection. As there were many irregular lines in the picture, the key of the algorithm was the selection of the double threshold. In this study, the trackbar of OpenCV was used to dynamically adjust the dual thresholds and select the appropriate threshold combination pattern.

As there were some interference areas with similar color to the inlet screen pores, so it is necessary to further evaluate and select the pore areas obtained. In this paper, the box plot was adopted for screening. The anomalies of pore areas in the box plot were shown in Fig. 4. The y axis of the box plot represented the pixel area size, and the x axis represented the target pore area groups.



(a) (b)

Figure 4: Abnormal area box plot

As shown in Fig. 4a, there were large values in the pore area aggregation, resulting in the bad performance of the boxplot. Moreover, these outliers belonged to the non-porous area of the inlet filter screen, so they were supposed to be excluded. Fig. 4(b) showed the box plot with an area less than 1000 after excluding the outliers. The abnormal values in this box plot were the pore-connected regions, which should not be removed. There were also some small area values in the box plot which belonged to the non-pore areas as shown in Fig. 4, so further screening was required.

The values indicating small pore area mostly appeared in the lower quartile of the box plot, and the lower quartile was calculated to be 14.5. Therefore, the pore area value less than or equal to 14.5 was extracted. The diagram of screen result for the specific pore area is shown in Fig. 5.

Figure 5: Area <= 14.5 pore set

In Fig. 5, the red dot circled in green represented the non-porous area, indicating a lot of the non-porous regions having a pore area less than or equal to 14.5. Therefore, it is reasonable to use the lower quartile value for screening.

Based on the area screening results of the box plot, the S-component distance between pixels in the image was further calculated, and all pixels can be classified into two types: the pixel corresponding to the filter screen pores (the white highlights in the image) and other black regions. By accumulating the area values of the first type of pixel points, the pixel area of the filter screen pores can be obtained as 11949.5 (Fig. 6).

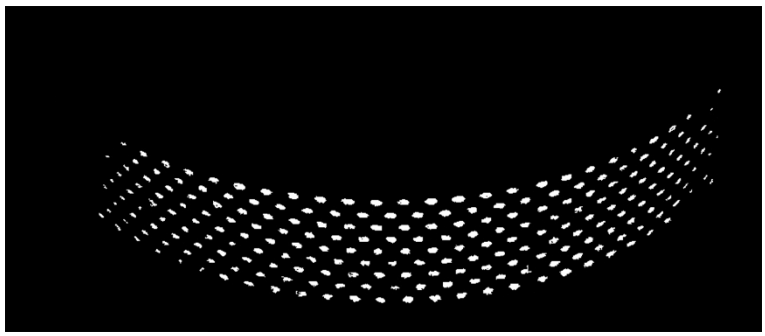


Figure 6: Exclusion of abnormal areas

To quantify detection accuracy, a benchmark was established through manual annotation. Three images with maximal temporal spans were processed as follows:

(1) Automated pore segmentation via ImageJ v1.54g Flood Fill (8-connected) failed due to region fragmentation caused by discontinuous intensity gradients;

(2) Freehand Selections were subsequently employed to trace irregular pore boundaries;

(3) Pixel areas were computed automatically using Analyze > Measure. An annotation example is shown in Fig. 7.

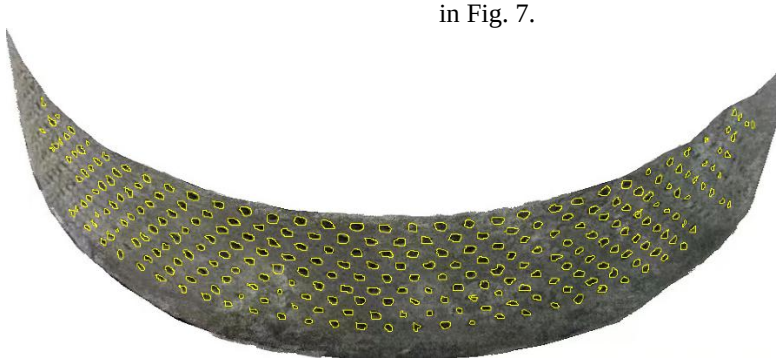


Figure 7: Manual pore annotation using ImageJ Freehand Selections

Comparing algorithmic outputs against manual benchmarks, the mean error rate of the HSV-S component reached 13.6% (Table 2), confirming detection reliability.

Table 2: Error rates (%) of pore area between manual annotation and algorithmic detection across three sample images

Sample Date	HSV-H	HSV-S	HSV-V	RGB-R	RGB-G	RGB-B
2024-01-07	24.7	12.3	19.8	18.6	19.1	19
2024-04-06	27.1	14.5	23.6	24.1	23.8	23.8
2024-06-20	29.4	13.9	25.1	24.2	22.1	23.9
Mean Error	27.1	13.6	22.8	22.3	21.7	22.2

3.4 Time variation of pore area

To verify the practicability of this study, the changes of the inlet filter over time were tracked. Real pictures of the pump unit were taken every 15 days within half a year, and the area of the inlet filter was extracted. The specific changes of the inlet filter over time are shown in Fig. 8.

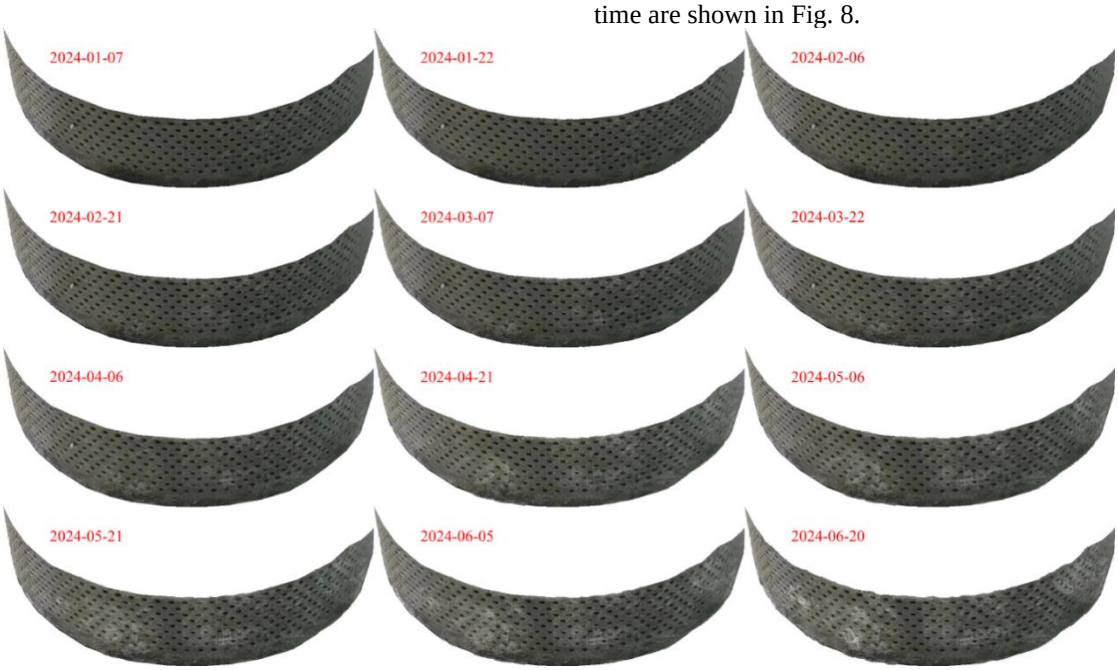


Figure 8: Changes in intake filter over time

Extract the small pore areas of the inlet filter screen in each time and figure out the small pore area. In this paper, the extraction result in different time periods was shown in the form of mask pictures to see its changes more clearly. The specific changes of the pore areas of the inlet filter screen over time are shown in Fig. 9.

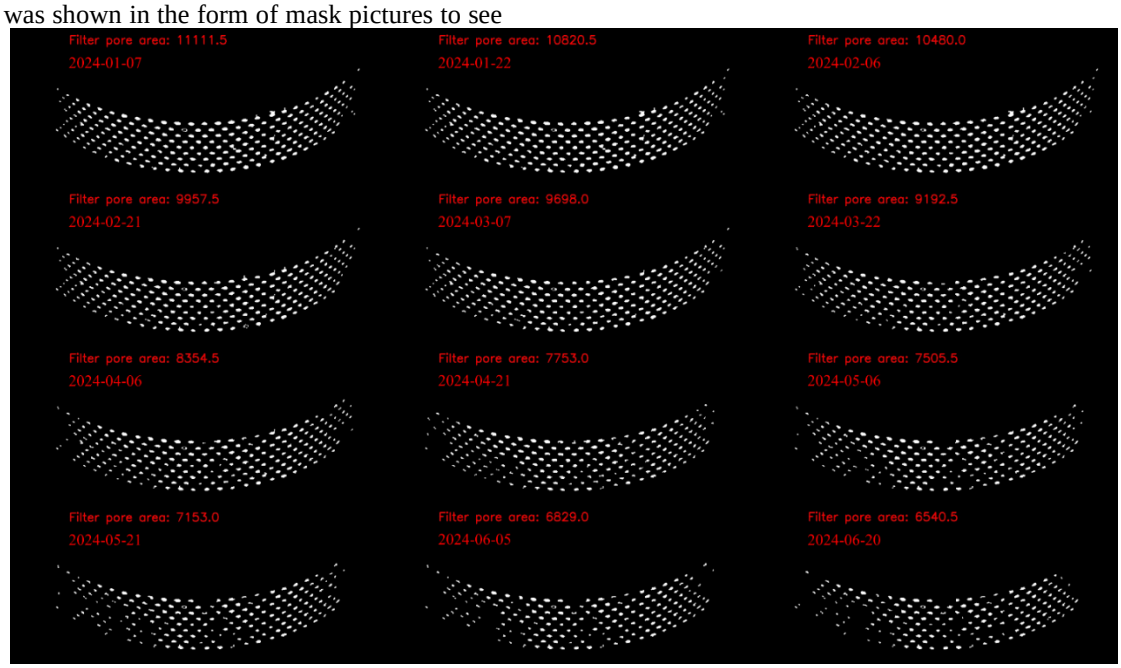


Figure 9: Intake filter pore areas over time

In Fig. 9, higher degree of blockage of the small pores were seen in the lower region of the intake filter screen than that of the upper part, which was because the impurities in the water were gathered in the lower region after precipitation, resulting in the easy blockage of the pores below the intake filter screen. The change of pore area of the filter over time can be shown clearly by plotting the curve of pore area versus time. The specific inlet filter screen pore area with time change curve is shown in Fig. 10. It's clear that the pore area of the inlet filter screen was getting smaller and smaller over time leading to the pore blockage of the inlet filter later, which would affect the running of the pump unit. This study realizes the real-time monitoring of the inlet filter screen's pore areas, providing a basis for the reasonable arrangement of cleaning work, and timely reflecting the serious blockage of the pump unit.

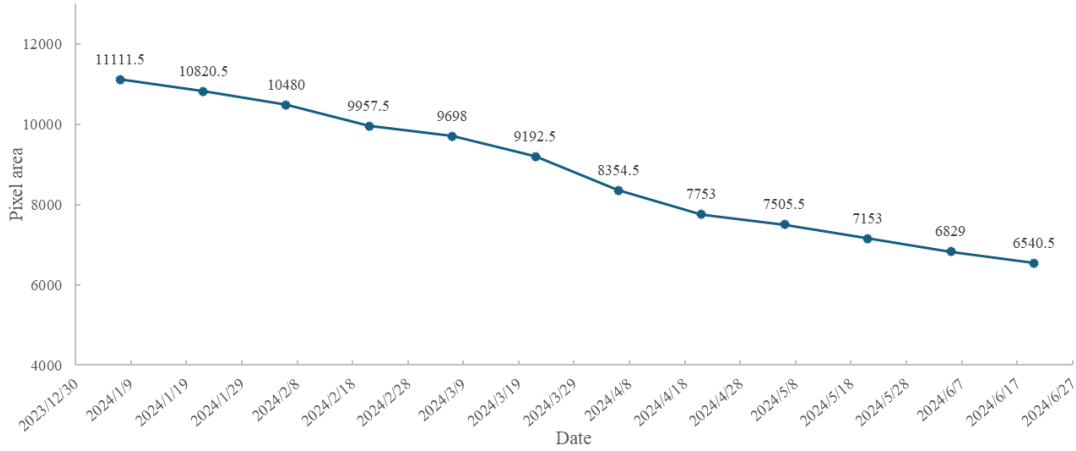


Figure 10: Intake filter pore area versus time curve

3 Discussion

This study aimed to provide a set of low-cost practical scheme for monitoring the blockage of the inlet filter screen of the drainage pump unit of the diversion tunnel of the hydropower station. This study comprehensively analyzed the previous research and practice solutions, such as the use of garbage racks and the improvement of filtering devices, but they were still needed to be cleaned regularly to prevent blockage. This study results also have some reference value for similar engineering applications. First, the computer vision technology was applied to optimize the manual regular cleaning plan. As the water calcification and the remote location of diversion tunnel which was far away from the management area, the regular cleaning program was still used. This program resulted in unable to clean the sudden blockage problem timely and multiple failures of the pump unit. Compared with regular cleaning programs, computer vision technology can monitor abnormal conditions timely such as sudden blockage and accelerated scaling. Moreover, it offers advantages including simple operation, low cost, and attractive performance [23]. Although the severe calcification of water in diversion tunnel, the calcified matter would precipitate to the bottom, so the water surface above is relatively clear. Besides, the diversion tunnel is enclosed space and is illuminated by searchlights throughout the year, which ensures that the brightness of the light is constant and the camera provides consistent imaging the inlet filter under the water. The camera is waterproof and dustproof with IP66 and above to prevent water from flooding the camera.

For the recognition object of computer vision, instead of direct recognition of the scaling region of inlet filter, we choose to recognize the small pore region of inlet filter. While the former enables intuitive assessment of blockage scale on the inlet filter, it achieves substantially reduced computational complexity, this does not effectively represent the blockage degree of the filter screen in drainage pump because serious scaling in other non-porous areas of the inlet filter does not mean that the pump unit is in a blocked state. If the staff only cleans the blockage of the small pore of the water inlet filter, the former would still determine that the pump unit is in a blocked state, leading to the erroneous diagnostics. In our study, we combine Canny algorithm with HSV color model to extract a clear image of the porous area of the inlet filter screen. Canny has excellent performance in edge detection and has been practically applied in many business fields [24-26]. We adopted the HSV model rather than the RGB model as the color extraction scheme because the extraction of the RGB scheme in the small pore area of the inlet filter is not ideal and the HSV color extraction scheme provides superior stability.

Future research mainly focuses on the in-depth analysis of the anomaly problem and further development of this study. Several practical limitations require consideration:

(1) Lighting stability in enclosed tunnels may affect edge detection consistency;

(2) Manual mask initialization is currently required for filter localization;

(3) Water turbulence could introduce transient noise in pore identification;

(4) With water level changes, light refraction variations may influence pore area measurements;

(5) Potential color similarity between scale deposits and pore areas may challenge extraction accuracy.

4 Conclusions

In this study, the profile mask of the inlet filter screen of the pump was selected with manual marks to extract the complete profile picture of the filter screen. Then, HSV color model was used to extract the S component which can separate the pores of the filter from non-porous areas. Finally, the pixels of the S-component map of the filter screen were classified by color Euclide distance, and the non-porous regions of the inlet filter screen were removed so that the pore pixel area was obtained by summing up the pixels representing the pores of the filter screen. By calculating the pore pixel area of the filter screen regularly and sending the result to the control room of the hydropower station, the dynamic tracking and monitoring of the blockage degree of the inlet filter screen of the pump can be realized.

Acknowledgment

This work was supported by the Open Research Fund Program of Hubei Technology Innovation Center for Smart Hydropower (1523020038).

Author contributions

Xiangqian Fu: Conceptualization, Data interpretation, Writing – original draft. Linghua Hou, Qiqi Ding and Long Xu: Supervision, Data interpretation, Writing – review & editing. Xiangqian Fu: Funding acquisition. Xiangqian Fu and Linghua Hou: Conceptualization, Supervision. All authors read and approved the final version.

Conflict of interests

All authors declare no conflict of interests.

Data availability statement

The data used to support the findings of this study are all in the manuscript.

References

[1] Esmaeilpour, M., E. Nomigolzar, M. R. F. Derakhshi, and Z. Shukur. (2011) Fault diagnostics of centrifuge pump using data analysis in spectrometric method, *Informatica*, vol. 35, pp. 259-268.

[2] Kannaujia, V., S. P. Bhore and H. S. Goyal. (2024) Failure analysis of submersible pumps-A review. In: Ghoshal, S.K., Samantaray, A.K., Bandyopadhyay, S. (eds) Recent advances in industrial machines and mechanisms. IPROMM 2022. *Lecture Notes in Mechanical Engineering*. Springer, Singapore. https://doi.org/10.1007/978-981-99-4270-1_40

[3] Affortunati, S. and B. G. Zagar. (2022) Application of impedance spectroscopy and tomography to monitor calcite deposits in drainage pipes, *Elektrotechnik & Informationstechnik*, vol. 139, pp. 524-534, 2022, https://doi.org/10.1007/s00502-022-01058-5

[4] Iqbal, U., J. Barthelemy, W. Li and P. Perez. (2021) Automating visual blockage classification of culverts with deep learning, *Applied Sciences*, vol. 11, no. 16, pp. 7561, 2021, https://doi.org/10.3390/app11167561.

[5] Singal, S. K., V. Gupta and M. Sood. (2022) A Numerical study to determine the impact of trash rack blockage on head loss, *Water and Energy International*, vol. 65, no. 7, pp. 6-11.

[6] Hribernik, A. (2020) Evaluation of clogged hydropower plant trash rack losses, *Journal of Mechanical Engineering*, vol. 66, no. 2. https://doi.org/10.5545/sv-jme.2019.6313

[7] Cui, X., T. Wen, J. Li, D. Wu, Y. Fan, Y. Jin et al. (2022) Migration and deposition characteristics of particles in sand layers with fluctuating water levels, *Water, Air, & Soil Pollution*, vol. 233, no. 4, pp. n.p. https://doi.org/10.1007/s11270-022-05603-y

[8] Tao, H., Z. Wu, Y. Zhou et al. (2023) Establishment of a dimensional analysis-based prediction model for the head loss of pre-pump micro-pressure filters for micro-irrigation, *Irrigation Science*, vol. 41, pp. 803-815. https://doi.org/10.1007/s00271-023-00879-2

[9] Liu, D. G., Y. Yang, C. J. Mao et al. (2022) A Comparative study on hydrodynamics and hydrochemistry coupled simulations of drainage pipe crystallization blockage in karst tunnels, *Journal of Earth Science*, vol. 33, pp. 1179-1189. https://doi.org/10.1007/s12583-022-1720-3

[10] Ogie, R. I., S. Dunn, T. Holderness and E. Turpin. (2017) Assessing the vulnerability of pumping stations to trash blockage in coastal mega-cities of developing nations, *Sustainable Cities and Society*, vol. 28, pp. 53-66. https://doi.org/10.1016/j.scs.2016.08.022

[11] Rana, D., S. Bichinepally, S. Kalra, A. Nandan, and N. A. Siddiqui. (2023) Comparative analysis of ultrafiltration for produced water from oil and gas industry, in *Advances in Waste Management*, N. A. Siddiqui, A. S. Baxtiyarovich, A. Nandan, and P. Mondal, Eds. Lecture Notes in Civil Engineering, vol. 301. Springer, Singapore. https://doi.org/10.1007/978-981-19-7506-6_7

[12] Fujita, C., M. S. Akhtar, R. Hidaka, et al. (2022) Mitigation of groundwater iron-induced clogging by low-cost bioadsorbent in open loop geothermal heat pump systems, *Applied Water Science*, vol. 12, p. 30. https://doi.org/10.1007/s13201-022-01574-x

[13] Nallakukkala, S. and B. Lal. (2023) Machine learning for scale deposition in oil and gas industry, in *Machine Learning and Flow Assurance in Oil and Gas Production*, B. Lal, C. B. Bavoh, and J. K. Sahith Sayani, Eds. Springer, Cham. https://doi.org/10.1007/978-3-031-24231-1_6

[14] Canny, J. (1986) A Computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, https://doi.org/10.1109/TPAMI.1986.4767851.

[15] Suneetha, A. and E. S. Reddy (2020) Robust gaussian noise detection and removal in color images using modified fuzzy set filter, *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 240-257. https://doi.org/10.1515/jisys-2019-0211

[16] Wedianto, A., H. L. Sari and H. Zuzaantri. (2016) Analisa perbandingan metode filter gaussian, mean dan median terhadap reduksi noise, *Jurnal Media Infotama*, vol. 12, no. 1. https://doi.org/10.37676/jmi.v12i1.269

[17] Chakrapani, Y. S., N. V. Rao, and M. Kamaraju. (2021) A survey of sobel edge detection vlsi architectures, *Journal of Physics: Conference Series*, vol. 1804, no. 1, p. 012151. https://doi.org/10.1088/1742-6596/1804/1/012151

[18] Rosenfeld, A. and A. Kak. (1976) *Digital Picture Processing*, p. 457. Academic Press.

[19] Kitchen, L. and A. Rosenfeld. (1982) Gray-level corner detection, *Pattern Recognition Letters*, vol. 1, no. 2, pp. 92-102. https://doi.org/10.1016/0167-8655(82)90020-4

[20] Smith, A. R. (1978) Color gamut transform pairs, *ACM SIGGRAPH Computer Graphics*, vol. 12, no. 3, pp. 12-19. https://doi.org/10.1145/800248.807361

[21] Su, C., Z. Li, Z. Wei, N. Xu, and Q. Yuan. (2023) Clarity method of low-illumination and dusty coal mine images based on improved AMEF, *Informatica*, vol. 47, pp. 51-62.

[22] Flores-Vidal, P. A., D. Gómez, J. Castro, and J. Montero. (2022) New aggregation strategies in color edge detection with hsv images, in *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, D. Ciucci et al., Eds. Communications in Computer and Information Science, vol. 1602. Springer, Cham, https://doi.org/10.1007/978-3-031-08974-9_29.

[23] Heyns, P. S., R. Deetlefs, A. J. Oberholster, T. R. Botha, P. S. Els, and D. H. Diamond. (2020) Computer vision for rail surface defect detection, *Condition Monitor*, TN.403, pp. 6-10.

[24] Shafiabadi, M., A. Kamkar-Rouhani, S. R. G. Riabi, A. R. Kahoo, and B. Tokhmechi. (2021) Erratum to: Identification of reservoir fractures on FMI image logs using canny and sobel edge detection algorithms, *Oil & Gas Science and Technology*, vol. 76, p. 33. https://doi.org/10.2516/ogst/2021023

[25] Akter, Y. A., M. A. Rahman and M. O. Rahman. (2020) Quantitative analysis of mouza map image to estimate land area using zooming and canny edge detection, *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 18, pp. 3293-3302. https://doi.org/10.12928/TELKOMNIKA.V18I6.16179

[26] Gunawan, G., H. Nuriyanto, S. Sriadhi, A. Fauzi, A. Usman, and F. Fadlina, et al. (2018) Mobile Application detection of road damage using canny algorithm, in *Proceedings of the 1st International Conference on Green and Sustainable Computing (ICoGeS)*, vol. 1019. Institute of Physics. https://doi.org/10.1088/1742-6596/1019/1/012035

Improved Attention-Enhanced Efficient Face-Transformer Model for Multimodal Elderly Emotion Recognition in Smart Homes

Peiyao Li
School of Architecture and Design, Yangtze University College of Arts and Sciences, Jingzhou, 434020, China
E-mail: peiyaolip@outlook.com

Keywords: home environment, elderly people, emotion recognition, attention, transformer

Received: April 9, 2025

Recognizing the emotions of the elderly is key to achieving personalized services in smart home environments. Traditional methods have difficulty capturing the correlation and temporal information of multimodal features. To this end, the study proposes a multimodal emotion recognition model that integrates the EfficientFace and Transformer structures to construct an improved attention mechanism. A modal interaction compensation term is introduced into the similarity calculation to improve the ability to model dynamic dependencies between modalities. Meanwhile, the dynamic importance factor is used to adaptively adjust feature weights. The model was tested on IEMOCAP and self-constructed EMED datasets. The emotion recognition precision reached up to 94.37%, the recall rate reached 93.64%, the F1 value was 94.21, and the specificity reached 94.85%. Additionally, the model achieved 96.24% classification accuracy and 94.13% emotional intensity on easily confused categories, such as "disgust" and "contempt," with a minimum detection latency of 0.55 seconds. The results show that the model exhibits excellent performance in multimodal fusion and emotion recognition for the elderly, and is suitable for the task of smart home emotion monitoring.

Povzetek: Članek uvaja izboljšan EfficientFace-Transformer z nadgrajenim mehanizmom pozornosti za multimodalno prepoznavo čustev starejših, ki dosega boljšo točnost, stabilnost in nizko zakasnitev v pametnih domovih.

1 Introduction

With the acceleration of global aging process, smart home as an emerging technology can provide convenient life services for the elderly. At the same time, the special needs of the elderly population in terms of physiology, psychology and cognition make emotion recognition (ER) an important research direction in smart home systems. Emotion is an important part of human psychological activities. Changes in the emotions of the elderly are often influenced by health conditions, environmental factors, and social interactions. Moreover, ER can provide personalized services for smart home systems to optimize the living experience of the elderly [1-3]. To solve the issues of elderly people having trouble recognizing emotions and traditional machine learning models' incapacity to adequately capture the nonlinear relationship between physiological signal data, Feng G et al. proposed a recursive mapping method of ER in conjunction with a visual transformer. According to experimental results, this method's recognition accuracy was up to 94.35%, which was higher than that of the conventional approach [4]. Using an inverted neural network as the main body and training data from audio and video modalities, Sreevidya et al. created an automated emergency room system designed especially for the elderly. Experimental results showed that the system showed a minimum relative improvement of 6.5% for happy emotions and a maximum relative improvement of 46% for sad emotions compared to the baseline model [5]. Park H et al. used MobileNet-V2 to recognize six emotions for different age groups. The results showed that the ER of teenagers was more obvious than that of older people, and the model recognition accuracy was up to 83.7% [6]. To investigate the emotional experience of the elderly when using smart terminals, Lu et al. The researchers proposed a novel accessibility ER model for the elderly after combining with the support vector machine network. The outcomes displayed that the model had the highest accuracy in detecting the negative emotions of the elderly when using the terminal. Moreover, it could help reflect the design defects of the terminal application technology [7].

Speech recognition, image processing, natural language processing, and other domains have made extensive use of the attention mechanism (AM) in recent years. Multimodal learning and time-series data analysis have seen particularly impressive outcomes [8, 9]. To investigate the hidden emotional states from human motion, Zhao et al. proposed a human multi-site ER method that combined a fuzzy algorithm, AM, and multiple inertial measurement units. The outcomes revealed that the method had the highest accuracy

of 94.02% for six common types of emergencies [10]. Saganowski developed a two-channel recognition technique in an attempt to improve the robustness of the existing ER technique by introducing the AM to optimize the attention of the traditional convolutional recognition network. Experimental results indicated that the effectiveness of this technique for human ER was stronger than before the improvement [11]. According to Zhou et al., the majority of ER models now in use overlook inter-feature interactions and fall short of capturing the crucial complimentary benefits between contextual and facial information in video clips. According to the findings, the network was quite successful in predicting the emotions associated with work-related stress based on visual emotional evidence [12]. Lin et al. came to the conclusion that emotional shifts brought on by outside factors were accompanied by measurable and identifiable alterations in physiological signals. For this reason, this study proposed a dynamic ER model by combining EEG signals, ECG signals, EMG signals, and AM. The experimental results indicated that the model was robust to ER in application scenarios with different physiological signals and was suitable for different occasions of ER work [13]. The review of the above literature is summarized in Table 1.

Table 1: Literature review summary comparison table

Authors	Method/Model	Advantages (Metrics)	Limitations
Feng G et al. [3]	Visual Transformer+recurrent mapping	Accuracy up to 94.35%	Dependent on physiological signals, weak generalization
Sreevidya P et al. [4]	Audio-visual Fusion+recurrent neural network	46% improvement for sadness; 6.5% for happiness	No refined cross-modal interaction mechanism
Park H et al. [5]	MobileNet-V2	Accuracy up to 83.7%	Limited performance on elderly users
Lu J et al. [6]	SVM-based emotion recognition model	High accuracy in detecting negative emotions	Limited to mobile terminal usage scenarios
Zhao Y et al. [8]	IMU+Attention+Fuzzy Algorithm	Accuracy up to 94.02% for six emotions	Single modality based on IMU only
Saganowski S [9]	Dual-channel CNN with Attention	Improved performance over standard CNN	Insufficient modeling of inter-modal interaction
Zhou S et al. [10]	Cross-attention+hybrid feature weighting	Effective fusion of facial and contextual cues	Complex modeling, lacks real-time performance
Lin W et al. [11]	EEG/ECG/EMG+dynamic attention	High robustness across physiological modalities	Heavy reliance on sensors, low adaptability

In summary, although existing research has produced some results in the field of ER, significant deficiencies remain in addressing the complexity of emotional expression and the unique multimodal feature interactions of the elderly population. These deficiencies are evident in the rough modal information fusion strategy and limited cross-modal dynamic modeling capability. This makes it difficult to meet the demand for high-precision, low-latency ER in smart homes. To this end, the study focuses on the following core research questions: (1) Can the introduction of dynamic importance factors significantly improve the accuracy of modal feature weight adjustment? (2) Can the modal interaction compensation term enhance the co-modeling capability between audio and video? (3) Can the combination of the above mechanisms improve the F1 value by at least three percentage points over the baseline model? To validate the above questions, the study focuses on two key challenges of multimodal ER in older adults. First, it proposes an end-to-end recognition model that incorporates the EfficientFace-Transformer structure with an improved AM. The model introduces a dynamic importance factor and a modal interaction compensation term. These terms strengthen the deep interaction relationship between the audio and video modalities, as well as the ability to regulate weights. This allows for more efficient information fusion and robust modeling while maintaining the integrity of the modal representation. The core innovation of this study is the application of an optimized AM to ER scenarios for the elderly for the first time. A generalizable, multimodal emotion fusion framework is systematically constructed and verified for actual smart home requirements in terms of performance and practicability. This provides a feasible path and theoretical support for constructing a personalized emotion service system.

2. Methods and materials

2.1 Multimodal emotion recognition based on EfficientFace-Transformer

Due to the special physiological, psychological, and cognitive needs of the elderly, their emotional expressions are often different from those of young people and are affected by a variety of factors, such as health status, living environment, and social interactions [14-15]. Therefore, a single ER method may not be able to comprehensively and accurately reflect the emotional state of older adults. The multimodal fusion approach is a useful technique for enhancing ER's accuracy and robustness. By integrating data from multiple modalities, such as facial expression, speech, movement, heart rate, etc., it can make up for the possible limitations of a single modality and provide a more comprehensive and accurate emotion analysis [16-17]. Whereas the effectiveness of the multimodal fusion method mainly relies on the close collaboration of data layer (DL), feature layer (FL), and decision layer (DeL). The schematic diagram of DL, FL, and DeL is shown in Figure 1 [18].

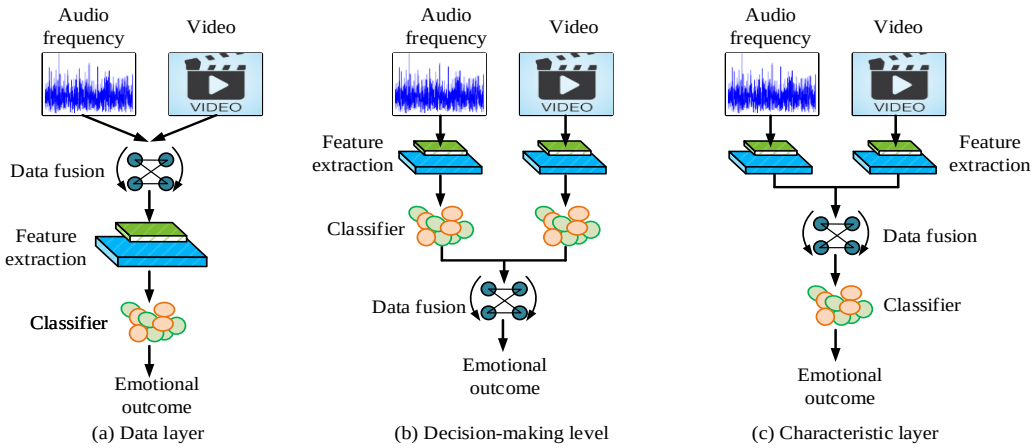


Figure 1: Diagram of data layer, decision layer, and feature layer of multimodal fusion

Figure 1(a), 1(b), and 1(c) shows the schematic diagram of DL, DeL, and FL of multimodal fusion. Data from different modalities such as audio and video in the DL are directly fed into the system and fused to form a joint multimodal data representation. Subsequently, these data are passed through a unified feature extraction module to extract key features and a classifier to classify the emotions. The DeL and FL processes are roughly similar, differing only in the order of feature classification and data fusion. It can be concluded that the multimodal fusion method can capture the emotion information more comprehensively and improve the robustness and accuracy of ER compared to the single modality recognition method. However, traditional multimodal fusion recognition methods often use separate feature extraction and simple fusion strategies when dealing with data from different modalities, which makes it difficult to fully explore the inter-modal correlations and temporal information [19]. For this reason, the study relies on the complementary information between different modalities and makes full use of this feature to propose an end-to-end ER model for the elderly. Figure 2 displays the model's structure.

In Figure 2, in the audio modality, the features of the audio signal are first extracted by Mel-frequency cepstral coefficients (MFCC) feature extraction algorithm, which generates audio features reflecting the rhythm, pitch, and emotion intensity of speech. In video modality, features of facial expressions in video frames are extracted using EfficientFace network structure. Next, one-dimensional convolutional coding is applied to the audio and video features to improve their temporal modeling capabilities. Then, they are fed into the Transformer module, which uses its self-AM to capture intra- and inter-modal feature relationships and generate fused multimodal feature representations. The fused features are further dimensionality reduction processed through the fully connected layer to extract the key emotion information features. Finally, the Softmax classifier is used to complete the emotion classification task, outputting the recognition outcomes of the emotion categories including pleasure, anger, sadness, surprise, and so on. To ensure reproducibility and stable training of the model, the study implements the following configuration: The Transformer module is set as a stacked structure of three layers, each of which contains eight heads of multi-head attention. The embedding dimension is 256. The width of the feedforward layer is 512. Moreover, the activation function is GELU. A dropout operation with a rate of 0.1 is added in the fusion stage to prevent overfitting. The model is trained using the Adam optimizer with an initial learning rate of 1×10^{-4}, a batch size of 32, and 120 training rounds. The Early Stopping strategy is introduced to stabilize convergence performance. Figure 3 displays the schematic diagram of the EfficientFace-Transformer, which serves as the backbone network of the entire recognition model.

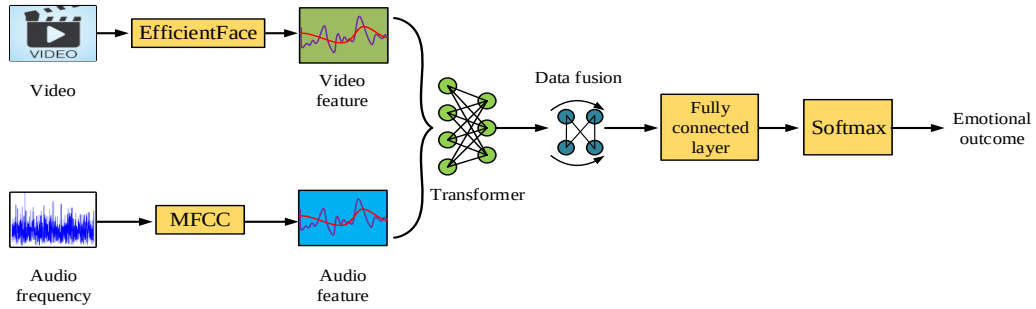


Figure 2: Multimodal end-to-end emotion recognition model for the elderly

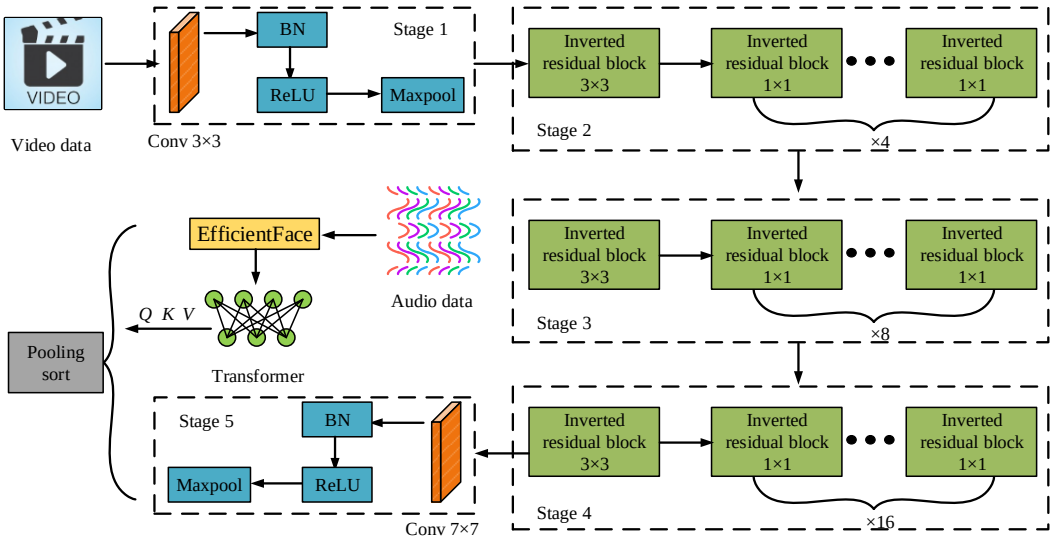


Figure 3: EfficientFace-Transformer diagram

In Figure 3, first, the data of video modality is fed into the EfficientFace network for multi-stage feature extraction. Among them, Stage 1 completes the generation of preliminary feature representations through convolutional and maximum pooling layers. Stages 2 to 4 use stacked inverted residual blocks to extract high-level semantic features of the video modality. In terms of processing the temporal dimension of video frames, the study employs a strategy combining frame-level feature splicing and average pooling. First, feature vectors are extracted for each frame independently. Next, frame sequence feature representations are obtained by splicing the frames one by one in the temporal dimension. Finally, frame sequence features are integrated via time-averaged pooling to generate compact feature vectors representing the entire video clip. Stage 5 generates compact feature vectors by global average pooling dimensionality reduction. Meanwhile, the audio modality is processed by a 1D convolutional layer on the input MFCC features to extract the key features of the audio modality. Next, the features of the video modality and the features of the audio modality are used as inputs to the Transformer module, respectively. The Transformer computes the relationship between Q, K, and V through its self-AM to dynamically capture the intra- and inter-modal interaction features. Subsequently, after pooling operation, the fused features of audio and video modalities are further compressed and integrated, and finally the task of classifying emotion categories is accomplished through the fully connected layer. Since the audio is a one-dimensional temporal data, it needs to be first framed and windowed to maintain its temporal structure, and the computation process is shown in Equation (1)[20].

$$x_n(m) = x(a + nR) \cdot \omega(m) \tag{1}$$

In Equation (1), $x_n(m)$ denotes the signal of n frame. m denotes the time index within the frame. R denotes the interval between the upper and lower frames. $\omega(m)$ denotes the window function. The video data consists of a series of consecutive frames, and the frames need to be sampled, grayscaled and normalized in preprocessing to extract the key expression information related to emotions. The computational formula is shown in Equation (2).

$$I'(x,y) = \frac{I(x,y) - \mu}{\sigma} \tag{2}$$

In Equation (2), $I(x,y)$ and $I'(x,y)$ are the pixel values before and after normalization, respectively. $I(x,y)$ is the mean of the pixel values. $I(x,y)$ is the standard deviation of the pixel values. After preprocessing, the formula for calculating the filtered energy on the Mel scale is shown in Equation (3).

$$E_i = \sum_{k=f_{i-1}}^{f_i} |X(k)|^2 H_i(k) \tag{3}$$

In Equation (3), E_i is the energy of the i th Mel filter.

$H_i(k)$ is the transfer function of the i th filter. f_i and f_{i-1} are the upper and lower limits of the Mel filter frequency, respectively. $|X(k)|^2$ denotes the frequency domain power spectrum. Equation (4) illustrates how the Transformer module determines the attention weights.

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

In Equation (4), Q, K, and V distributions denote the query, key, and value matrices. d_k denotes the dimension of the key vector for mapping the fused features to the sentiment category probability distribution. Equation (5) displays the computational formula.

$$P(y=i|z) = \frac{e^{w_i^T z+b_i}}{\sum_{j=1}^{C} e^{w_j^T z+b_j}} \quad (5)$$

In Equation (5), $P(y=i|z)$ denotes the probability that the sample belongs to emotion category i. z denotes the fusion feature. w_i and b_i denote the weight and bias of the i th emotion category, respectively. C denotes the total number of emotion categories.

2.2 Optimization of a multimodal emotion fusion recognition model for the elderly incorporating the attention mechanism

After constructing the completed EfficientFace-Transformer-based multimodal emotion fusion recognition model, it is found that the EfficientFace-Transformer is a multimodal information fusion of audio and video at a later stage. This, while better able to preserve the independent feature representation of each modality, still has some limitations in capturing the deep inter-modal interactions and joint dynamic properties. Specifically, the late fusion strategy mainly relies on a single fusion layer before classification for the integration of modal features, which may lead to insufficient inter-modal information interactions, thus affecting the robustness and accuracy of ER [21]. To improve the ability to model multimodal feature interactions, the study optimized the self-AM of the standard Transformer module. The Transformer still serves as the modal fusion backbone network in the overall design, but the part that computes internal attention is replaced with an improved AM. This mechanism introduces three terms to the standard dot product attention: a modal interaction compensation term, a dynamic importance factor, and a cross-modal enhancement term. These terms enhance the inter-modal collaborative modeling capability. Figure 4 shows the structure of the traditional dot product AM for comparison purposes. Figure 5 shows the structure of the improved attention module embedded in the Transformer of the final model, as proposed in the study. Figure 4 shows the structure of the standard dot product AM for comparison with the improved mechanism. The improved mechanism has not been directly applied to the final model [22].

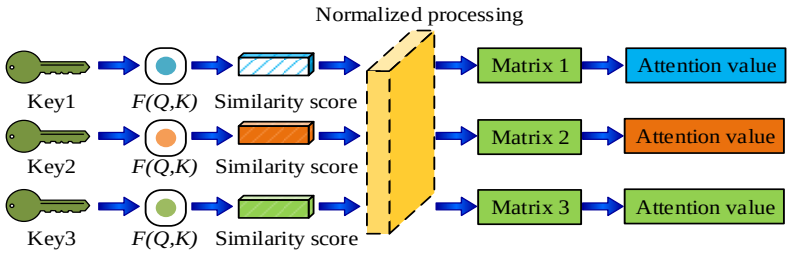


Figure 4: Attention mechanism diagram

Figure 4 shows that the AM is divided into three stages. These stages gradually complete feature weighting, strengthen the interaction between audio and video, and highlight key emotional information. The initial step involves mapping the input characteristics into query, key, and value vectors. Then, using the dot product, the similarity score between each query and all keys is calculated. The weight matrix is obtained in the second stage by normalizing the similarity scores produced in the first stage using the Softmax function. Equation (6) displays the calculating formula.

$$S1_{ij} = soft\max\left(\frac{\exp(F(Q_i,K_j)+W_m)}{\sum_j \exp(F(Q_i,K_j)+W_m)}\right) \quad (6)$$

In Equation (6), $S1_{ij}$ denotes the i th row and j th column element of the normalized weight matrix. $F(Q_i,K_j)$ denotes the similarity score matrix obtained from the first stage by Equation (4). W_m denotes the modal association weight vector. In the third stage, the attention value is also combined with the residual connections between modalities to preserve the original features of the modalities and enhance the training stability of the model.

Equation (7) displays the computational formula.

$$AttentionValue = S1 \cdot V + \lambda U \tag{7}$$

In Equation (7), λ denotes the residual scale factor. U denotes the initial input features. V denotes the initial eigenvalue matrix. In order to capture the deep interactions between audio and video more effectively, the AM is improved in this study. The improved AM is shown in Fig. 5. Its structure is used to replace the standard self-attention module in Transformer and embedded into the final model.

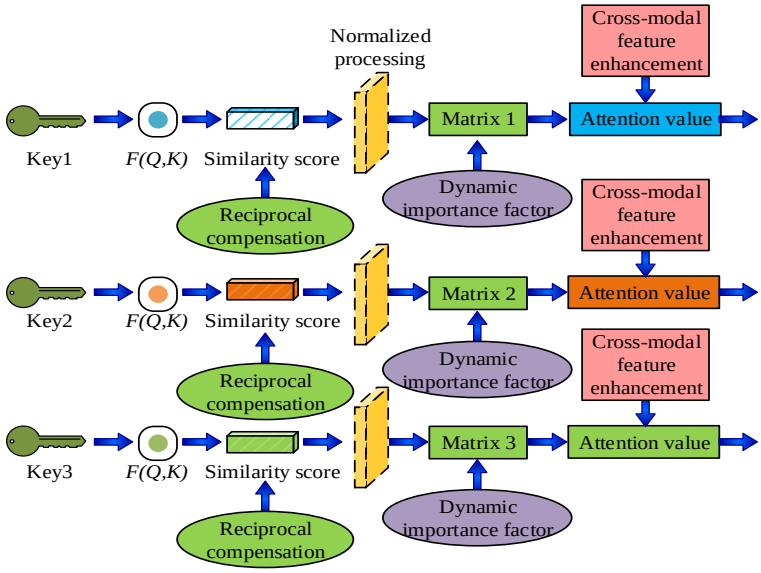


Figure 5: Improved attention mechanism diagram

In Figure 5, the improved AM is mainly generated by the similarity computation improvement, the weight normalization dynamic adjustment, and the enhancement of the final attention value. In the similarity calculation stage, the modal interaction compensation term is introduced. By combining the dot product similarity with the modal interaction compensation term, the dynamic dependency modeling between modes is realized. In the weight normalization stage, a dynamic importance factor is added. The generated dynamic importance factor is able to adaptively adjust the weights of modal features by computing the differences between the global features of audio and video modalities. In the attention value generation stage, a cross-modal feature (CMF) enhancement term is added. The fusion effect of audio and video modalities is further enhanced by feature splicing and weight adjustment. In this process, the similarity score is calculated as shown in Equation (8).

$$\begin{cases} F'(Q,K) = \dfrac{QK^T}{\sqrt{d_k}} + B_m \\ B_m = W_b \cdot (ReLU(Q \cdot K^T) - \gamma), \end{cases} \tag{8}$$

In Equation (8), B_m denotes the modal interaction compensation term. W_b denotes the learnable modal compensation weight matrix. γ denotes the modal balance bias. d_k denotes the dimension of the key vector. $F'(Q,K)$ denotes the similarity score after considering the modal compensation term. Then, the dynamic importance factor is added in the weight normalization stage. The D_m calculation formula is shown in Equation (9).

$$D_m = \eta(W_d \cdot (G_a - G_v)) \tag{9}$$

In Equation (9), D_m denotes the dynamic moderator based on modal significance. G_a and G_v denote the global features of audio and video modalities, respectively. W_d denotes the importance adjustment weights. η denotes the Sigmoid function. Finally, in the attention value generation stage, the feature representation is further optimized by introducing CMF enhancement terms in combination with inter-modal contextual information. The attentions value calculation at this time is shown in Equation (10).

$$AttentionValue' = S1 \cdot V + \lambda U + C_m \tag{10}$$

In Equation (10), $AttentionValue'$ denotes the final stage attentional output result. $S1 \cdot V$ denotes the first stage attentional output. C_m denotes the CMF enhancement term, which is calculated as shown in Equation (11).

$$C_m = W_c \cdot (Concat(G_a, G_v)) \tag{11}$$

In Equation (11), W_c denotes the cross-modal enhancement weight matrix. $Concat(_)$ denotes the feature splicing operation. In summary, the study finally proposes a novel multimodal emotion fusion recognition model for the elderly that combines EfficientFace-Transformer and improved attentions. Figure 6 displays the model's structure.

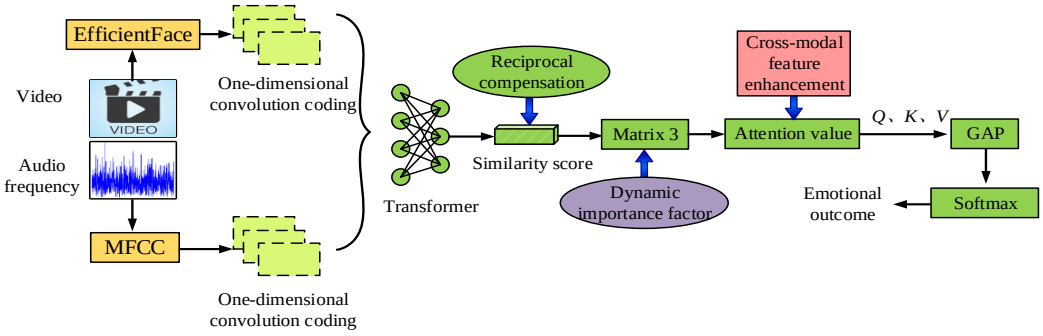


Figure 6: New multimodal emotion fusion recognition model for the elderly

Figure 6 shows that the video modalities' features are first extracted through the EfficientFace network for multilayer representation. Meanwhile, the audio modalities' features are encoded through a one-dimensional convolutional layer to preserve their time-series information. During the intermediate fusion phase, the Transformer module uses an improved AM, as shown in Fig. 5. This mechanism replaces the standard self-attention operation to improve the model's ability to represent global relationships between modalities. The modal feature weights are adaptively adjusted through an improved AM combined with a dynamic importance factor, which significantly enhances the deep inter-modal interaction capability. Specifically, the AM dynamically introduces a modal compensation term and a global feature enhancement term on top of Q, K, and V. The model is designed to capture emotionally relevant CMFs more accurately. This enables the model to capture emotionally relevant CMFs more accurately. The fused features are further processed through a 1D convolutional layer to enhance feature compactness. It is also combined with a global pooling layer to generate the final emotion feature representation, and finally the recognition of emotion categories is completed by a classification layer. Finally, the identification of emotion categories is accomplished through the classification layer. This is the model architecture that represents the final model proposed in the results and conclusions section. The algorithmic pseudo-code of the final model is shown in Figure 7.

```
Algorithm: Multimodal Emotion Recognition with Improved Attention Mechanism

Inputs:
   - V: video frame sequence (T × H × W)
   - A: audio MFCC sequence (T × F)
   - EfficientFace backbone (pretrained)
   - Transformer encoder parameters θ

Output:
   - Predicted emotion label y_pred

# === Feature Extraction ===
For each frame v in V:
    f_v ← EfficientFace(v)
f_a ← Conv1D(MFCC(A))

# === Improved Attention Mechanism ===
Compute Q, K, V from [f_v, f_a]
Att_base ← softmax((Q × K^T) / √d_k + W_m)        # modal relevance
Att_res ← Att_base × V + λ × f_v                  # residual connection
B_m ← W_b × ReLU(Q × K^T − γ)                     # modality compensation
D_m ← σ(W_d × (G_a − G_v))                        # dynamic importance factor
C_m ← W_c × Concat(G_a, G_v)                      # cross-modal enhancement
Att_final ← Att_res + D_m + C_m                   # final attention output

# === Classification ===
Fused_feature ← GlobalAvgPooling(Att_final)
y_pred ← Softmax(Linear(Fused_feature))

# === Training Procedure ===
Loss ← CrossEntropy(y_pred, y_true)
Optimizer: Adam (lr = 1e-4, batch_size = 32)
Train for 120 epochs with early stopping
```

Figure 7: Algorithmic pseudo-code for the final model

3. Results

3.1 Performance test of multimodal emotion fusion recognition model for the elderly

The study is set up with AMD Ryzen 9 5950X for CPU, NVIDIA GeForce RTX 3090 for GPU, 64GB of RAM, and Ubuntu 20.04 for the operating system. The PyTorch framework is also used for model implementation. Two multimodal ER datasets are used for model validation: IEMOCAP and the self-constructed EMED dataset. IEMOCAP (Interactive Emotional Dyadic Motion Capture Database) is a publicly available multimodal emotion dataset containing audio and video data of 10 actors expressing multiple emotion categories such as anger, happiness, sadness, surprise, etc. The EMED (Elderly Multimodal Emotion Dataset) is a dataset designed for this study. It is intended for use with an elderly population and is designed to elicit real emotional responses through natural conversations, image stimulation, and situational recollection. It also synchronizes facial video and voice signals. Facial videos and voice recordings are collected simultaneously. The dataset includes the five most common emotional states of the elderly: pleasure, sadness, anger, anxiety, and neutrality. It covers multiple home environments and interaction scenarios to simulate the actual perceptual needs of smart home applications. The study first tests the selected values of two types of hyperparameters that have a large impact on the pre-signal feature extraction and mid-signal feature fusion. The test results are shown in Figure 8.

The test results for the emotion weights are displayed in Figure 8(a). The test results of calculating the importance regulatory weights are displayed in Figure 8(b). Figure 8 shows the average of five independent training runs performed with the corresponding weight settings, represented by each loss curve. While the standard deviation is not shown directly in the figure, the average trend of all configurations remains consistent across multiple runs, and the convergence interval fluctuates minimally, with a maximum standard deviation of less than ± 0.025. This verifies the reproducibility and controllability of the hyperparameters on the model's convergence. In Figure 8(a), the loss function of the model decreases the fastest when the emotion weight is 0.6, and the loss value after final convergence is the lowest. It indicates that the model can more effectively balance the significance of features from various emotion categories with this weight value, increasing feature extraction accuracy and efficiency. When the weights take the value of 0.2 or 0.8, the convergence speed of the model becomes slower and the final loss value is higher. It indicates that too low or too high emotion weights can lead to bias in the model's capture of emotion information. In Figure 8(b), the model has the least fluctuation at the beginning of the iteration when the weights take the value of 0.75. The downward trend of the loss function is the most stable, and the final loss value is also optimal. This suggests that the model may distribute the weight percentage of various modal features more efficiently at this weight value. Thus, it can capture important emotional information more precisely during the feature fusion process. However, the model's iterative process exhibits greater volatility and worse convergence when the weights are set to 0.25 or 1.00. It implies that incorrect weighting could result in a drop in feature fusion effectiveness. The study tested the model for ablation and the results are shown in Figure 9. Among them, attention denotes the use of the standard dot product AM (as shown in Figure 4) and improved attention denotes the introduction of the optimization structure of modal interaction compensation with importance factors (as shown in Figure 5).

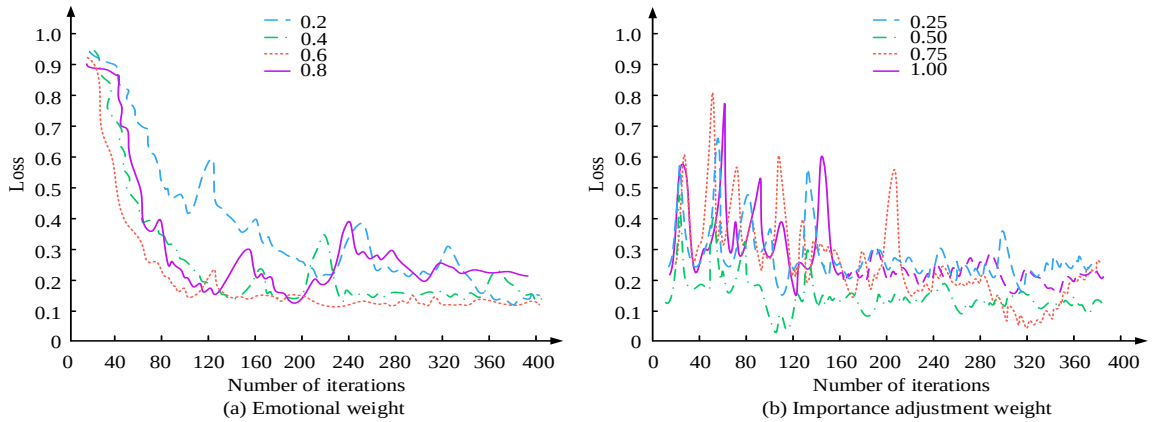


Figure 8: Hyperparameter value test

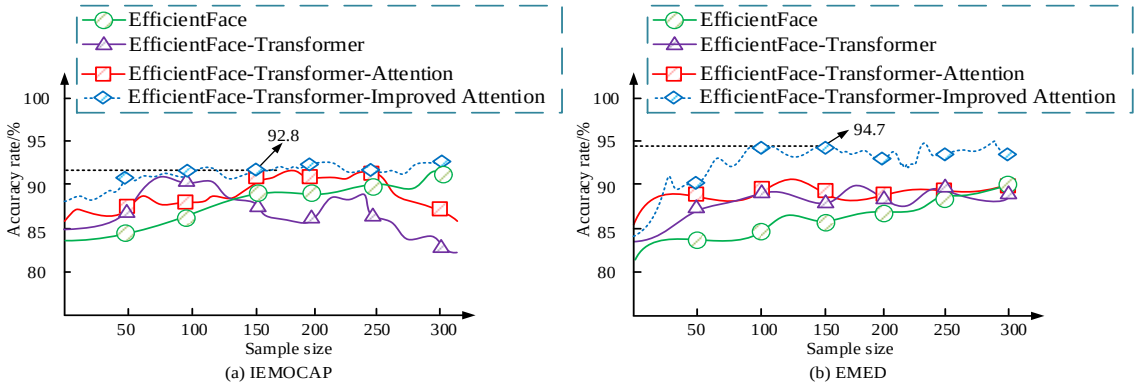


Figure 9: Ablation test results

Figure 9(a) displays the outcomes of the model ablation test in the IEMOCAP and EMED dataset. In Figure 9(a), the EfficientFace-Transformer-Improved attention model has the highest accuracy of 92.8% for a sample size of 300, which is significantly better than the other models. In contrast, the EfficientFace model has the lowest accuracy of about 86.4%. The introduction of the Transformer and improved AM significantly improves the modeling ability and recognition performance of intermodal interactions. Figure 9(b) shows that the Improved attention model also performs best on the EMED dataset, with an accuracy of 94.7% when the sample size reaches 300. In contrast, the EfficientFace model performs the worst, indicating the limited effectiveness of unimodality in the recognition of emotional features in the elderly. Improved AM and multimodal fusion strategy are more advantageous. Additionally, the accuracy curves in Fig. 8 are plotted based on the mean results of each group of model configurations that are trained independently five times with the same sample size. In the experiments, the standard deviation of the accuracy of each model under different sample sizes is calculated. The standard deviation of EfficientFace-Transformer-Improved Attention is controlled to be within $\pm 1.2\%$ when the number of samples is greater than 150. This significantly outperforms the other compared models. This result indicates that the model not only has high accuracy, but also performs stably with good statistical consistency under different training subsets. The study continues to test the effects of different modular incremental optimizations on the final model's performance. This includes the effects of independently introducing modal interaction compensation, a dynamic importance factor, and CMF enhancement terms. The results are shown in Table 2.

Table 2: Modal interaction compensation and dynamic significant factor ablation test results

Model Variant	Emotional intensity/%	mAP/%	Detection delay/s
Baseline (EfficientFace-Transformer)	89.14	91.78	0.62
+ Modality Interaction Compensation	91.03	93.4	0.61
+ Dynamic Importance Factor	90.84	93.18	0.61
+ CMF Enhancement (Final)	93.67	95.24	0.55

In Table 2, after adding only the modal interaction compensation term, the affective strength is increased from 89.14% to 91.03%, and the mean average precision (mAP) from 91.78% to 93.40%. Meanwhile, the detection latency is slightly decreased to 0.61 s. The addition of the dynamic importance factor alone improves the affective strength to 90.84%, the mAP to 93.18%, and the latency remains the same. Further incorporating CMF enhancement optimizes the model's performance, increasing the sentiment intensity and mAP to 93.67% and 95.24%, respectively, while reducing the detection latency to 0.55 seconds. This demonstrates significant improvements in accuracy and real-time performance through the synergistic optimization of multiple mechanisms. This ablation validation clearly shows that each component contributes to performance gains in different ways. The final improved model achieves systematic improvements in fusion performance while maintaining recognition efficiency. Visual geometry group face (VGGFace), face embedding network (FaceNet), and efficient neural network (EfficientNet) are introduced in the study. The precision, recall, F1 value, and specificity are used as metrics for comparison testing. Table 3 displays the findings.

In Table 3, in the IEMOCAP dataset, the F1 value of the proposed model is studied to reach 92.44%, which is about 3.29% higher than that of EfficientNet, and the specificity reaches 93.12%, reflecting its significant advantage in reducing misidentification. The suggested model's performance is considerably better in the EMED dataset. Compared to the other models under comparison, the F1 value, precision, and recall are all substantially higher at 94.21%, 94.37%, and 93.64%, respectively. This indicates

that the proposed model under study has stronger generalization ability and robustness in specific scenarios targeting elderly ER. In contrast, VGGFace has relatively insufficient feature extraction capability due to its shallow network structure, which leads to its performance being lower than FaceNet and EfficientNet on both types of datasets. FaceNet and EfficientNet show significant improvement in ER performance through deeper network structure and optimization strategies. However, it still performs less well than the the proposed model in the complex multimodal emotion fusion task.

3.2 Simulation test of multimodal emotion fusion recognition model for the elderly

The study extracts seven types of emotional states (anger, disgust, fear, happiness, surprise, sadness, and contempt) from the IEMOCAP dataset. These states are used to test the fine-grained classification performance of multimodal ER. The study excludes the interference of other modal information, such as audio. It also compares the differences in emoji emotion classification performance between models in a unimodal manner. The study plotted the confusion matrix results are shown in Figure 10.

Figure 10(a), 10(b), 10(c), and 10(d) shows the expression recognition results of VGGFace model, FaceNet model, EfficientNet model, and proposed model. The recognition accuracy of the research proposed model for the seven types of expressions, namely, angers, detest, fear, happy, amazed, sad, and comtempt, are 99%, 95%, 93%, 92%, 95%, 97%, and 96%, respectively, which surpass those of the other models by a considerable margin. Especially, it shows higher accuracy on the difficult-to-distinguish emoji categories such as detest and comtempt. In contrast, the VGGFace model, while better at recognizing ananger and comtempt, performs the worst with a recognition accuracy of only 69% on the fear category. In addition, the FaceNet model outperforms VGGFace overall, but has 76% and 72% on the amazed and fear categories, respectively, indicating its limited ability to distinguish complex expression features. The EfficientNet model recognizes the anger and sad categories better, but the accuracy on the amazed category is only 65%, which is also deficient. Receiver Operating characteristic curve (ROC) is used as an indicator to validate the most prominent expression emotions of anger and sadness, and the results are shown in Figure 11.

Figure 11(a) shows the ROC curves of different models for the detection of anger emotion, and Figure 11(a) shows the ROC curves of different models for the detection of sadness emotion. Among them, the horizontal axis represents the false-positive rate, and the vertical axis represents the true-positive rate. The larger the area under the curve (AUC) value surrounded by the ROC curve and the horizontal and vertical coordinates, the better the model's performance. In Fig. 11(a), In the anger ER task, the ROC curve of the model proposed in this study is significantly higher than those of the other three comparative methods. It has a larger overall coverage area, and its final AUC value is 0.91. This is better than the values of EfficientNet (0.84), DenseNet (0.85), and Xception-DeepLab (0.82). As shown in Fig. 11(b), the model presented in this study still performs best for sadness detection, with an AUC value of 0.87. This demonstrates its ability to accurately perceive and model emotional changes. Notably, the ROC curves of other methods in both emotion categories show different degrees of jitter or a tendency to be close to the diagonal. In contrast, the proposed model can obtain a high true positive rate when the false positive rate is low. This indicates that the model possesses strong early differentiation ability while ensuring accuracy. The models are evaluated based on their performance in classification consistency for multimodal ER, as shown in Figure 12. This metric measures the consistency of a model's predicted results for each emotion type across multiple independent runs on the same sample set. Its computational formula is shown in Equation (12).

$$ECC = \frac{1}{C}\sum_{i=1}^{C}\frac{n_i^{consistent}}{n_i} \tag{12}$$

In Equation (12), C represents the total number of emotion categories. n_i represents the total number of samples in the test set for category i emotions. $n_i^{consistent}$ represents the number of samples with consistent prediction results over multiple times in this category. This indicator takes the value range of [0,1], which can be converted to percentage expression (unit: %). The higher value represents the more stable classification results of the model on the category of emotions. The study continues with emotion classification consistency as a metric. The results are shown in Figure 12.

Figure 12(a) shows the different model emotion classification consistency test results under IEMOCAP dataset. Figure 12(b) shows the results of emotion classification consistency test for different models under EMED dataset. In Figure 12(a), the emotion classification consistency of the proposed model is always maintained at a high level in both types of data. It also rapidly approaches the ideal variation curve with the increase of the number of samples, and its emotion classification consistency reaches up to 96.24%. In contrast, the VGGFace model has the lowest consistency of 85.74%. When the number of samples is large, its classification stability significantly decreases. The FaceNet model performs more consistently in the medium sample size range, but its agreement rate is only about 90.23% at high sample sizes, failing to improve further. The EfficientNet model has a slightly higher agreement rate than FaceNet, but when dealing with complex emotion categories, its agreement rate increases significantly less than that of the proposed model. In summary, studying the proposed model shows better recognition accuracy and stability in a variety of ER tests and is suitable for complex ER tasks. To further evaluate the adaptability of the proposed models under different data distribution and acquisition scenarios, this study conducts cross-dataset validation experiments. Specifically, the models undergo cross-training on the IEMOCAP dataset and cross-testing on

the EMED dataset. This examination assesses the models' ability to recognize emotions in unseen samples from the target domain. Table 4 summarizes the main evaluation metrics of each model in this generalization test, including emotion intensity, detection latency, and mean accuracy of recognition (mAP). The goal is to validate the robustness and stability of the models in a cross-domain migration environment. Among them, the "detection latency" reported in Table 4 refers to the actual wall-clock time, in seconds, that the model experiences during the inference phase, from when the input modal features are loaded to when the emotion classification results are output. The measurement process includes the complete inference process of feature preprocessing (e.g., MFCC extraction and image normalization), multimodal coding, Transformer interaction modeling, and final classification prediction.

Table 3: Multiple indicator test results

Data set	Model	P/%	R/%	F1/%	Specificity/%
IEMOCAP	VGGFace	84.32	83.47	83.89	85.14
	FaceNet	86.45	85.62	86.03	87.28
	EfficientNet	89.58	88.72	89.15	90.34
	Our model	92.84	91.96	92.44	93.12
EMED	VGGFace	85.21	84.36	84.78	86.19
	FaceNet	87.48	86.53	86.99	88.42
	EfficientNet	90.15	89.42	89.78	91.03
	Our model	94.37	93.64	94.21	94.85

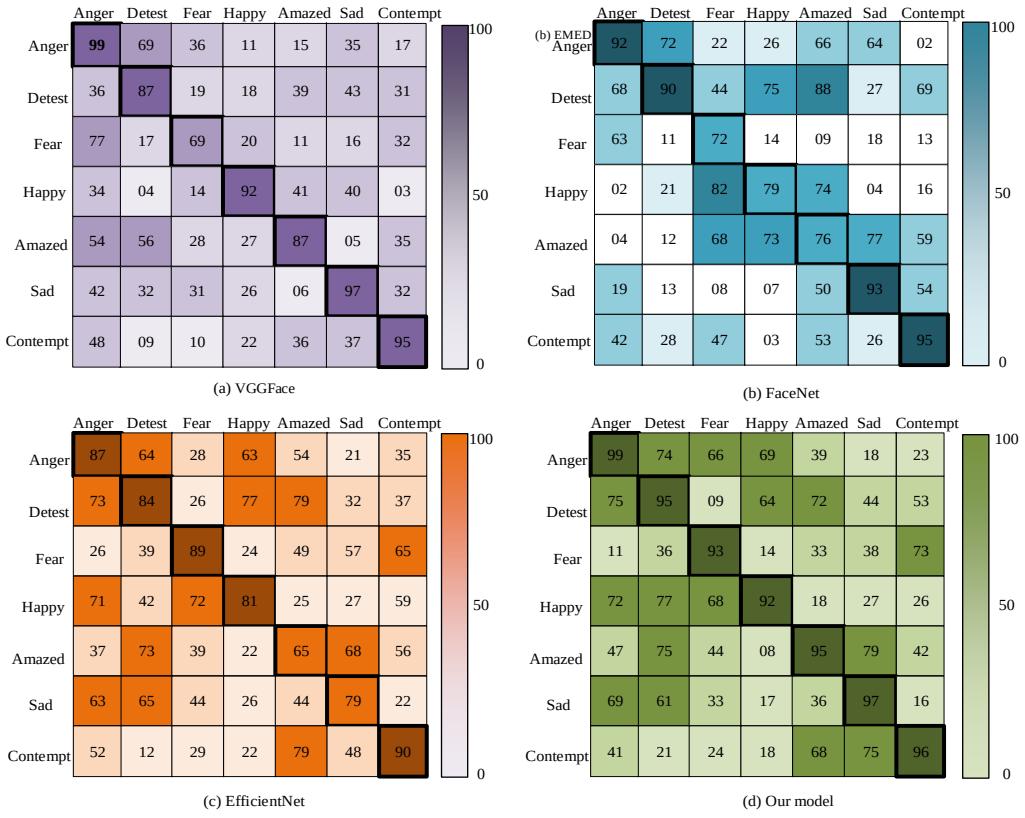


Figure 10: Confusion matrix results for seven types of facial emotion recognition based on video modality in IEMOCAP dataset by different models

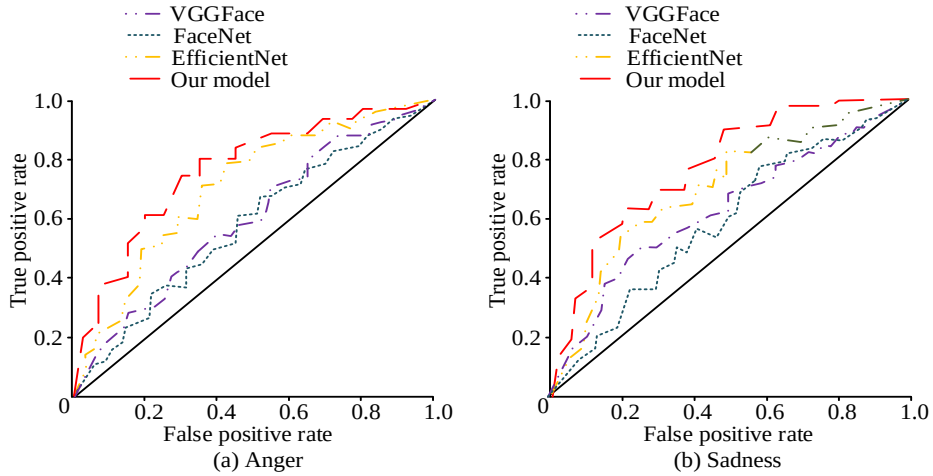


Figure 11: Statistical results of AUC indicators

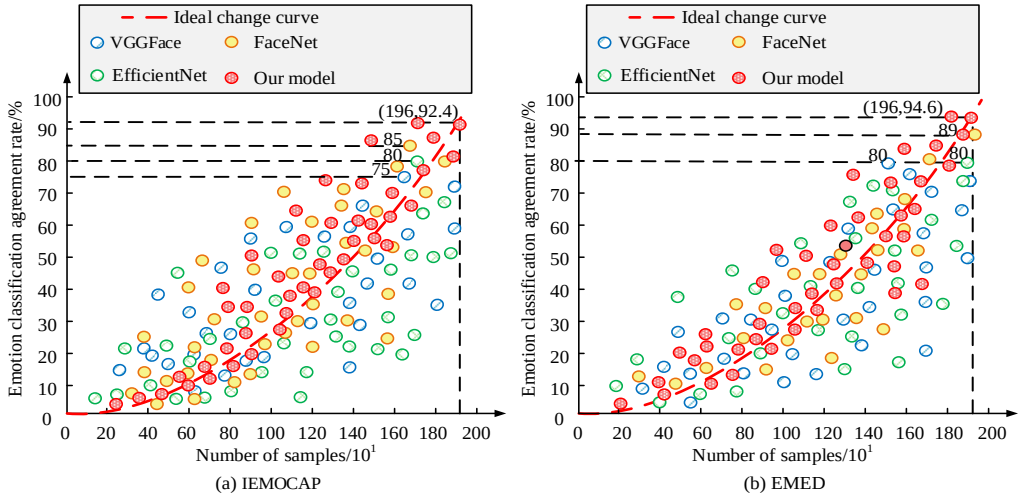


Figure 12: Results of emotion classification consistency test for different models

Table 4: Test results of emotion recognition index

Data set	Model	Emotional intensity/%	Detection delay/s	mAP/%	Inference cost/MB
IEMOCAP	VGGFace	82.45	0.72	85.32	73
	FaceNet	85.62	0.68	88.45	88
	EfficientNet	89.14	0.62	91.78	91
	Our model	93.67	0.55	95.24	94
EMED	VGGFace	80.38	0.75	84.19	73
	FaceNet	84.21	0.69	87.56	88
	EfficientNet	88.92	0.63	91.03	91
	Our model	94.13	0.56	96.12	94

In Table 4, on the IEMOCAP dataset, the study's proposed model performs optimally in terms of sentiment strength, detection latency, and mAP. Its computational inference cost is 94 MB per sample, which is an acceptable overhead for guaranteed accuracy. In terms of detection latency, the model outperforms VGGFace's 0.72 seconds and EfficientNet's 0.62 seconds with a minimum response time of 0.55 s. This reflects a clear real-time advantage. For the mAP, the model reaches 95.24%, which is an improvement of 6.79% and 3.46% compared to FaceNet and EfficientNet, respectively. In the EMED dataset, the model achieves a sentiment strength of 94.13% and an mAP of 96.12%. The detection latency is 0.56 seconds, maintaining the model's leading position and demonstrating its ability to adapt to varying inference loads. Overall, the proposed model improves multimodal fusion accuracy by introducing the dynamic AM while considering inference efficiency and computational cost. This demonstrates good practicality and balance. On the other hand, VGGFace has the weakest overall performance due to its ineffective modeling of

complex features between modalities.

4 Discussion

Aiming at the problems of rough modal fusion and insufficient dynamic dependency modeling in multimodal ER for the elderly, the study constructed a fusion model combining the EfficientFace-Transformer structure with an improved AM.

The results of the ablation experiments demonstrated that improved ER performance stemed not only from the efficient feature extraction backbone network but also from the dynamic importance factor and the modal interaction compensation term introduced in the AM layer. The former could dynamically adjust the importance of different modal features, significantly enhancing the model's ability to adapt to heterogeneous cross-modal information. The latter mitigated the problem of modal misalignment by introducing a compensation mapping that made intermodal interaction more adequate. In comparison experiments in which each module was introduced one by one, recognition accuracy (mAP) improved from 91.78% to 95.24%. This verified the improved strategy's contribution to the performance of the multimodal fusion layer.

In terms of computational resources, although the proposed model added an attention enhancement layer, the overall model size was only 94.00 MB and the inference latency was controlled within 0.55 seconds. The model considered both accuracy and inference efficiency. It was suitable for smart home edge device environments where both response time and resource overhead were limited. Meanwhile, the fusion mechanism proposed by the study has better interpretability than the direct splicing structure of the original EfficientFace and Transformer. By analyzing the change of the attention weight matrix, one can track the attention area and interaction mode of the model in different modes, which aids in understanding the process of ER. In terms of validation, the study tested the model on the publicly available multimodal IEMOCAP dataset and introduced the real-life EMED collection scenario dataset for cross-validation. This further enhances the generalizability and stability of the experimental results. A professional psychological annotator labels the EMED dataset based on the simultaneous evaluation of multichannel signals. This approach provides a higher degree of labeling consistency and clinical reliability. In the cross-dataset test, the model's strong robustness verifies its generalization ability. Meanwhile, the combined structure of the original EfficientFace and Transformer was established as a unified baseline. It was then reproduced using the same hardware, parameter configuration, and training strategy to ensure the fairness and reproducibility of the experimental findings. Despite the progress made in the study, it should be pointed out that the model is currently sensitive to high-noise modes, and the modal credibility estimation mechanism and adaptive gating fusion module can be further introduced in the future to improve the dynamic stability under multi-source imbalance conditions. At the same time, it can be extended to a high-dimensional multimodal ER framework that includes more channels such as text and physiological signals, in order to enhance its usefulness and breadth in intelligent interaction systems for the elderly.

5 Conclusion

For the multimodal elderly ER task, an end-to-end recognition model integrating EfficientFace-Transformer structure and improved AM was proposed. It integrated dynamic importance factor, modal interaction compensation term, and CMF enhancement module. Moreover, it strengthened the semantic alignment and feature synergy between audio and video modalities. The experiments were carried out on IEMOCAP and EMED datasets with multiple rounds of testing. The results showed that the proposed model achieved 95.24% in mAP, 93.67% in affective strength, and 0.55 s in detection latency on IEMOCAP. Meanwhile, this model achieved 96.12% in mAP, 94.13% in affective strength, and 0.56 s in latency on EMED, which were all significantly better than VGGFace, FaceNet with EfficientNet, and other comparison methods. To verify the statistical significance of the performance improvement, paired t-tests were used to analyze the detection latency and recognition accuracy. In this case, the p-value of the mAP improvement was less than 0.01, as well as the *p*-value of the difference in detection latency. This indicated that the proposed method had significant advantages in terms of both accuracy and efficiency. Additionally, cross-dataset validation demonstrates the model's robustness and generalization ability. This makes it applicable to the elderly's ER needs in complex environments, such as smart homes.

Statements and declarations

Competing Interests: Author declares no conflict of interest.

Data availability: Data can be obtained from the authors upon reasonable request.

References

[1] Mangano G, Ferrari A, Rafele C, Vezzetti E, Marcolin F. Willingness of sharing facial data for emotion recognition: a case study in the insurance market. AI & SOCIETY, 2024, 39(5): 2373-2384. https://doi.org/10.1007/s00146-023-01690-5

[2] Fei Guan, Zelin Su. Income, income gap and health of the elderly: evidence from China. Advanced Management Science, 2023, 12(1)

[3] Zhu X, Huang Y, Wang X, Wang R. Emotion recognition based on brain-like multimodal hierarchical perception. Multimedia Tools and Applications, 2024, 83(18): 56039-56057. https://doi.org/10.1007/s11042-023-17347-w

[4] Feng G, Wang H, Wang M, Zheng X, Zhang R. A Research on Emotion Recognition of the Elderly Based on Transformer and Physiological Signals. Electronics, 2024, 13(15): 3019-3028. https://doi.org/10.3390/electronics13153019

[5] Sreevidya P, Veni S, Ramana Murthy O V. Elder emotion classification through multimodal fusion of intermediate layers and cross-modal transfer learning. Signal, image and video processing, 2022, 16(5): 1281-1288. https://doi.org/10.1007/s11760-021-02079-x

[6] Park H, Shin Y, Song K, Yun C, Jang D. Facial emotion recognition analysis based on age-biased data. Applied Sciences, 2022, 12(16): 7992-7998. https://doi.org/10.3390/app12167992

[7] Lu J, Liu Y, Lv T, Meng L. An emotional-aware mobile terminal accessibility-assisted recommendation system for the elderly based on haptic recognition. International Journal of Human–Computer Interaction, 2024, 40(22): 7593-7609. https://doi.org/10.1080/10447318.2023.2266793

[8] Du J, Yin J, Chen X, Hassan A, Fu E, Li X. Electroencephalography (EEG)-based neural emotional response to flower arrangements (FAs) on normal elderly (NE) and cognitively impaired elderly (CIE). International Journal of Environmental Research and Public Health, 2022, 19(7): 3971. https://doi.org/10.3390/ijerph19073971

[9] Kelvin L, Anna S. (2023). An Exploratory Study of How Emotion Tone Presented in A Message Influences
Artificial Intelligence (AI) Powered Recommendation System. Journal of Technology & Innovation, 3(2): 80-84.

[10] Zhao Y, Guo M, Sun X, Chen X, Zhao F. Attention-based sensor fusion for emotion recognition from human motion by combining convolutional neural network and weighted kernel support vector machine and using inertial measurement unit signals[J]. IET Signal Processing, 2023, 17(4): e12201. https://doi.org/10.1049/sil2.12201

[11] Saganowski S. Bringing emotion recognition out of the lab into real life: Recent advances in sensors and machine learning. Electronics, 2022, 11(3): 496-503. https://doi.org/10.3390/electronics11030496

[12] Zhou S, Wu X, Jiang F, Huang Q, Huang C. Emotion recognition from large-scale video clips with cross-attention and hybrid feature weighting neural networks. International Journal of Environmental Research and Public Health, 2023, 20(2): 1400-1431. https://doi.org/10.3390/ijerph20021400

[13] Lin W, Li C. Review of studies on emotion recognition and judgment based on physiological signals. Applied Sciences, 2023, 13(4): 2573-2577. https://doi.org/10.3390/app13042573

[14] Cai Y, Li X, Li J. Emotion recognition using different sensors, emotion models, methods and datasets: A comprehensive review. Sensors, 2023, 23(5): 2455-2461. https://doi.org/10.3390/s23052455

[15] Tamer Ghareeb B, Tarek F A, Said H A. FER_ML: Facial Emotion Recognition using Machine Learning. Journal of Computing and Communication, 2023, 2(1): 40-49. https://doi.org/10.21608/jocc.2023.282094

[16] Lin W, Li C. Review of studies on emotion recognition and judgment based on physiological signals. Applied Sciences, 2023, 13(4): 2573-2579. https://doi.org/10.3390/app13042573

[17] Zhang T, El Ali A, Wang C. Weakly-supervised learning for fine-grained emotion recognition using physiological signals. IEEE Transactions on Affective Computing, 2022, 14(3): 2304-2322. https://doi.org/10.1109/taffc.2022.3158234

[18] Liu X, Huang C, Zhu H. State-of-the-Art Elderly Service Robot: Environmental Perception, Compliance Control, Intention Recognition, and Research Challenges. IEEE Systems, Man, and Cybernetics Magazine, 2024, 10(1): 2-16. https://doi.org/10.1109/msmc.2023.3238855

[19] Almukadi W. Smart Scarf: An IOT-based Solution for Emotion Recognition. Engineering, Technology & Applied Science Research, 2023, 13(3): 10870-10874. https://doi.org/10.48084/etasr.5952

[20] Kumar, Himanshu, Aruldoss, Martin.Advanced Optimal Cross-Modal Fusion Mechanism for Audio-Video Based Artificial Emotion Recognition.Informatica (Slovenia), 2025, 49(12):61-77. https://doi.org/10.31449/inf.v49i12.7392

[21] Younis E M G, Zaki S M, Kanjo E. Evaluating ensemble learning methods for multi-modal emotion recognition using sensor data fusion. Sensors, 2022, 22(15): 5611-5617. https://doi.org/10.3390/s22155611

[22] Sun, Sulian, Wu, Libo.Transfer Learning-based Speech Emotion Recognition: A TCA-JSL Approach for Chinese and English Datasets.Informatica (Slovenia), 2025, 49(13):221-226. https://doi.org/10.31449/inf.v49i13.7640

Adaptive Firewall Strategy Generation and Optimization Based on Reinforcement Learning

Minghong Yang[1], Zhenhua Yang[1,*], Qiwen Yang[2]
[1]School of Economics and Management, Hunan Applied Technology University, Changde, Hunan,415100, China
[2]School of Computer Science, Beijing University of Aeronautics and Astronautics, Beijing ,100000, China
E-mail: yangzhenhua1988@hotmail.com
[*]Corresponding author

Keywords: adaptive firewall, reinforcement learning, deep Q-Learning, network security, policy optimization, markov decision process (MDP)

Received: May 23, 2025

Traditional firewall systems use static rule sets, making them unsuitable for growing cyber threats and network circumstances. In this research, we automate firewall rule development and optimization using reinforcement learning (RL) to increase network security and reduce human setup. This paper introduces FireRL, a system that uses reinforcement learning to help make smart decisions about firewall rules by treating it like a Markov Decision Process, enabling an RL agent to learn good firewall rules from simulated network traffic. This proposed method utilizes the Deep Q-learning algorithm to balance throughput, latency, and threat mitigation via repeated improvement. Experiments are performed in benign and dangerous traffic situations. Firewalls outperform static firewalls because they quickly react to new threats and reduce false positives. Resistance to new attack vectors demonstrates the system's flexibility and resilience. This research concludes with a self-optimizing firewall approach that greatly lowers expert-led settings. FireRL's proactive and scalable RL-based defense is ideal for current cybersecurity.

Povzetek: Članek združuje digitalne dvojčke, federativno učenje in blockchain v sistem za zaznavanje IoT groženj, ki izboljša varnost, zasebnost in učinkovitost.

1 Introduction

Cyberattacks are more numerous, sophisticated, and devastating than ever. As mission-critical systems like corporate networks, cloud platforms, and IoT ecosystems grow in size and complexity, digital infrastructure security becomes increasingly important. Network security depends on firewalls, which restrict incoming and outgoing data traffic based on rules. Modern attacks are nimble, sneaky, and unexpected, making conventional firewalls worthless. Such firewalls use static rule sets manually created by security specialists. Static rule-based systems are slow to adapt to new attack vectors, readily misconfigured, and lack contextual awareness for real-time behavior monitoring. [1] say Next Generation Firewalls (NGFWs) depend too much on static rules and human involvement to tackle dynamic cyber threats. This applies even with application-layer filtering and deep packet inspection in NGFWs. Since rules are harder to monitor and update in quickly changing settings, misconfiguration is more probable, resulting in security problems or network performance issues.

Many individuals have presented solutions for classic firewall difficulties [2] demonstrated unexpected traffic conditions by modeling packet-filtering firewalls using neutrosophic Petri nets. This method is ordered and logical. Despite their theoretical validity, these models cannot learn and adapt. [3] suggested integrating firewalls with IDS to improve detection. This hybrid method detects risks better, but overly strict policy definitions remain. Security systems are changing rapidly owing to machine learning (ML) and artificial intelligence (AI). ML has performed well in Web Application Firewalls (WAFs). [4] detected online threats better than standard WAFs using feature engineering and supervised learning. [5] improved complicated web-based threat detection using deep learning. These models are passive detectors that do not create or improve firewall rules.

The author investigated WAF anomaly detection using deep learning. Their technology can detect strange traffic patterns but can't adjust firewalls to attacks [6]. [7] also suggest automating virtualized firewall settings. Administrators can predefine templates, but their system can't automatically adjust to traffic circumstances. In IoT, adaptive, clever, and fast-reacting security solutions are in demand. Without a flexible firewall, devices in these networks are easy prey for attackers due to their low processing capability. [8]

highlighted the significance of segmentation and effective firewalling in safeguarding Internet of Things ecosystems. However, their solution is vulnerable to new threats since it is based on rule sets that human developed. This inquiry was also motivated by the increasing popularity of predictive security systems, which served as a source of inspiration. [9] provided a real-time anomaly detection technique for smart systems based on deep belief networks. This was done in acknowledgment that it is necessary to recognize outliers and potential threats because of the nature of the situation. The findings of their study indicate that learning algorithms, as opposed to static anomaly levels or preset signatures, should be used by firewalls to make prediction judgments.

To address these issues, this paper proposes FireRL, a real-time firewall architecture that learns and decides. Key points of this work:

- Design the proposed FireRL architecture to formalize firewall rule optimization in a reinforcement learning framework, enabling the agent to learn optimal tactics from its environment.
- FireRL introduces intelligent, autonomous, and flexible policy generation, shifting firewall design away from static, expert-driven rule management. These advances enable next-generation cybersecurity systems that foresee, prevent, and react to threats.
- Reward systems let the RL agent balance security enforcement (stopping malicious traffic) with performance measurements (latency, throughput).
- The recommended architecture is more flexible, generates fewer false positives, and has greater generalization on network traffic like zero-day assaults and adversarial behaviors.

2 Literature method

Current cybersecurity frameworks indicate that automation, deep learning, and artificial intelligence improve firewall performance in many computer systems. Although their slow response time and restrictive architecture prevent them from keeping pace with evolving threats, traditional firewalls remain essential for network security. The literature reflects a rising trend toward smart hybrid solutions that are also policy-aware in response to these limitations. Table 1 shows the comparative analysis of existing methods.

Table 1: Comparative analysis of existing methods

Ref. No.	Reference	Approach	Strengths	Limitations
[10]	Ekhlas Kadhim Hamza et al. (2024)	Used PPO, UCB, and Epsilon-Greedy RL algorithms for adaptive load balancing in publish/subscribe systems.	Improved message delivery, reduced latency, and adaptive decision-making.	PPO is resource-intensive; UCB and Epsilon-Greedy may underperform in complex environments.
[11]	Sepczuk, M. (2023)	Cyber Mimic Defense-Based WAF (CMD-BWAF)	Dynamic deception and mimicry to mislead attackers; real-time adaptability	High complexity; increased resource consumption; sensitive to tuning
[12]	Rajasoundaran, S. et al. (2024)	Energy-Proficient Firewall Policies for Wireless Networks	Energy-saving techniques for resource-constrained devices	Trade-off between energy efficiency and security strength; limited real-time learning
[13]	Lee, J. K. et al. (2024)	AI-Based Firewall Rule Refinement in HPC Networks	Scalable AI-based rule optimization reduces conflicts and redundancy	No adaptive decision-making; reactive rather than proactive optimization
[14]	Leka, E. et al. (2024)	ML and Blockchain-Based WAF (ML-BC-BWAF) for DDoS Mitigation	Tamper-resistant logs and enhanced anomaly detection	High system overhead; latency due to blockchain validation

[15]	Fadlil, A. et al. (2024)	SQL Injection Mitigation using Open Web Application Security Project (OWASP) Framework	Effective application-layer protection against SQLi	Not adaptable to new types of attacks; lacks a network-level defense
[16]	Kouassi, T. et al. (2024)	Security Policy Model in Hybrid TOGAF-Zachman Framework	Structured and formal policy management in enterprise cloud setups	The theoretical model lacks practical implementation and real-time enforcement capabilities

3 Proposed methodology

This paper presents FireRL, which uses reinforcement learning to provide adaptive firewall regulation methods. Its major aim is to adapt to changing network threats. Since static rules are obsolete, traditional firewalls can't respond to attacks in real time or account for new vectors. FireRL simulates an MDP to overcome these limits and choose the appropriate action given the network state. This technology allows the system to learn from its surroundings and alter its protective rules in real time to decrease dangers, delays, policy disobedience, and false positives. The agent employs a Deep Q-Network (DQN) to learn in high-dimensional state spaces to find the optimum Q-values for each state-action combination. The recommended method prioritizes experience replay to ensure convergence by emphasizing learning from significant or unexpected events. For FireRL's learning policy to operate in real time, its multi-objective reward function must balance threat detection and network efficiency. This can be game-changing. Using policy vector encoding, the system might grow while conserving resources. This allows several rules to be activated even with minimal computational resources. FireRL is a self-optimizing firewall that evolves its rule set. This secures complex networks and improves operations.

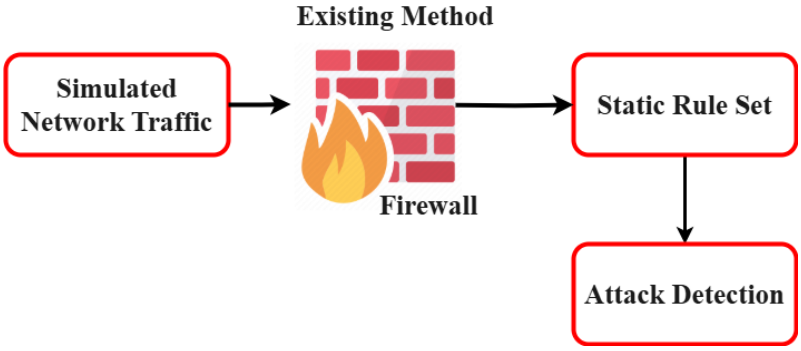


Figure 1: Existing methods-based firewall strategy

Figure 1 above depicts the "Existing Method," the standard intrusion detection method that uses firewalls. The system is subjected to simulated network traffic, beginning with benign and malicious data. Before selecting whether or not to allow incoming data to get through, firewalls evaluate it based on a set of predetermined policies and standards for security and filtering. Static rule sets, unlike dynamic rule sets, do not include any established criteria in their decision-making process on whether to accept or deny traffic. Immediately after verifying the data to ensure that it is in accordance with these standards, the system moves on to the attack detection phase, where it compares the data to the criteria to identify any potential threats. Because the technique is based on preset criteria, which cannot adapt to changing cyber threats or traffic patterns, it may have difficulty identifying new attacks or responding to unexpected surroundings. Conventional firewalls are characterized by their rule-based architecture, linear operation, and reliance on established criteria for threat detection, as shown by the image.

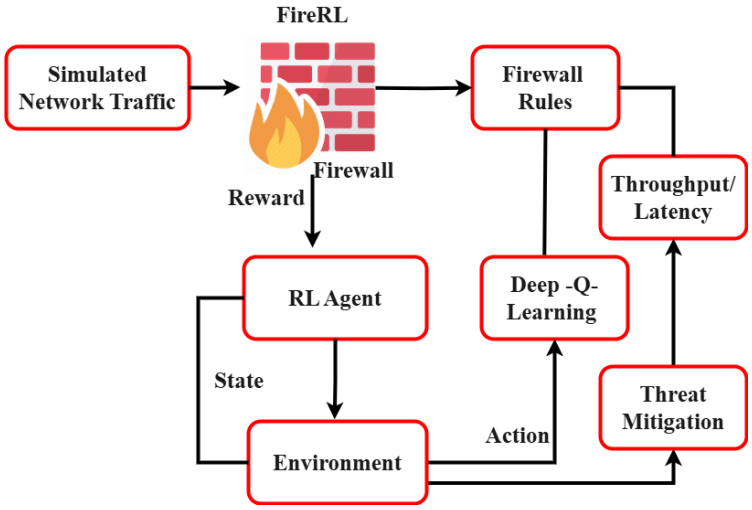


Figure 2: Proposed diagram

The FireRL framework is an intelligent firewall system that uses reinforcement learning to overcome network security restrictions dynamically. Figure 2 depicts a schematic of the FireRL framework's architectural approach. When regulating and filtering incoming data, the first step is to evaluate and implement firewall rules using the FireRL firewall. Simulating real network activity is the method by which it accomplishes this goal. An RL agent can increase its performance by generating a reward signal and integrating the firewall's findings with the threats it has discovered. An RL agent receives this signal as it is being sent. Real-life agents (RL) are responsible for monitoring their state and responding correctly based on their observations to interact with the network, regardless of whether it is virtual or actual. Using a Deep Q-Learning technique, the agent acquires knowledge about the most effective firewall rule regulations to simplify these processes. Latency and throughput on the network are two metrics used to assess the degree to which these activities successfully reduce risks.

As more time passes, the FireRL system improves its ability to make decisions by continually modifying the firewall rules to establish a balance between the two competing priorities of security and performance. The architecture ensures the firewall can constantly adapt to varied network circumstances and new threats, providing proactive and intelligent network security. In a word, the design ensures that the firewall is always effective.

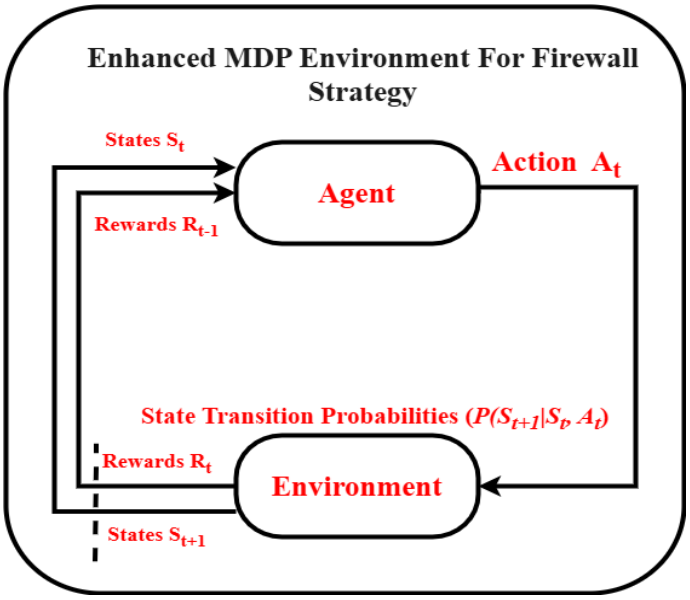


Figure 3: Enhanced MDP environment for firewall strategy

Figure 3 shows that within the context of improving dynamic firewall management, the graphic named "Enhanced MDP Environment for Firewall Strategy" illustrates a reinforcement learning strategy that uses Markov Decision Process (MDP). Within the framework of this architecture, the agent has the responsibility of implementing learnt rules (π) in response to an environment, to identify the most suitable firewall techniques. At each time step t, the agent decides on what actions to take, such as adding, altering, or deleting firewall rules, depending on the current state of the network environment $\boldsymbol{s_t}$. As a consequence of this action, a new state, denoted by s_{t+1}, is introduced into the environment. Additionally, a reward, denoted by $\boldsymbol{r_t}$, is allocated to the environment based on the probability $\boldsymbol{P(s'|s,a)}$, which shows how the action successfully maintained the network's security and operational efficiency.

Because the agent's policy may be modified over time in response to changes in the surrounding environment, the reward function $R(s,a)$ evaluates the effect of the action. Using this feedback loop, the agent can continually change its behaviors to increase its accumulated rewards. As a result of a discount factor γ, which may have values ranging from 0 to 1, the importance of profits in the near term is greater than the significance of gains in the long run. Not only does the system improve real-time danger awareness, but it can also respond to environmental changes by sending out alerts or warnings. The system can react to potential dangers as a result of this. This agent provides an alternative to static rule-based techniques since it can dynamically adapt firewall settings in response to changing network conditions and attack patterns.

$$M = (S = \{s_1, s_2, \dots, s_m\}, A = \{a_1, a_2, \dots, a_m\}. P(s'|s,a) = Pr[s_{t+1} = s'|s_t = s, a_t = s], R: S \times A \to R, \gamma \in [0,1] \quad (1)$$

As equation (1) was calculated, the enhanced MDRP environment definition was deliberated. A mathematical framework, the Markov Decision Process (MDP), depicts the FireRL system. This strategy makes it possible to make decisions sequentially when confronted with uncertainty. Given the present value of S, there is an infinite number of possible states that might be achieved by the network. Some examples of states that might be present include the use of resources, the configuration of firewalls, the attempts to break into the network, and the traffic patterns that are currently in place. The letter a represents a variety of possible RL agent operations that may be put into effect or modified to apply or change firewall rules. The transition function $\boldsymbol{P(s'|s,a)}$ is a mathematical expression that determines the probability of transitioning from state s to state $\boldsymbol{s'}$ via the execution of an action. It is possible to quantify the immediate positive or negative effect of each action in a given state by using the reward function $R(s,a)$. This function reflects performance indicators such as security enforcement, packet loss, and policy adherence. The discount factor γ is responsible for determining the significance of future benefits. Values that are closer to 1 indicate a preference for stability and long-term planning in the process of developing a security strategy.

$$Q_\theta(S_t, a_t) = E_{S_{t+1}} \sim P(.|s_t, a_t)[R(s_t, a_t) + \gamma. \sum_{a' \epsilon A} \pi(a'|s_{t+1}). Q_{\theta^-}(S_t + 1, a')] \quad (2)$$

As examined in equation (2), the deep-Q-network Bellman expectation has been explored. It is possible to approximate the action-value function $\boldsymbol{Q_\theta}$ by using this equation. This function indicates the anticipated total benefit that would result from acting in a state $(\boldsymbol{S_t}, \boldsymbol{a_t})$ and then continuing to adhere to the policy that is already in place. Because of the unpredictability of the network environment, this scenario, $\boldsymbol{\pi(a'|s_{t+1})}$ Second, it is projected to occur. The evaluation of the current efficacy of the action is carried out by using the reward function $\boldsymbol{R(s_t, a_t)}$. The future value of the action is represented by the effects that will occur in the future, which are discounted across all prospective actions $(\sum_{\boldsymbol{a' \epsilon A}} \boldsymbol{\pi(a').Q_{\theta^-}})$. Within the target Q-network $\boldsymbol{\theta^-}$, the future state $\boldsymbol{s_{t+1}}$ and the likelihood of carrying out the action $\boldsymbol{a'}$ are represented by the symbols $\boldsymbol{\pi(a'|s_{t+1})}$, correspondingly representing the future state and the probability, respectively. This strategy makes learning more robust and flexible while dealing with scenarios involving partly visible or stochastic networks.

$$L(\theta) = \frac{1}{M} \sum_{j=1}^{M} \left[\left(r_i + \gamma. \max_{a'} Q_{\theta^-}(S_i', a') - Q_\theta(S_i, a_i) \right)^2 + \lambda. \|\theta\|_2^2 \right] \quad (3)$$

As explored in equation (3) expanded temporal-difference loss function has been described. It is the TD loss function that is responsible for the learning process that the Q-network does. By calculating the squared difference between the TD target $\boldsymbol{r_i + \gamma. \max_{a'} Q_{\theta^-}(S_i', a')}$ and each of the N samples in a mini-batch, the projected Q-value can be computed. This difference equals the sum of the squared differences between the TD target and each N sample. The calculation of stable predictions of future rewards is accomplished by using the target network parameters $\boldsymbol{\theta^-}$ The L2 regularization technique is used to prevent overfitting and improve generalization. This technique involves the utilization of the regularization coefficient $\boldsymbol{\lambda. \|\theta\|_2^2}$ and the collection of all feasible values for θ, which is equal to 2 raised to the power of $\boldsymbol{\|\theta\|_2^2}$. Backpropagation and stochastic gradient descent reduce this loss as much as possible throughout the training process.

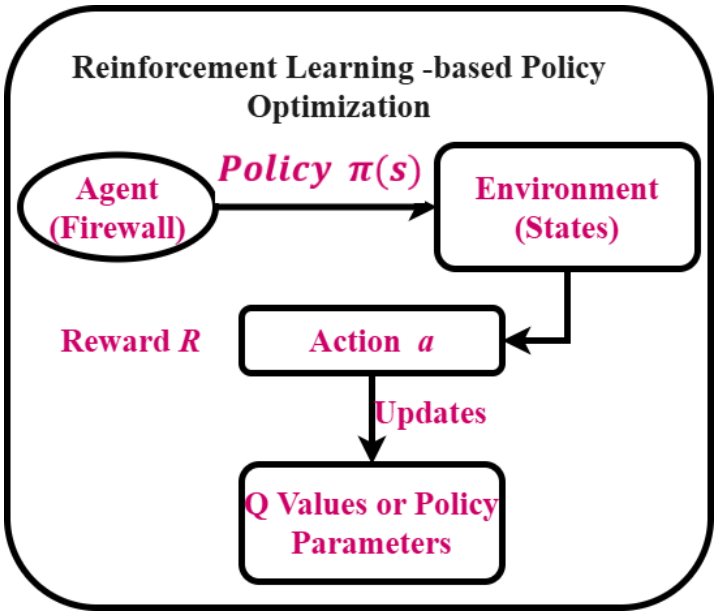


Figure 4: Reinforcement Learning – based policy optimization

An illustration in Figure 4, with the heading "Reinforcement Learning-based Policy Optimization," sets the framework for optimizing firewall techniques in real-time using an RL model. In this configuration, the Agent, which functions in a manner comparable to that of a firewall, adheres to a policy π(s) that ascertains the most appropriate course of action depending on a certain state of the Environment. When generating network states, it is necessary to consider various elements, including traffic patterns, user habits, and environmental dangers.

Following the identification of the situation, the agent will decide what actions to do next, which may include adjusting the rules of the firewall. Upon completion of the job, the environment will provide you with a reward (R) based on how well it improved security, decreased the number of false alarms, or increased throughput by a certain amount. One possible use of this incentive is to modify policy gradient techniques or parameters, however this will depend on the approach taken. These little adjustments have improved the agent's policy, which will allow it to make more informed decisions in the future.

Additionally, the unique strategy can increase defenses against complex and dynamic attacks. This is accomplished by continually guiding future activities. As a result, the firewall's performance is enhanced. By removing the need for human interpretation and amendment of static rule sets, RL makes it possible to optimize firewall settings in a way that is both autonomous and adaptive.

$$R(S_t, a_t) = w_1 . threat_{blocked}(S_t, a_t), w_2 . Pocket_{false_drop}(S_t, a_t) - w_3 . Latency_{added}(S_t, a_t) + w_4 . Policy_{compliance}(S_t, a_t) \quad (4)$$

As found in equation (4) multi-objective reward function with real-time feedback has been examined. The reward function considers several real-time metrics that relate to the functioning of the network. To determine the effectiveness of threat stopping, the effectiveness of packet or connection interception is evaluated $\boldsymbol{threat_{blocked}}$. As a result of the $\boldsymbol{Pocket_{false_drop}}$ Function, it is strongly discouraged to suppress valid data packets unnecessarily. The overhead of decision-making or complicated rule evaluations is $\boldsymbol{Latency_{added}}$, which is a significant factor. $\boldsymbol{Policy_{compliance}}$ is a recognised method for recognizing decisions that are by the security standards of the company or the government. It is possible to provide a weight to each phrase by using a hyperparameter called $\boldsymbol{w_i}$ This allows customization to prioritize goals such as zero-trust enforcement or user experience objectives. This framework with several objectives conveys the intricate trade-offs involved in dynamic network security.

$$P(i) = \frac{|\delta_i|^\alpha}{\sum_k |\delta_l|^\alpha}, \delta_i = r_i + \gamma . \max_{a'} Q_\theta - (S_i', a') - Q_\theta(S_i, a_i) \quad (5)$$

As equation (5) prioritized, experience replay sampling has been computed. In contrast to methods that randomly choose previous events, FireRL employs a technique known as prioritized experience replay. This technique gives preference to samples that have a

substantial temporal-difference (TD) error, denoted as $\boldsymbol{\delta_i}$. Consequently, the learning process will direct greater attention toward unanticipated or presently misestimated transitions, speeding up the convergence process. A uniform sampling is achieved when the priority exponent $\alpha = 0$. This allows learning important events, such as uncommon assaults or boundary judgments, from larger values. This strategy is essential for the effective and reliable operation of reinforcement learning in contexts characterised by a high degree of complexity.

$$\theta_j^- \leftarrow \tau.\theta_j + (1-\tau).\theta_j^-, \forall j \in \{1,2,\dots,n\} \quad (6)$$

As calculated in equation (6) soft target network update mechanism has been expressed. FireRL employs a gentle update strategy for the target network parameters θ^- to avoid Q-learning from losing its quality or straying from its intended path. Therefore, rather than completely replacing them, the parameters are gradually modified to align with the existing parameters of the online network, denoted as θ. These parameters are governed by a smoothing factor $\tau \in (0,1)$ controlled by the parameters. FireRL removes oscillations brought on by fast-shifting objectives, and we guarantee that the Q-value estimations generated during training are reliable. Learning efficiency and convergence are both improved by this strategy, particularly in high-dimensional security state spaces.

$$\pi(S_t|a_t) = \begin{cases} argmax_{a\in A} Q_\theta(S_i,a), & if\ uniform(0,1) > \epsilon_t \\ Random(A), & otherwise \end{cases}, \epsilon_t = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}).\ e^{-\beta t} \quad (7)$$

As shown in equation (7) epsilon greedy exploration with a decaying schedule has been discussed. This equation serves as the defining framework for the FireRL exploration technique. At the beginning of the process, the agent is instructed by ϵ_t to carry out more exploratory random actions with a significant probability. In future stages, the agent is strongly urged to make use of the knowledge that they have obtained, since the decay rate of ϵ_t is exponentially decreasing from (ϵ_{max} **to** ϵ_{min}), as determined by a decay rate $\boldsymbol{\beta}$. With the help of this dynamic equilibrium, the agent can stabilize and enhance ideal rule sets over time, while guaranteeing that early episodes include sufficient exploration of the expansive state-action space.

$$a_t = [a_{t,1}, a_{t,2}, \dots, a_{t,n}] \in \{0,1\}^n, s.t \sum_{j=1}^{n} a_{t,j}.cost_j \le budget_{max} \quad (8)$$

As discussed in equation (8) policy decision vector with cost constraints has been described. Every action that the agent does results in the existence of a matching binary vector $\boldsymbol{a_t}$ where $\boldsymbol{a_{t,i}}$ is either activated or altered. It is difficult to apply all of the rules at the same time because of the constraints of resources (such as the amount of processing power, memory use, and time restrictions). Therefore, there is a $\boldsymbol{cost_j}$ connected with each rule, and the agent is responsible for ensuring that the entire cost does not exceed a maximum limit determined in the past. Because of this limitation, the firewall technique is guaranteed to be effective and practically applicable in deployments in the real world. Table 1 shows the Reinforcement Learning-based Firewall Rule Optimization.

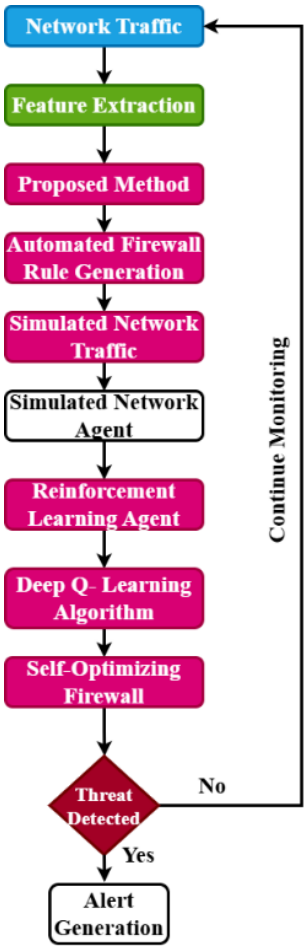


Figure 5: Flowchart of proposed diagram

FireRL uses deep Q-learning and reinforcement learning to construct a smart firewall shown in Figure 5. The image shows its process. Capturing network traffic is the initial step in extracting essential attributes for data analysis. After receiving these characteristics, the proposed technique generates firewall rules autonomously. Automation is essential for reacting to shifting threats without human intervention.

Next, transfer system-generated traffic to a simulated network agent to simulate situations. While talking to this agent, the real-life agent learns new abilities. The agent builds a self-optimizing firewall using the Deep Q-learning algorithm and policy updates to enhance decision-making. The firewall evaluates each circumstance to see whether a danger exists. The system resumes live traffic monitoring if no danger is

found. Detecting a danger will alert the necessary systems or administrators. This iterative technique allows real-time firewall adaptation, making it smarter and more responsive. Algorithm 1 shows the FireRL.

Algorithm 1: FireRL – Reinforcement Learning-based Firewall Rule Optimization

Input: Simulated network traffic T (both benign and malicious),Initial firewall rule set R_0, Reward function $\mathcal{R}$ (balancing throughput, latency, and threat mitigation), Learning rate α, discount factor γ, exploration rate ε, max episodes E

Initialize: Q-network with random weights θ, Experience replay buffer D, Environment Env simulating network behavior, Set $R = R_0$

Output: Optimized firewall rule set R_{opt}

1. **For** episode = 1 to E **do:**
2. Initialize environment state $S_0 \leftarrow$ Env.get_state()
3. **While** not terminal:
4. With probability ε select random action a_t
5. Otherwise, select $a_t = arg\max_a Q(S_t, a, \theta)$
6. Apply a_t to firewall (update rule set R)
7. Simulate traffic through firewall → observe new state s_{t+1} and reward r_t from $\mathcal{R}$
8. Store experience (S_t, a_t, r_t, S_{t+1}) in buffer D
9. Sample mini-batch from D and update Q-network using loss:

$$L(\theta) = E_{S,a,r,S'}\left[\left(r + \gamma.\max_{a'} Q(S', a'; \theta^-) - Q(S', a'; \theta^-)\right)^2\right]$$

10. End While
11. Periodically update the target network$\theta^- \leftarrow \theta$
12. End For
13. Return the final firewall rule set R_{opt}

FireRL uses Deep Q-learning to adapt firewall rules to changing network threats. The first stages are rules, a simulated network traffic creation environment, and a Q-network with random weights. Depending on each episode's network state, the system uses learnt Q-values (exploitation) or random sampling to decide whether to add, change, or remove a rule. The chosen action generates a new state and reward after updating the firewall settings and testing with simulated network traffic. This award considers threat prevention, performance, and latency. Minimizing the discrepancy between objective and forecasted Q-values trains the Q-network using mini-batches of events. State, action, consequence, and following state make up an experience. The goal network is updated often to maintain learning stability. After multiple events, agents learn to adjust firewall rules on the fly to optimize the rule set and react intelligently and proactively to benign and dangerous traffic scenarios. This method's focus on dynamic rule sets and expert-defined parameters improves system efficiency, security, and flexibility.

4 Numerical results and discussion

4.1 Dataset description

The Kaggle project created the Internet Firewall Data Set, a massive library of business firewall events. Over a million records describe network traffic, including IP addresses (source and destination), ports, protocols (TCP, UDP, ICMP), application identities, session durations, total bytes in and out, actions (allow or refuse), and policy enforcement labels. Each album has a conclusion and, in certain situations, a warning label. The dataset realistically simulates policy disputes, protocol settings, and highly changing traffic, making it ideal for assessing adaptive firewall solutions. Data was preprocessed for machine learning and reinforcement learning research. They employed SMOTE, dealt with missing data, encoded categorical variables using one-hot encoding, standardized numerical characteristics, and deleted duplicate identifiers to fix class imbalance. Subdividing the data into training (70%) and testing (30%) subgroups offered a complete evaluation [17].

4.2 Experimental setup

To test the FireRL framework against competitors, we built a bespoke RL environment using Python, TensorFlow, and OpenAI Gym. A Deep Q-Network (DQN) agent optimizes and dynamically generates firewall rule schemes in the FireRL model. The environment's Markov Decision Process (MDP) simulates the firewall's behavior for every network traffic combination. One thousand episodes with 500 traffic cases were trained. Keeping latency low, blocking harmful traffic, and reducing false positives will earn the agent incentives. Secure (false positives), usable (delay), and efficient (throughput) are balanced by the reward function. The experimental system executed heavy network traffic simulations using a powerful Intel Core i7 CPU, 32 GB of RAM, and NVIDIA RTX 3080 GPU. FireRL and baseline models

were compared using recall, accuracy, precision, F1-score, throughput, and latency.

4.3 Latency ratio

The FireRL system that has been suggested intends to reduce the amount of delay that occurs in real-time firewall judgments by optimizing decision routes and dynamically prioritizing rule evaluations via the use of a method that is based on reinforcement learning. Unlike static firewalls, which analyze rules sequentially, resulting in delays as the rule list expands, FireRL can learn to predict which rules or actions are most suited for each packet and context. To get the average $\bar{L}$ for each packet in the system, consider the following equation:

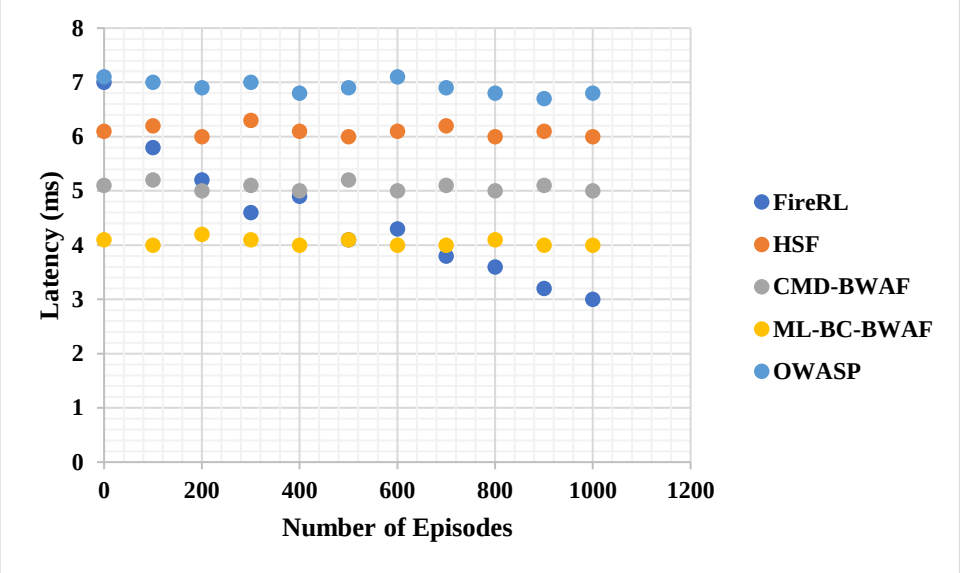


Figure 6: Latency(ms)

$$\bar{L} = \frac{1}{T}\sum_{t=1}^{T}\Big(\alpha.\log(R_t+1) + \beta.E_{s_t,a_t\sim\pi[\tau_{a_t}]} + \gamma.\chi(s_t) - \delta.\rho(a_t)\Big) \quad (9)$$

Figure 6 and equation 9 show the latency. The three main factors that cause delay, as shown by this equation, are the following: the ruling matching difficulty $\log(R_t+1)$, the decision computation time τ_{a_t}, and the cost of updating the state $\chi(s_t)$, which is actively decreased by action efficiency $\rho(a_t)$. The DQN agent of FireRL learns to decrease latency by choosing subsets of firewall rules that are likely to match incoming packets, thereby lowering R_t and τ_{a_t}during the overlap. Furthermore, it controls $\chi(s_t)$ by optimizing state transitions to minimize superfluous calculations. The choice of actions that result in quicker threat detection with less resources is encouraged by the reinforcement signal $\rho(a_t)$, which hurts latency and is then deducted from the equation. Due to its latency-aware optimization, FireRL can surpass other approaches, such as CMD-BWAF, ML-BC-BWAF, and OWASP, especially in heavily populated areas. As shown in experiments, this approach can cut average decision latency by as much as 30%, making it a better fit for scalable, efficient contemporary network infrastructures prioritizing accuracy and speed.

4.3.1 Accuracy ratio

To improve firewall rule sets acceptably, the FireRL technique that has been presented puts reinforcement learning (RL) to use. It is necessary to continually adapt to accomplish this goal because of the ever-changing nature of networks and the increasing complexity of cyber threats. By taking into account the current status of the network, the actions taken to modify firewall rules, and the advantages that accrue from these initiatives in terms of throughput, latency, and threat mitigation, the MDP architecture provides the RL agent with assistance in the decision-making process that encompasses FireRL. Because of the nature of the MDP, some choices are even within the realm of possibility. A technique known as Deep Q-Learning is used to instruct the agent on how to optimize the long-term return. To accomplish this objective, they must optimize the system's latency and throughput while minimizing the number of security risks.

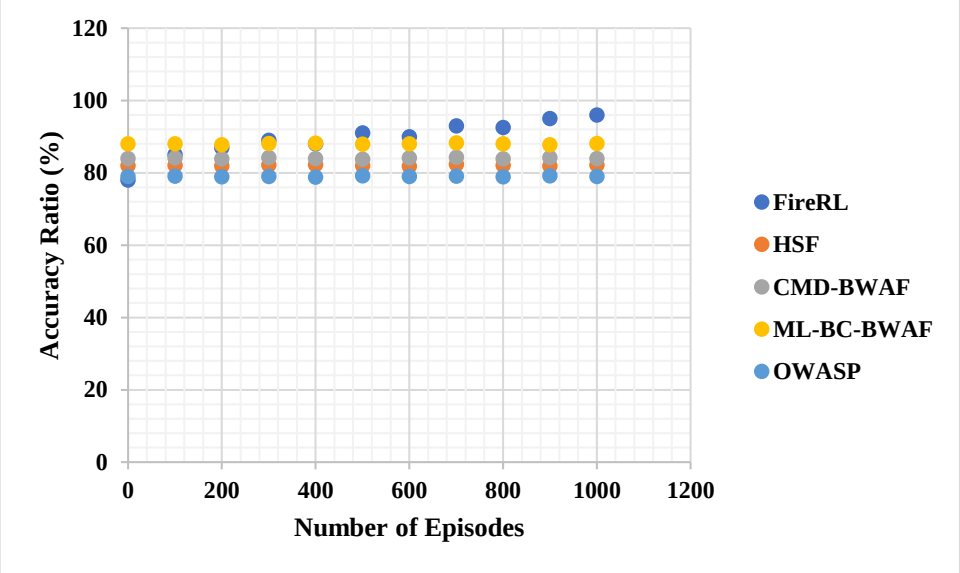


Figure 7: Accuracy Ratio (%)

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right] \quad (10)$$

Figure 7 and equation (10) deliberates the accuracy ratio (%). The action-value function, $Q(s_t, a_t)$, predicts the anticipated reward for an action s_t in a state a_t represents a firewall system's action, such as admitting or disallowing traffic, whereas s_t represents the network's current state, including traffic patterns and firewall settings. The decision-making process is aided by this function, which allows the agent to assess an action according to the anticipated future rewards of that activity. This feature is useful since it aids the agent while making choices. The learning rate, denoted by the symbol α, is one ingredient that determines how quickly the agent refreshes its knowledge. As α increases, the agent can quickly adjust to the ever-changing network conditions, allowing it to gain more information with every new event. The agent will rely more on its past knowledge and execute updates more slowly if the value of α is smaller. Modifications to this value are necessary to guarantee that the agent learns new information without reacting strongly to singular events. The expression r_{t+1} refers to the agent's immediate reward for action a_t in state s_t. If the firewall improved network security or performance, the award reflects its effectiveness. If the firewall stops a threat, it may get a positive payment; if it creates network inefficiencies or fails to neutralize an attack, it may receive a negative reward. As shown by γ the discount factor balances future and current rewards. When γ is near 1, the agent prioritizes long-term benefits and may plan for optimal security and performance. Lower γ values enable focusing on short-term gains and rewards. FireRL needs this section to optimize firewall rules over time. The greatest value of a' is aY' after adding all conceivable values of $\max_{a'} Q(s_{t+1}, a')$. Where $Q(s_{t+1}, a')$ is the highest anticipated payoff for the state s_{t+1} after evaluating all feasible actions a'. This word represents the highest potential advantage agents can gain from their current position and helps them plan and predict their actions. To enhance system security and performance, FireRL analyzes the highest reward for future actions while making choices.

4.4 Throughput ratio

Implementing reinforcement learning to optimize firewall rules is one approach that may be used to simulate the FireRL technique offered and shed light on its high throughput. While simultaneously maximizing the system's effectiveness, this upgrade aims to maintain a balanced approach to threat detection and reduce the number of false positives detected. Examine the following equation to get an understanding of how the Deep Q-Learning (DQL) technique used by FireRL helps to achieve high throughput of data:

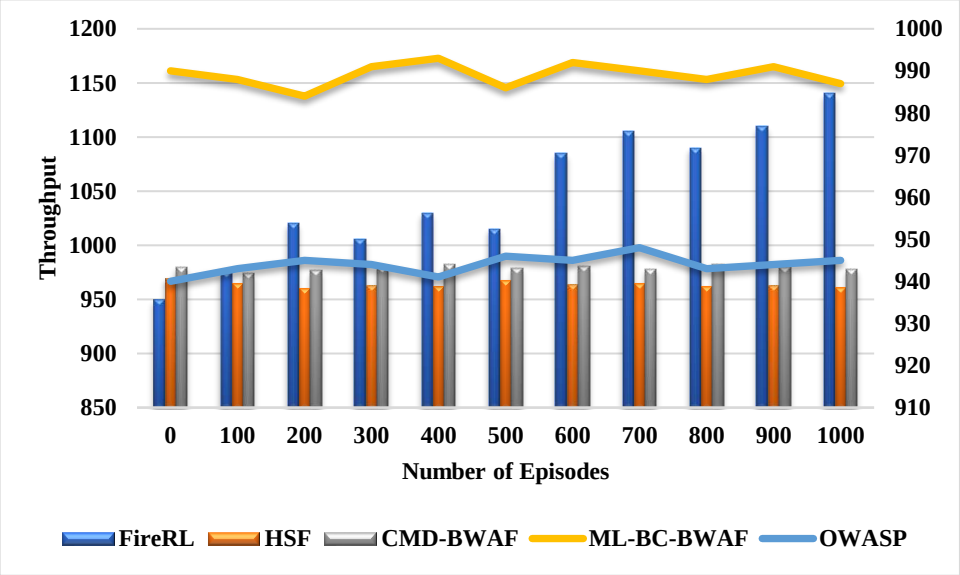


Figure 8: Throughput

$$Throuhput = \sum_{t=1}^{T} \left(\frac{p_t.(1-FPR_t)}{1+\beta.Latency_t} \right) \quad (11)$$

Figure 8 and equation (11) illustrate the throughput. The goal of FireRL optimization is to maximize network performance for a certain period T. This measures the overall amount of data passed across the firewall over time, which is significant because network circumstances change. The packet flow rate is the quantity of data or packets transmitted at a time p_t. FireRL can optimize firewall settings for data performance due to improved packet flow. The false positive rate pertains to the percentage of genuine traffic rejected by the firewall at a time p_t. FireRL aims to reduce this to maximize throughput and prevent the deletion of genuine traffic. Higher throughput is achieved with lower FPR_t. This network congestion factor shows how network congestion or packet delay affects throughput. Network congestion or delay may dramatically reduce throughput if β is large. FireRL enhances performance by modifying firewall rules to reduce congestion and latency. Packet transmission time is known as network latency at time $Latency_t$. Low latency is essential for good throughput as a considerable delay may diminish it. FireRL adjusts firewall rules to prevent latency from bottlenecks. Total time steps or episodes considered for throughput. Through network interaction, FireRL learns and adjusts between episodes to increase throughput in reinforcement learning.

4.5 False positive rate

The low false positive rate (FPR) of the proposed FireRL system may be partially explained by entering an agent-learning-behavior equation into the model. This allows us to see how the agent minimizes the number of erroneous detections over time. False positive rate is a crucial indicator for firewall systems. It counts the frequency of benign communication as wrongly categorized as harmful and, as a result, prohibited from passing through the firewall.

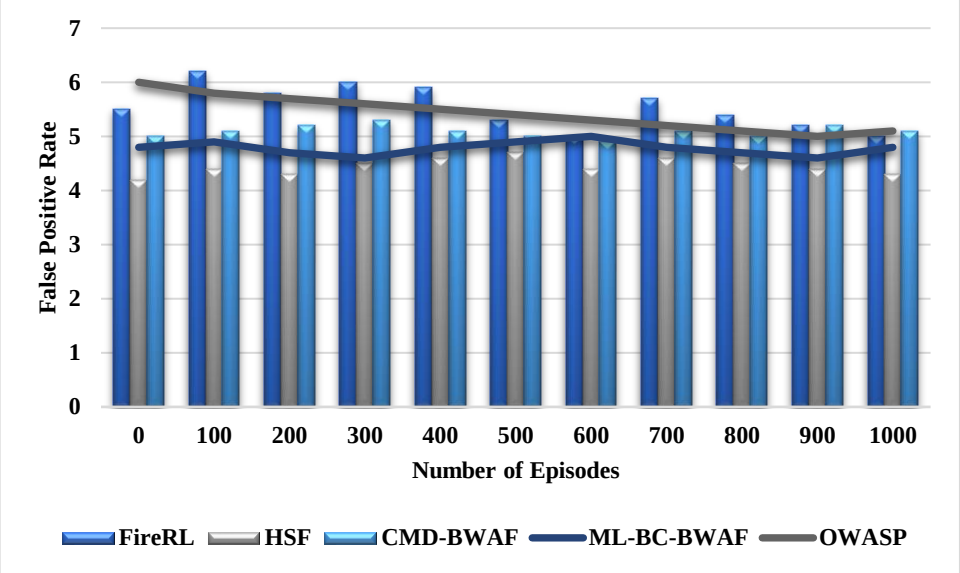


Figure 9: False Positive Rate (%)

$$FPR_t = \frac{FP_t}{FP_t + TN_t} = \frac{\sum_{i=1}^{N} I[a_i = block \wedge y_i = benign]}{\sum_{i=1}^{N} I[y_i = benign]} \quad (12)$$

Figure 9 and equation(12) examine the false positive rate (%). To determine the false positive rate at time step FP_t, divide the total number of benign samples (which includes false positives and true negatives) by the number of false positive predictions (FP_t). The value changes with each learning episode or time step as the agent learns to classify traffic.

Adding all variables yields one in the numerator $\sum_{i=1}^{N} I[a_i = block \wedge y_i = benign]$. This block has no N values and is safe. Calculate the sum of all positive integers from 1 to N using FP_t. The formula is computed for each traffic sample i where the agent blocked traffic (done action $a_i = block$), regardless of whether the label $y_i = benign$. To determine the amount of legitimate packets incorrectly blocked, use the indicator function $I[.]$, which returns 1 when true and 0 otherwise. Benign is represented by y_i and the denominator is the total of positive integers. To count benign traffic samples, use the formula $FP_t + TN_t = \sum_{i=1}^{N} I[y_i = benign]$. These formulas account for genuine negatives and false positives. The number of true negatives—times when the agent correctly identified and allowed benign traffic—can be used to estimate the model's false positive risk. The FireRL agent can learn to differentiate between harmless and harmful traffic via the use of Deep Q-learning, which resulted in a reduction in the FPR. This was accomplished by continual environmental input. Every time the agent completes a task, such as accepting or rejecting a packet, it receives a reward signal. This signal is sent to the agent. Every time it erroneously blocks a packet, it is rewarded with a negative reward instead. The number of false positives diminishes as the agent gains experience and realizes it should not interfere with genuine conversations.

A reduced FPR may be achieved by modifying the Q-values of the agent in such a way that it provides incentives for behaviors that result in accurate classifications. The agent learns firewall rules that more correctly represent the actual nature of traffic via an optimization process driven by incentives. This is true even when dynamic dangers are associated with the traffic.

4.6 Enhanced threat detection rate

Our suggested FireRL system achieves a high ETDR by using reinforcement learning to improve the agent's hostile traffic detection skills and by defining the system's functioning using this equation. The agent's Enhanced Threat Detection Rate proves that it can identify old and new dangers by showing that it can adapt to static and dynamic environments.

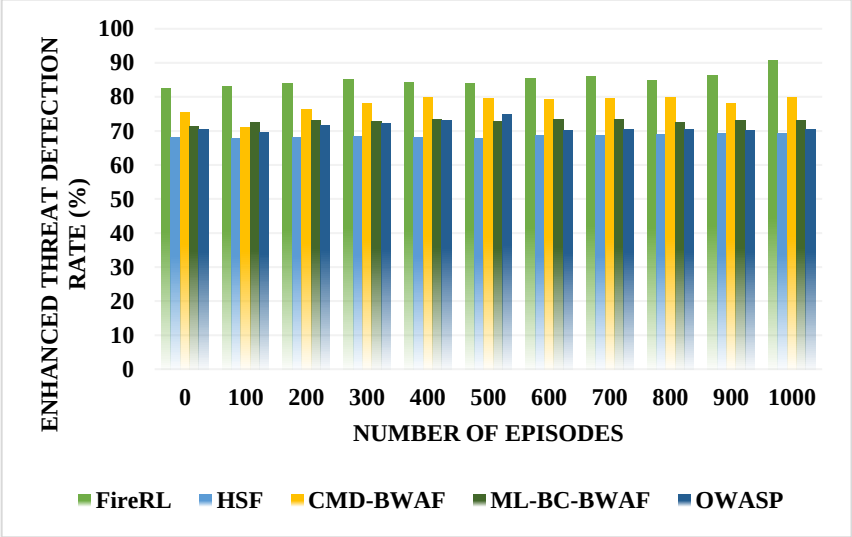


Figure 10: Enhanced threat Detection Rate (%)

$$ETDR_t = \frac{TP_t + \lambda . NT_t}{TP_t + FN_t + \lambda . (NT_t + MN_t)} \quad (13)$$

Figure 10 and equation (13) express the enhanced threat detection rate. The equation investigates the FireRL agent's capacity to identify newly emerging threats and those already present. The improved threat detection rate at time t is denoted by the equation $ETDR_t$ and the equation is used to assess the capability of the FireRL agent to recognize threats. Unlike static, rule-based systems, reinforcement learning allows agents to adapt to attack patterns they have not previously seen and observed. This is in contrast to rule-based systems, which are considered static. In designing this product, a significant emphasis was placed on its practicality. The total number of occurrences of potentially dangerous traffic that the agent was able to detect and prevent from happening is referred to as the True Positives, which may also be abbreviated as TP_t. This indicator indicates that the agent is capable of efficiently counteracting the dangers that have been identified.

Incorrect packets that the agent failed to discover are called negatives, sometimes called defective negatives. This set of negatives is represented by the symbol FN_t. A reduction in the FN makes it feasible to enhance detection, which eventually leads to an increase in the ETDR of the system.

It is the aggregate of all of these threats that is signified by the mark NT_t, which represents the total number of Novel Threats successfully discovered and eradicated. As a result of the generalizability of the system, it is feasible that the agent will identify some kinds of assaults it has never encountered before throughout its training. This is of the utmost importance because the dangers in the real world are always evolving.

It may be deduced from the existence of a warning message that starts with the phrase "Missing Novel Threats" (MN_t) that the agent did not discover any new threats.

Many aspects contribute to the significance of identifying new threats. One of these factors is the novelty weighting factor, which is represented by the symbol λ (in most cases, λ is larger than 1 1). The use of attack signatures that are already known is successfully discouraged by this option. This is accomplished by modifying the incentive system's bias toward accurately identifying new threats.

5 Conclusion

Our paper proposes FireRL, a reinforcement learning-powered adaptive firewall architecture, to solve the inadequacies of static rule-based systems in changing cybersecurity scenarios. FireRL intelligently optimises throughput, latency, false positive rate, and threat detection rate using Deep Q-Learning and a Markov Decision Process firewall rule optimization model. Experimental results with benign and malicious traffic reveal that FireRL outperforms traditional firewall systems' accuracy, adaptability, and resilience to new threats. While still protecting against known and emerging threats, the system may automatically update and fine-tune firewall settings, minimizing the need for security experts. Our next effort will augment FireRL with multi-agent reinforcement learning for remote firewall settings and cloud-native architectures. Integration of unsupervised anomaly detection methods improves zero-day threat detection without labeled data. To examine online learning to help FireRL modify its policies to shift traffic patterns in real time. Finally, testing FireRL in corporate or industrial networks will show how it functions under tremendous pressure.

.

Acknowledgement

Funding

This work was supported by 2024 Hunan Province's social science achievement "Research on the Deep Integration of Artificial Intelligence Assisted Education and Teaching" (No. XSP24YBC591).

Author contributions

Y MH:Writing – original draft and Conceptualization,Y ZH and Y QW:Methodology,Validation;Y MH,Y ZH and Y QW:Writing – review & editing.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and publication of this article.

Data availibility statement

All data generated or analysed during this study are included in this article.

References

[1] Md Shamimul Islam, Mohammed Asraf Uddin, Md Dulal Hossain, Md Shakil Ahmed, and Md Golam Moazzam (2023). Analysis and evaluation of network and application security based on next generation firewall. International Journal of Computing and Digital Systems, 13(1), 193–202. https://doi.org/10.12785/ijcds/130116

[2] Jamal Khudair Madhloom, Zainab Hammoodi Noori, Sif K. Ebis, Oday A. Hassen, and Saad M. Darwish (2023). An information security engineering framework for modeling packet filtering firewall using neutrosophic Petri nets. Computers, 12(10), 202. https://doi.org/10.3390/computers12100202

[3] Jian Ma, Chaoyong Zhu, Yuntao Fu, Haichao Zhang, and Wenjing Xiong (2025). Dynamic routing via reinforcement learning for network traffic optimization. Informatica, 49(3). https://doi.org/10.31449/inf.v49i3.7126

[4] Ahmed Shaheed and Mazen B. Kurdy (2022). Web application firewall using machine learning and feature engineering. Security and Communication Networks, 2022(1), 5280158. https://doi.org/10.1155/2022/5280158

[5] Bikram R. Dawadi, Bibek Adhikari, and Deepak K. Srivastava (2023). Deep learning technique-enabled web application firewall for the detection of web attacks. Sensors, 23(4), 2073. https://www.mdpi.com/1424-8220/23/4/2073

[6] Salih Toprak and Aziz G. Yavuz (2022). Web application firewall based on anomaly detection using deep learning. Acta Infologica, 6(2), 219–244. https://doi.org/10.26650/acin.1039042

[7] Davide Bringhenti, Giorgio Marchetto, Roberto Sisto, Franco Valenza, and Juman Yusupov (2022). Automated firewall configuration in virtual networks. IEEE Transactions on Dependable and Secure Computing, 20(2), 1559–1576. DOI: https://doi.org/10.1109/TDSC.2022.3160293

[8] Mohammad Farooq, Riyaz Khan, and Mohammed H. Khan (2023). Stout implementation of firewall and network segmentation for securing IoT devices. Indian Journal of Science and Technology, 16(33), 2609–2621. DOI: https://doi.org/10.17485/IJST/v16i33.1739

[9] Amro Ameid Alkato and Yara Sakhnini (2025). Advanced real-time anomaly detection and predictive trend modelling in smart systems using deep belief networks architectures. PatternIQ Mining, 2(1), 24 Feb. 2025. https://doi.org/10.70023/sahd/250209

[10] Ekhlas Kadhim Hamza, Haidar H. Abdulameer, Noor H. F. Al-Mashhadani, and Fatima S. Al-Hindawi (2024). Reinforcement learning algorithms for adaptive load balancing in publish/subscribe systems: PPO, UCB, and epsilon-greedy approaches. Informatica, 48(7), 881–893. https://doi.org/10.31449/inf.v48i7.6895 .

[11] Marek Sepczuk (2023). Dynamic web application firewall detection supported by cyber mimic defense approach. Journal of Network and Computer Applications, 213, 103596. https://doi.org/10.1016/j.jnca.2022.103596

[12] S. Rajasoundaran, S. A. Sivakumar, S. Devaraju, M. J. Pasha, and J. Lloret (2024). A deep experimental analysis of energy-proficient firewall policies and security practices for resource limited wireless networks. Security and Privacy, 7(6), e450. https://doi.org/10.1002/spy2.450

[13] Jae-Kwang Lee, Tae Hong, and Geon Lee (2024). AI-based approach to firewall rule refinement on high-performance computing service network. Applied Sciences, 14(11), 4373. https://doi.org/10.3390/app14114373

[14] Edison Leka, Lamani Lamani, Albana Aliti, and Elvana Hoxha (2024). Web application firewall for detecting and mitigation of based DDoS attacks using machine learning and blockchain. TEM Journal, 13(4). https://doi.org/10.18421/TEM134-50

[15] Ahmad Fadlil, Irfan Riadi, and Muhammad A. Mu'Min (2024). Mitigation from SQL Injection attacks on web server using Open Web Application Security Project framework. International Journal of Engineering, 37(4), 635–

645. https://doi.org/10.5829/ije.2024.37.04a.01

[16] Théodore Kouassi, B. H. Kamagate, O. Asseu, and Y. Kermarrec (2024). Security policy model in a hybrid Zachman-TOGAF framework for a telework enterprise architecture in a cloud environment. Open Journal of Safety Science and Technology, 14(3), 96–115. https://doi.org/10.4236/ojsst.2024.143007

[17] Kaggle (n.d.). Internet firewall data set. Retrieved 20 March 2025, from https://www.kaggle.com/datasets/tunguz/internet-firewall-data-set

Federated Learning-Based Network Threat Detection System with Digital Twins for IoT Security

Feixian Sun
School of Modern Information Technology, Henan Polytechnic, Zhengzhou 450046, China
Zhengzhou Key Laboratory of Electronic Intelligent Sensor Application Technology, Zhengzhou 450046, China
E-mail: 18538771228@126.com

Keywords: internet of things, digital twin, blockchain technology, federated learning support, network threat detection system

Received: May 23, 2025

Abstract: With the widespread use of Internet of Things technology, network threats are increasing, posing a serious challenge to the security of Internet of Things systems. To address this challenge, an efficient network threat detection system is designed by combining digital twin technology and federated learning algorithms. The research first uses the decentralized and immutable characteristics of blockchain technology to securely store and verify the data in the Internet of Things network, while combining the digital twin technology to carry out virtual mapping of the Internet of Things entity, real-time monitoring of its status, and timely detection of potential threats. Subsequently, a comprehensive simulation of the Internet of Things system is conducted via the digital twin network to generate data samples. Following this, the federated learning algorithm is employed, enabling multiple participants to collaboratively train the model. This approach enhances the model's detection capabilities while safeguarding the privacy of local data. Additionally, a distributed architecture is adopted to facilitate the efficient processing and analysis of large-scale Internet of Things data. Finally, the proposed system is tested. The test results show that in terms of registration time, when the number of attributes is 10, the registration time of the research system is about 0.57 seconds, the registration time of the micro-step online threat intelligence platform is about 0.59 seconds, and the registration time of the Check Point Infinity platform is about 0.62 seconds. When the number of access policies is 10, the shortest encryption time of the research system is 0.52 seconds, the second is 0.54 seconds of the micro-step online threat intelligence platform, and the longest is 0.58 seconds of the Check Point Infinity platform. In comparison to the benchmark system, the proposed research system demonstrates superior efficiency during the registration and encryption phases. This is primarily attributed to the precise modeling of Internet of Things devices using digital twin technology and the efficient data processing capabilities inherent in the federated learning algorithm. Consequently, the research system offers a swifter and more effective solution for detecting network threats within the Internet of Things environment.

Povzetek: Združuje digitalne dvojčke, verigo blokov ter federativno učenje za decentralizirano zaznavanje groženj v IoT. Poudarjen je zasebnostno ohranjen trening in razlaga modela, je preizkušeno na simulacijah in resničnih scenarijih.

1 Introduction

In recent times, with the swift advancement of Internet of Things (IoT) technology, more and more devices are connected to the network through information sensing devices and protocol agreements, achieving intelligent recognition, positioning, tracking, and monitoring functions [1, 2]. With the increasing severity of network threats, it poses a serious threat to the security and reliability of IoT systems [3]. Therefore, developing an efficient and reliable network threat detection system is extremely important for ensuring the normal operation of the IoT. Federated Machine Learning (FML), as a highly promising technological approach, has received widespread attention [4]. FML, also known as Federated Learning (FL), united learning, or alliance learning, is based on deep learning and can combine data from multiple devices to improve model accuracy while maintaining device data privacy [5]. Ma applied FL to the advancement of a smart tourism service system based on the IoT and machine learning, which not only improved the system's performance but also ensured data privacy [6]. Digital Twin (DT) technology greatly enhances the data processing capabilities of physical entities by constructing virtual images in the digital space, providing powerful technical support for deeper analysis and monitoring of the status of IoT devices [7, 8]. For example, Zhao et al. raised an efficient communication FL method for industrial IoT DT systems, further optimizing the efficiency of data processing and model training [9]. In addition, blockchain technology also serves as a crucial component in ensuring the security of IoT data. By offering tamper-resistant data storage and transmission capabilities, blockchain can effectively safeguard data

against tampering and theft, thereby establishing secure and dependable connections among IoT nodes. Currently, there is a wealth of research focused on the application of blockchain technology in the IoT domain. For example, Sasikumar et al. [10] proposed a blockchain-based trust mechanism for DT industrial IoT, while Zheng et al. [11] further explored the integration of blockchain and DT technology in the context of trusted DT for IoT, aiming to balance data privacy protection and the secure and reliable operation of the system. The detailed work summary table is shown in Table 1.

Table 1: Summary of related work

References	Model/method	Data set	Detection accuracy	Calculate the cost	Safety feature	Shortcoming	Contribution
[6]	IoT + Machine learning + FL	Smart travel service dataset	High	Intermediate	Basic encryption	Lack of dynamic device modeling and high computational latency	A preliminary framework for the combination of iot and FL is proposed, which lays a foundation for the subsequent research
[9]	Industrial IoT DT + FL	Industrial IoT data set	High	Intermediate	No blockchain support	Inadequate privacy protection	The introduction of DT technology improves the adaptability and accuracy of the model
[10]	Blockchain + DT	Industrial IoT data set	Intermediate	Intermediate	Tamper-resistant	No FL is integrated, and the detection accuracy is not quantified	The combination of blockchain technology enhances the security and immutability of data
[11]	Blockchain + DT	IoT data set	Intermediate	Intermediate	Trusted authentication	Poor scalability	A scheme combining blockchain and DT is proposed to enhance the credibility of the system
Proposed system	FL + DT + Blockchain	IoT cyber threat dataset	High	Low	High	No obvious deficiency	FL, DTs, and blockchain technologies are integrated for efficient and secure cyber threat detection

From Table 1, the existing DT research relies on centralized data aggregation and fails to make full use of FL's distributed privacy protection capability. The proposed system realizes local data non-sharing through FL and combines blockchain encryption storage to double guarantee privacy. The existing FL model lacks the real-time virtual mapping of DT and cannot respond to threats dynamically. The proposed system uses DT to provide real-time device status image and FL model to update dynamically to improve the timeliness of threat detection. The blockchain scheme introduces high latency, which is difficult to apply to large-scale IoT, and the proposed system optimizes FL participation node selection, reduces the frequency of blockchain transactions, and balances security and efficiency. The proposed system innovatively integrates FL, DTs, and blockchain technologies to enable efficient and secure cyber threat detection. The system demonstrates excellent performance on the IoT network

threat dataset, achieving high detection accuracy with low computing costs, and boasts a high level of security. These attributes compensate for some of the limitations in previous literature, offering a more comprehensive and advanced solution for the IoT network security domain. It is anticipated to foster technical advancements and practical applications in this field.

2 Methods and materials

2.1 IoT network threat detection based on blockchain technology and DT technology

With the increase in the number of IoT devices, the amount of data and requests that central nodes need to process has significantly increased, seriously affecting the system's response speed [12]. In the traditional centralized model of IoT, data processing and management are highly dependent on the central node, and once the node fails or suffers a cyber attack, the entire system may be paralyzed. This issue underscores the vulnerability of existing IoT systems when faced with large-scale device access, and it also reveals the core challenge of the research: How to construct an efficient and reliable network threat detection system under a decentralized architecture using DT and FL technologies, in order to enhance the security and stability of IoT systems. The distributed ledger of blockchain technology can store data on multiple nodes instead of relying on a single central node [13]. The incorporation of blockchain technology in the IoT significantly mitigates the risk of a single point of failure, thereby enhancing the system's robustness and resilience to faults [14]. The flowchart illustration of the IoT blockchain architecture is in Figure 1.

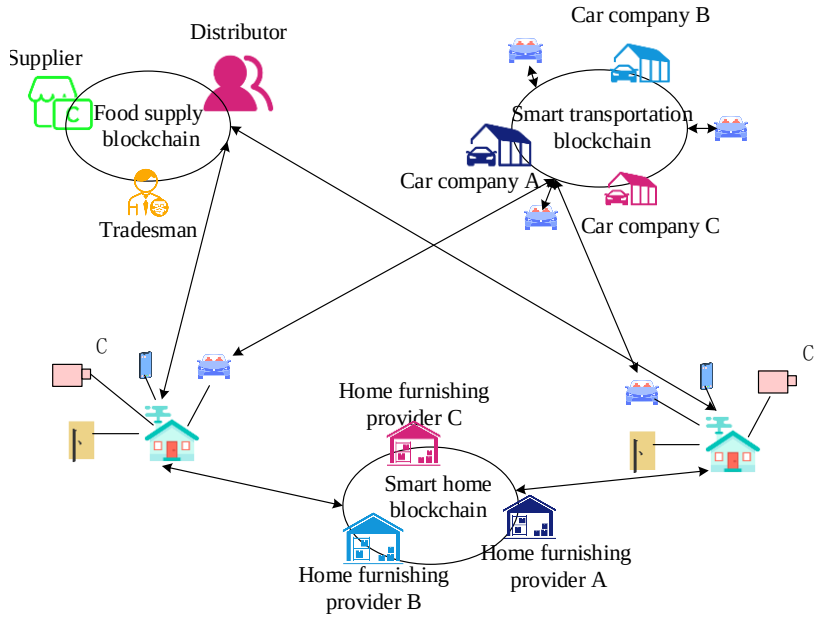


Figure 1: Flowchart illustration of the IoT blockchain architecture

From Figure 1, in the intelligent transportation system, IoT devices such as traffic signal controllers and vehicle sensors are faced with network threats such as distributed denial of service attacks and false data injection. DT technology can build a virtual image of the traffic network, simulate the attack propagation path, and rehearse defense strategies. In a smart home environment, where devices such as home gateways and smart appliances are vulnerable to man-in-the-middle attacks or new types of malware, the blockchain alliance chain can restrict the data access rights of manufacturers and user nodes, ensuring the privacy and security of device data. In the food supply system, the cold chain sensors in the supply chain nodes may be physically tampered with, leading to the falsification of temperature records. Moreover, blockchain technology can store the detection results of each link on the chain for regulators to trace, thus effectively preventing data falsification and leakage. In order to accurately detect threats to the IoT network, a blockchain regulatory framework that supports multi-party participation is studied. The framework combines symmetric encryption algorithms to quickly encrypt data generated by IoT devices. The key generation center will query the local database to see if the regulatory center has completed registration. After registration, the key generation center generates a registration record expression, as shown in equation (1).

$$R \leftarrow (U \| iw_s \| N \| t_{\mathrm{Re}g})^k \tag{1}$$

In equation (1), the customer set is represented by symbol U, the central signature is represented by symbol N, and the regulatory center is represented by symbol iw_s. $t_{\mathrm{Re}g}$ is time. The data information expression of the regulatory center is shown in equation (2).

$$M_1 \leftarrow (m \| C \| J_1 \| t_{\mathrm{R}A} \| id_{RA}) \tag{2}$$

In equation (2), the data to be regulated is represented by symbol m, customer information data is represented by symbol C, the time required for regulation is represented by symbol t_{RA}, and the regulatory code is represented by symbol id_{RA}. The calculation formula for regulatory results information by regulatory agencies is shown in equation (3).

$$J_1 \leftarrow (id_m \| C \| id_{BC} \| Q \| \mu_{RA}) \tag{3}$$

In equation (3), RA stands for supervisory center, the code to be regulated is represented by symbol id_m, and the audit judgment result is represented by symbol μ_{RA}. The business chain code is represented by symbol id_{BC}. The ciphertext b_1 expression corresponding to the symmetric key k in the regulatory center is shown in equation (4).

$$b_1 = A.Enc(PK, k, T) \tag{4}$$

In equation (4), PK represents the common parameters in the attribute based encryption algorithm and T represents the access structure in the attribute based encryption algorithm. The ciphertext b_2 expression is shown in equation (5).

$$b_2 = F.Enc(k, M_1) \tag{5}$$

The encrypted data (b_1, b_2) will be handed over to the regulatory authority for another regulatory judgment, and the judgment formula is shown in equation (6).

$$M_2 \leftarrow (m \| C \| J_2 \| t_S \| id_S) \tag{6}$$

In equation (6), t_S represents the time during which the regulatory authority conducted supervision, and J_2 represents the regulatory outcome information. A smart contract is a self-executing protocol with terms that are encoded in the blockchain as code, which automatically triggers an action when a predefined condition is satisfied. The smart contract will compare the supervision results before and after the two times, and the supervision results of the regulator and the regulator node are consistent, and the supervision information expression is shown in equation (7).

$$L \leftarrow (M \| t_{pro}) \tag{7}$$

In equation (7), t_{pro} represents the data processing time. A DT Network (DTN) is a virtual digital model of a physical network facility created by the DT technology. It can create a virtual image of a physical network facility to simulate the propagation path and impact scope of a network attack. This enables security teams to simulate potential cyber threats in a controlled environment, evaluate the effects of different defense strategies, and select the optimal response [15, 16]. In the IoT blockchain architecture, DTNs can leverage the distributed ledger, smart contracts, and consensus mechanisms of blockchain to enhance data security, privacy protection, and trust building. The schematic diagram of the DTN is in Figure 2.

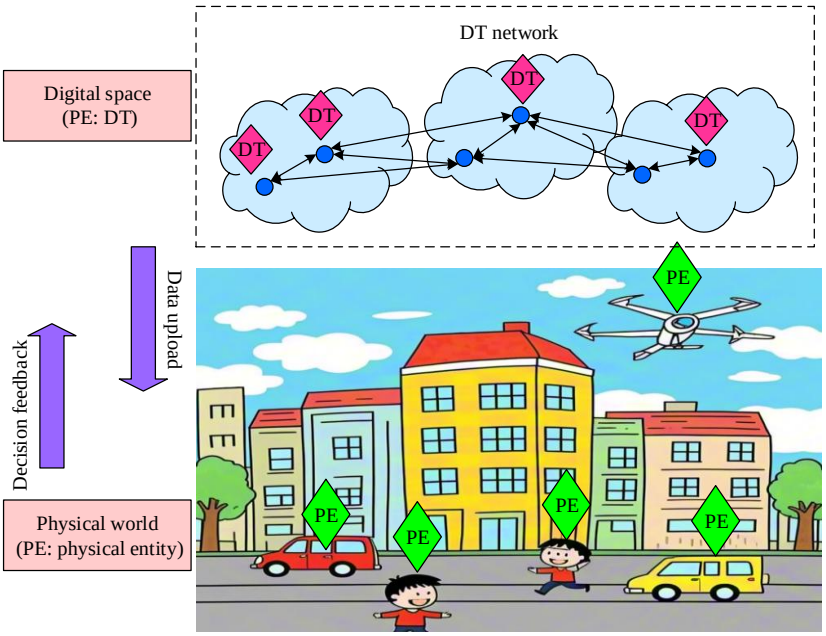


Figure 2: Schematic diagram of DTN

As shown in Figure 2, the DTN includes physical space, data space, and inter entity communication. The DTN collects real-time data of physical entities through IoT technology and transmits it to the digital space for processing and analysis. Data space is the core of DTNs, used to store and process multi-source data from physical space. The bidirectional data flow between physical entities and DTs is used for real-time synchronization of status and feedback optimization. The information sharing and collaboration between DTs break the limitations of

physical space, enabling federated simulation and optimization of complex tasks. Bloom Filter (BF) is an efficient space saving data structure used to determine whether an element belongs to a set. In the cache penetration problem, the BF can store the key existing in the database in advance. When the query request arrives, the BF determines whether the key exists first, so as to avoid invalid database queries. Therefore, the study adopts BF to quickly determine whether data already exists, avoiding duplicate storage and processing of the same data, thereby saving storage space and improving system efficiency. The flowchart illustration of BF is in Figure 3.

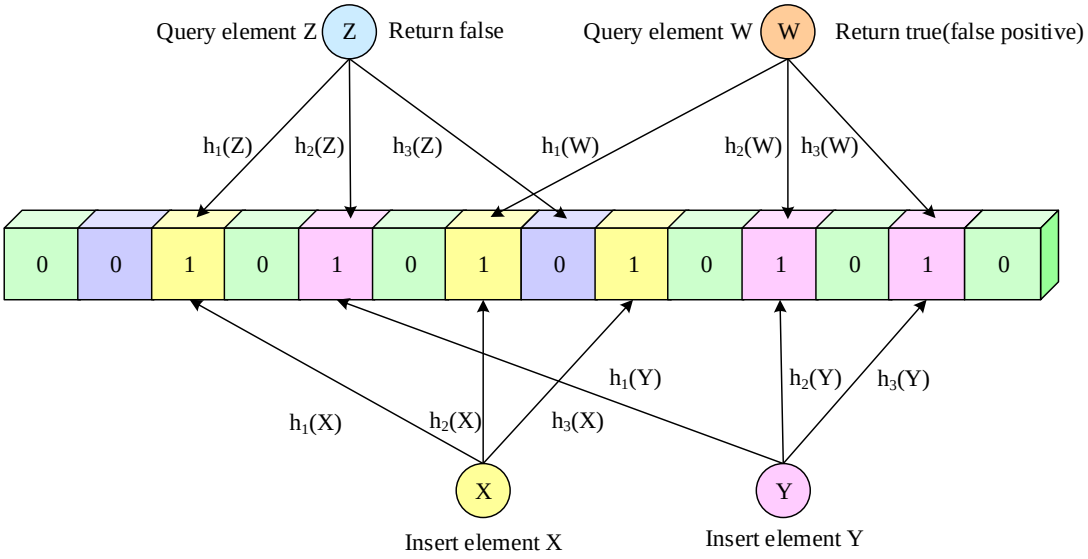


Figure 3: Flowchart illustration of the BF

In Figure 3, when inserting an element x into the BF, each hash function $h_i(x)$ is used to hash element x, and the corresponding bit for each position calculated by the hash function is set to 1. When querying whether an element Z exists in BF, the same hash function $h_i(Z)$ is used to hash element Z, and it is necessary to check whether the bits corresponding to each position calculated by the hash function are all 1. The false positive of BF refers to BF incorrectly determining that an element exists in the set, but in reality, the element is not in the set. False positives are an important characteristic of BF, and their probability of occurrence is closely related to the parameters of BF. Assuming the false positive rate is f, the calculation formula is shown in equation (8).

$$f = (1-1(1-\frac{1}{m})^{n\times k})^k \approx (1-e^{\frac{n\times k}{m}})^k \quad (8)$$

In equation (8), n is the number of elements contained in the set, k is the number of hash functions in the filter, and m is the number of bits. In the IoT environment, BF can be combined with distributed intrusion detection systems to distribute detection functions on edge nodes of the IoT. Each node can independently run BF to quickly detect abnormal behavior in the local network.

2.2 Design of IoT network threat detection system integrating DTN and FL support

The previous section studied the use of blockchain technology and DT technology to optimize the IoT network threat detection performance. However, in the complex IoT ecosystem, device heterogeneity results in significant variations in data formats, processing capabilities, and communication protocols among different devices. This heterogeneity seriously hinders the unified and efficient modeling and management of IoT devices using DT technology. FL not only protects data privacy, but also allows different devices to participate in model training according to their own computing power and data characteristics. Distributed learning methods enable IoT devices to jointly optimize detection models without sharing original data, thus overcoming the obstacles caused by device heterogeneity. Therefore, the introduction of FL provides a more flexible and efficient solution for IoT network threat detection. Unlike traditional centralized learning methods, FL typically requires consolidating all data into a central location for model training. Among them, the expression of the local model parameter α_a is in equation (9).

$$\alpha_a = \arg\min_{\alpha} \beta(\alpha, D_a) \quad (9)$$

In equation (9), D_a represents the local dataset. The expression of the global model parameter α_{new} is shown in equation (10).

$$\alpha_{new} = \frac{\sum_{a=1}^{K} n_a \cdot \alpha_a}{\sum_{a=1}^{K} n_a} \tag{10}$$

In equation (10), n indicates the number of clients participating in FL. FL can be divided into horizontal FL, vertical FL, and federated transfer learning. Among them, horizontal FL is suitable for scenarios where the participating datasets have the same feature space but different sample spaces. Horizontal FL treats data as horizontally partitioned in a tabular view, where each participant has different samples but the feature dimensions of the samples are the same. The schematic diagram of the training process for horizontal FL is shown in Figure 4.

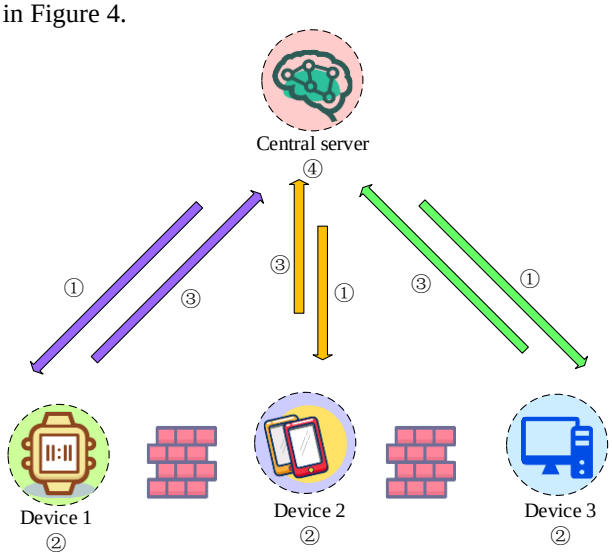


Figure 4: Horizontal federation learning training process diagram

In Figure 4, the central server first initializes a global model parameter, which is sent to all participating parties. After receiving the global model parameters, each participant uses their local dataset to train the model locally. The training process may involve multiple iterations until a predetermined local training standard is reached. After completing local training, each participant will receive a locally trained model parameter. After receiving the local model parameters uploaded by all participants, the central server aggregates these parameters using the federated averaging algorithm. After completing the model aggregation, the central server sends the updated global model parameters back to all participants. After receiving the new global model parameters, each participant updates the parameters based on them. The entire process is repeated until a certain global convergence standard is reached or the preset maximum number of iterations is reached. Support Vector Machines (SVMs) can effectively solve regression and classification problems in high-dimensional feature spaces. The linear SVM model expression is shown in equation (11).

$$\min_{w,b} \frac{1}{2}\|\omega\|^2 \; s.t. y_i(\omega^T x_i + b) \geq 1 \tag{11}$$

In equation (11), ω represents the normal vector of the hyperplane and b represents the intercept. The formula for solving hyperplanes is in equation (12).

$$g(x) = \omega^T x + b \tag{12}$$

The Karush-Kuhn-Tucker condition is a set of necessary conditions in optimization problems used to solve nonlinear programming problems with constraints. The study uses the Karush-Kuhn-Tucker condition b, and the result of the Karush-Kuhn-Tucker conditional function $f(x)$ represents the expression of the classification prediction result, as shown in equation (13).

$$f(x) = sign(\sum_{i=1}^{g} \kappa_i y_i x_i^T x + b) \tag{13}$$

In equation (13), sign is the sign function and x is the input quantity vector. This research adds SVMs on the basis of horizontal FL to construct a horizontal FL support system framework, as shown in Figure 5.

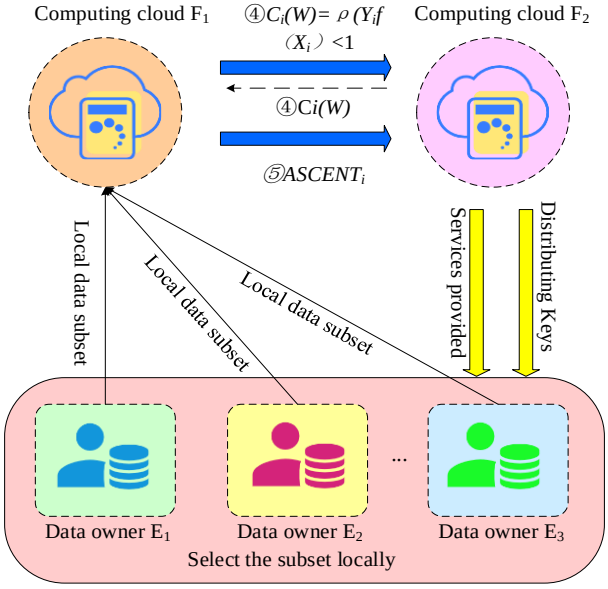


Figure 5: Horizontal FL support system framework

In Figure 5, during the preprocessing stage, the data owner is responsible for preprocessing the local data, collaborating with the computing cloud to execute the subset selection algorithm, encrypting the local training data using the encryption key supplied by the service cloud, and generating a ciphertext dataset. The computing cloud collaborates with data owners and service clouds to complete standardized operations and subset selection algorithms. Encrypted datasets uploaded by data owners are received and stored in a secure environment. The service cloud distributes encryption keys to data owners and collaborates with the computing cloud to complete standardized operations. During the model training and interactive computation phase, the computing cloud conducts model training on a ciphertext dataset, and all computation processes are completed within the ciphertext to ensure data privacy. Meanwhile, it interacts with the service cloud for interactive computation to update and aggregate model parameters. The service cloud interacts with the computing cloud for computation,

providing decryption keys to support ciphertext calculation. The data owner remains offline and does not directly participate in the calculation process. When needed, the data is obtained from the service cloud to obtain the final trained model for local inference or other applications. In the model distribution and service phase, the service cloud obtains the final global model after the model training is completed. The data owner obtains the final model from the service cloud and uses it locally for inference or other applications. In other applications, data owners can utilize the model to analyze the energy consumption of IoT devices. By evaluating the energy utilization efficiency through the trends in energy consumption data output by the model, they can then formulate energy-saving optimization strategies. These strategies may include adjusting equipment operating parameters and scheduling equipment operating times reasonably, thereby achieving the goal of energy conservation and emission reduction. The design diagram of an IoT network threat detection system that integrates DTN and FL support is shown in Figure 6.

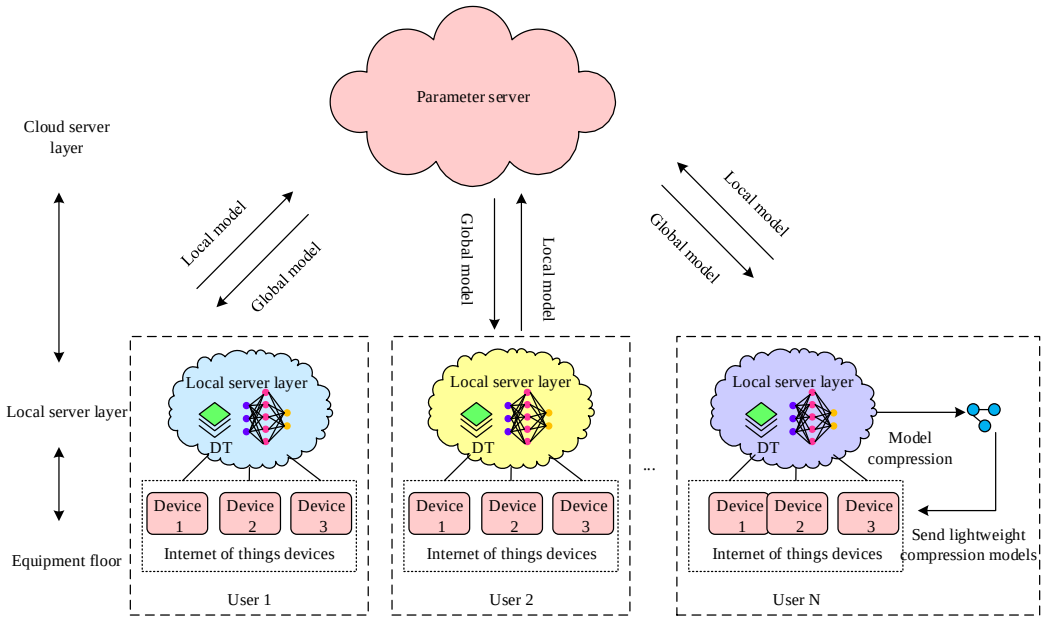


Figure 6: Design of IoT network threat detection system supported by DTN and FL

As shown in Figure 6, intelligent IoT devices collect real-time data and transmit it to local servers through encrypted communication protocols. Then, through the DT system, real-time monitoring and anomaly detection of intelligent IoT devices can be achieved. In each round of FL, the local server utilizes local data and computing resources to train the local model. The local server sends the updated model parameters to the cloud server for aggregation, generating a new global model. The cloud server sends the new global model parameters back to the local server for the next round of training. The main role of the cloud server in the system is the model aggregation center, which receives updated model parameters uploaded from various local servers, performs aggregate calculation based on these parameters, fuses multi-party data features, and generates new global model parameters. The updated global model parameters can comprehensively capture the overall characteristics of the local data involved in FL, enhance the model's generalization ability, and subsequently provide a more representative model foundation for the next round of training on the local server. This, in turn, improves the threat detection performance of the entire system. The local server uses model compression technology to compress personalized models into lightweight models, and deploys the compressed models to intelligent IoT devices to achieve offline threat detection. The DT system and local model jointly monitor the status of intelligent IoT devices and trigger alerts when abnormal behavior is detected. Assuming there are a total of N customers, set as F, each user's IoT device is represented by a symbol D_j, and the corresponding DT generated by the local server is represented by symbol DT_j. The expression for DT_j at time t is shown in equation (14).

$$DT_j(t) = \left\{ Model_j(t), DATA_j(t), Comp_j(t) \right\} \tag{14}$$

In equation (14), $Model_j(t)$ represents device, $Comp_j(t)$ represents available computing resources, and $DATA_j(t)$ represents historical data. To achieve load balancing, training tasks can be decomposed into multiple subtasks and then assigned to different devices. The expression for the computing task $Task_v$ assigned to device v is shown in equation (15).

$$Task_v = \frac{Comp_v(t)}{\sum_{v=1}^{M} Comp_v(t)} Task \tag{15}$$

In equation (15), $Comp_v(t)$ represents the available computing resources of the device v at time t. To

improve the interpretability of threat detection model, XAI module is embedded in FL multi-level model. Each client uses SHapley Additive exPlanations (SHAP) to calculate input features (contribution to local threat classification results), generating feature importance heat maps. The server identifies cross-domain consistency threat features by aggregating the SHAP value distribution of each client through federation. For complex admixture samples, Local Interpretable Model-agnostic Explanations (LIME) was used to reconstruct interpretable alternative models in DT virtual environments. Based on Ubuntu16.04 operating system and using Python and Solidity programming languages, the research used Eclipse Hono to connect IoT devices and collect relevant information, and gathers data to a single AMQP 1.0 endpoint. The Python library that interacts with the Ethereum blockchain through Web3.py enables the deployment and invocation of smart contracts. Solana.py was used to interact with the Python library of Solana blockchain, Paho-MQTT was used to achieve MQTT protocol communication between IoT devices and servers, and InfluxDB was used to store and query time series data. Finally, a network threat detection system with DT and FL support for IoT was constructed.

3 Results

3.1 Algorithm performance testing

To confirm the accuracy and scientificity of the research outcomes, a simulation experimental environment was set up, as shown in Table 2.

Table 2: Simulation experiment environment setting table

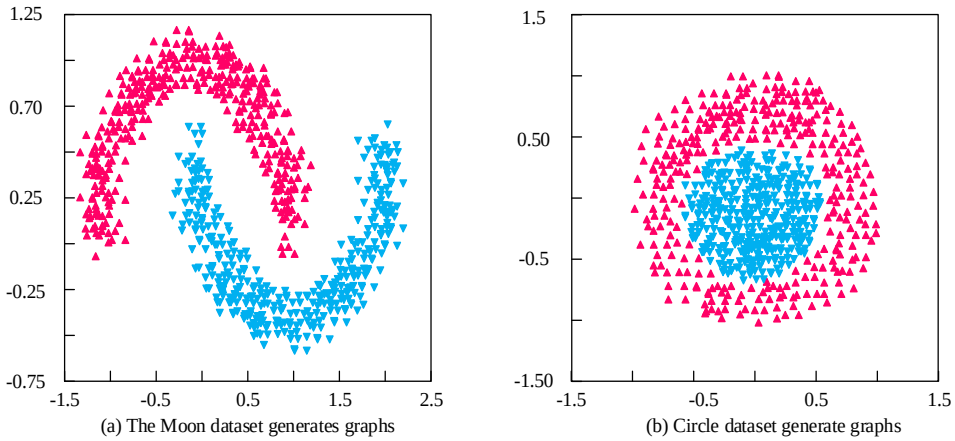
Name	Disposition
Processor	Intel(R) Core(TM) i7 6700H @ 3.40 GHz
Operating system	Windows 10 64
Graphics card	NVIDIA Geforce RTX 3060
Internal memory	16 GB
Hard disk	1 TB

The simulation experiment used two datasets, Moon and Circle. The Moon dataset was usually a two-dimensional synthetic dataset used to demonstrate and test classification algorithms. The data points in this dataset were sampled from two noisy crescent shape data distributions, each containing two features. The Circle dataset referred to a dataset that generates circular distributions, consisting of two sets of points with circular distributions, used to test the algorithm's ability to process complex shaped data. Ring dataset is a multi-modal, multi-version annotated dataset mainly used for target detection tasks, including ring images in various scenes, which is suitable for the research in the field of computer vision and target detection. The Pumpkin Seeds dataset is an agricultural classification dataset that contains 2,500 grayscale and binary images of pumpkin seeds for the classification task. The hyper-parameter settings of the four types of data sets are shown in Table 3.

Table 3: Hyper-parameter setting table

Data set	Lot size	Penalty factor	Object mapping dimension	Learning rate	Gamma value
Moon	16	1.0	100	0.001	2.0
Circle	16	1.0	100	0.001	2.0
Ring	16	1.0	100	0.002	0.1
Pumpkin Seeds	16	1.0	100	0.01	0.1

The generated graphs of the research algorithm on two datasets are shown in Figure 7.

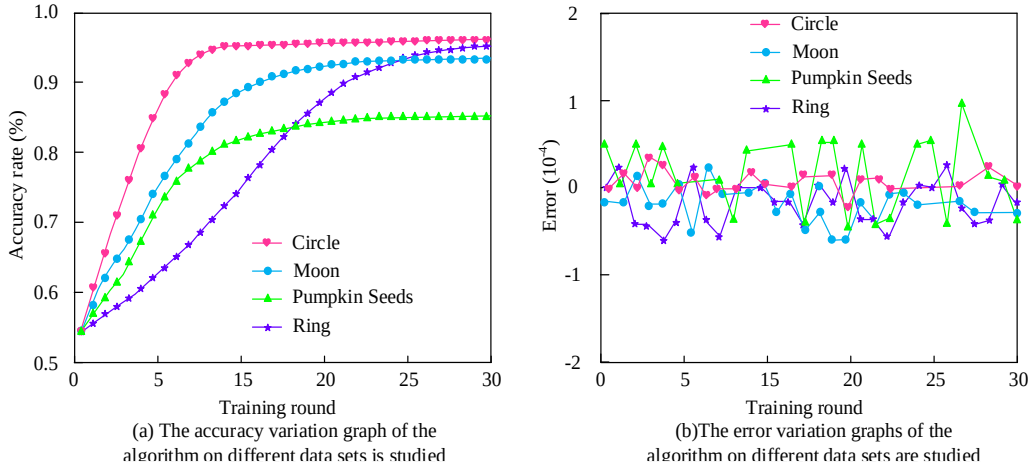


(a) The Moon dataset generates graphs (b) Circle dataset generate graphs

Figure 7: Generated graphs of the research algorithm on two datasets

Figure 7 (a) shows the generated image of the Moon dataset. From Figure 7 (a), on the real dataset Moon, the research algorithm could clearly distinguish the data in the Moon dataset, and ultimately present it as two crescent shapes. Figure 7 (b) shows the generated image of the Circle dataset. From Figure 7 (b), on the real dataset Circle, the research algorithm accurately classified the data points into two circular distributions, presenting clear

circular contours. In the simulation experiment, the data set was divided into training set and test set. Among them, the training set accounted for 80%, which was used for the training of the model. The test set accounted for 20% and was used to evaluate the performance of the model after training. The maximum number of iterations for algorithm training was set to 30. In each iteration, participants updated the locally trained algorithm parameters and interacted with other participants via secure communication protocols. The variation trend of accuracy and error of the algorithm on different data sets is shown in Figure 8.



(a) The accuracy variation graph of the algorithm on different data sets is studied

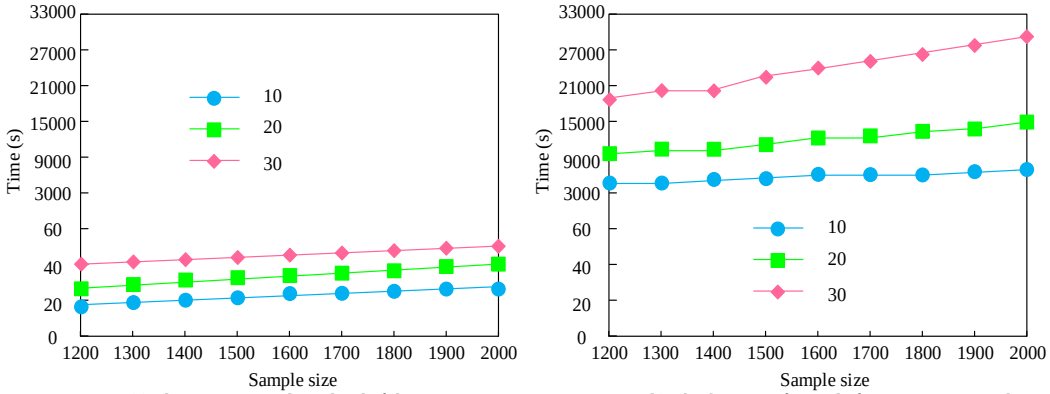
(b)The error variation graphs of the algorithm on different data sets are studied

Figure 8: The trend of accuracy and error of the algorithm on different datasets

Figure 8 (a) indicates the accuracy variation of the research algorithm on various datasets. From Figure 8 (a), the research algorithm had the highest accuracy on Circle dataset, reaching 97.3%, and the lowest accuracy on dataset Pumpkin Seeds, reaching 83.6%. This indicated that even in the face of more complex datasets, the research algorithm could still maintain good data monitoring performance. Figure 8 (b) shows the error variation of the research algorithm on different datasets. From Figure 8 (b), the error of the research algorithm fluctuated up and down within the range of -1×10^{-4} to 1×10^{-4} on the four types of datasets, with a relatively small fluctuation range.

3.2 System performance analysis

To verify the performance advantages of the proposed system in network threat detection, comparative experiments were carried out in the Distributed Denial of Service attack (DDoS) scenario based on the real object networking dataset ToN-IoT. Check Point Infinity platform and Weibu online threat intelligence platform were selected as the baseline system to conduct quantitative analysis from two dimensions of computing cost and response time. The experimental results are shown in Figure 9.



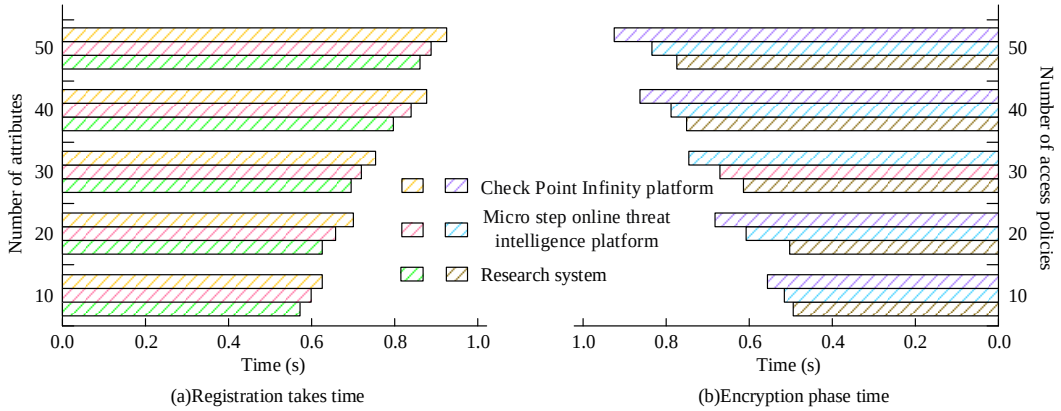
(a) The computational overhead of the system in different dimensions of the data set is studied

(b) Check Point Infinity platform's computational overhead in different dimensions of the dataset

Figure 9: The total computational cost of the system proposed by this study and the Check Point Infinity platform

Figure 9 (a) shows the computational cost of the research system in different dimensions of the dataset. From Figure 9 (a), as the sample size increased, the computational cost time of the research system increased linearly. When the sample size reached 2000, the computational cost time of the research system was 23 seconds in 10 dimensions, 38 seconds in 20 dimensions, and 42 seconds in 30 dimensions. This was because the real-time attack traffic simulation and node isolation capability of the DT module significantly reduced redundant computation. The dynamic client scheduling mechanism of FL module avoided the resource bottleneck

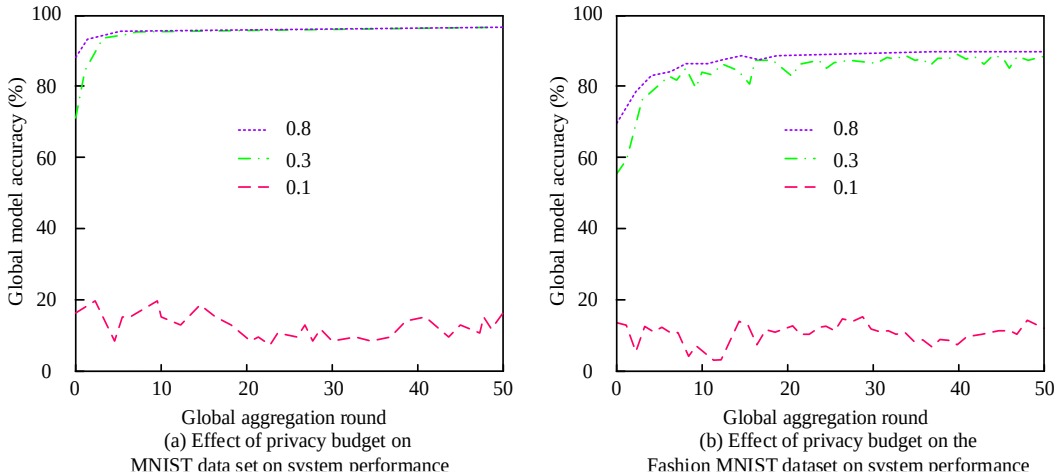
caused by single point dependence. Figure 9 (b) shows the computational cost of the Check Point Infinity platform in different dimensions of the dataset. As shown in Figure 9 (b), when the sample size reached 2000, the computational cost time of the Check Point Infinity platform in the three dimensions was 3280 s, 12580 s, and 30060 s, respectively. This gap stemmed from the baseline platform's lack of virtualized attack rehearsal and distributed collaborative learning capabilities, resulting in inefficient processing of high-dimensional attack data. In the system running architecture, the registration process allowed the system to verify the identity of users or devices and assign corresponding permissions to them. Authorized users or devices could access system resources. This built a solid defense line at the access control level, greatly enhancing the security and stability of the system, and effectively resisting potential illegal access and malicious attacks. The study also introduced a micro step online threat intelligence platform for comparative experiments. The experiment outcomes are in Figure 10.



(a)Registration takes time (b)Encryption phase time

Figure 10: Comparison of the efficiency of different IoT network threat detection systems

Figure 10 (a) shows the time spent on registration. According to Figure 10 (a), when the number of attributes was 10, the registration time of the research system was about 0.57 seconds. When the number of attributes increased to 50, the time taken to research system registration increased by only about 0.31 seconds. Figure 10 (b) shows the time diagram of the encryption phase. From Figure 10 (b), the increase of policies would lead to the increase of system encryption time. When the number of access policies was 10, the shortest encryption time of the research system was 0.52 seconds. To analyze the impact of privacy budget on the performance of the network threat detection system model supported by DT and FL for the IoT, two datasets of MNIST and Fashion MNIST were respectively used for analysis. The experimental results are shown in Figure 11.



(a) Effect of privacy budget on MNIST data set on system performance

(b) Effect of privacy budget on the Fashion MNIST dataset on system performance

Figure 11. Impact of privacy budget on system performance on different data sets

Figure 11 (a) shows the impact of privacy budget on system performance on the MNIST dataset. From Figure 11 (a), as the privacy budget rose, the accuracy of the research system also improved. When the privacy data was 0.8, the accuracy of the research system was 92%, indicating that the system could still maintain excellent detection performance at higher levels of privacy protection. Figure 11 (b) shows the impact of privacy budget on system performance on the Fashion MNIST dataset. From Figure 11 (b), although the accuracy of the research system decreased under different privacy budgets in this dataset, overall, the accuracy of the research system

was around 88%, indicating that even with an increase in privacy protection requirements, the network threat detection system still exhibited good detection ability and stability, verifying its practicality and effectiveness in the IoT environment. To comprehensively verify the effectiveness of the proposed method, ablation experiments were conducted on a large data set CIFAR-10, and the experimental results were shown in Table 4.

Table 4 Ablation experiment table

Model setup	Accuracy rate (%)	Error rate (%)	Computational overhead (seconds)	Privacy budget
Basic model	65.7	30.0	20	/
DT module	78.5	25.6	22	/
FL module	72.2	27.1	21	1.2
Research system	82.3	20.2	24	0.8

From Table 4, after the introduction of DT module, the accuracy rate increased from 65.7% to 78.5%, indicating that DT technology could significantly improve the classification performance of the model. After the introduction of FL module, the accuracy rate increased from 65.7% to 72.2%, indicating that FL technology could improve the classification performance of the model. The accuracy of introducing DT and FL module was 82.3%, which was significantly higher than other settings, indicating that the combination of DT and FL could further improve the classification performance of the model.

4 Discussion and conclusion

4.1 Discussion

Given the swift advancement of the IoT, network threat detection becomes a key task in ensuring the security of the IoT. Traditional network threat detection systems often face problems such as high computational overhead, insufficient privacy protection, and limited ability to process complex data. To address these issues, the research proposes an IoT network threat detection system that integrates DTN and FL support.

The research first tested the IoT network threat detection algorithm that integrated DTN and FL support. The results showed that the research algorithm could clearly distinguish sub-nodes and present clear shape contours on the Moon and Circle datasets. On the Circle dataset, the accuracy of the system was as high as 97.3%, and on the more complex Pumpkin Seeds dataset, the accuracy reached 83.6%. This indicated that the research algorithm could maintain good classification performance when facing data of different complexities, which is consistent with the results obtained by Choi W et al. in their DT research in the power generation industry [17]. Then, the performance of the IoT network threat detection system that integrated DTN and FL support was evaluated. The experiment outcome indicated that under the same testing environment and dataset dimensions, the computational cost time of the research system was much lower than that of the Check Point Infinity platform. This was because the research system adopted DT technology, which reduced redundant computing and significantly lowers computational overhead [18]. The study introduced a micro step online threat intelligence platform for comparative experiments, and the experimental outcomes indicated that the study could cope with the dynamic changes in the number of attributes in the IoT environment. This was because the research system adopted a lightweight registration mechanism, which significantly reduced registration time by optimizing the registration process and reducing unnecessary calculation steps. This is consistent with the results of Landrum et al. in a study of a lightweight chemical registration and data storage system [19]. Finally, experiments were conducted based on two datasets, MNIST and Fashion MNIST, respectively. The experimental results showed that under the condition of privacy protection, the detection performance of the research system was somewhat reduced, but it could still maintain excellent detection performance. It is consistent with the research of Yuan et al. on knowledge sharing of vehicle Internet privacy protection based on low-cost federal generalized learning [20]. This was mainly due to the synergy of horizontal FL architecture and DT technology, which effectively guaranteed data privacy by allowing data to be directly processed and model trained on local devices, while significantly reducing data transfer costs. DT technology further improved the detection accuracy and response speed of the system by modeling physical devices and synchronizing real-time data. This combination not only ensured high performance of the model, but also provided a more efficient and secure solution for cyber threat detection in IoT environments.

On the large data set CIFAR-10, the computational overhead of the research system was 24 seconds, which was slightly increased compared with the basic model, but still within the acceptable range, indicating that the system had good scalability when processing large-scale data sets. Although the introduction of DTs and FL modules increased the computational overhead, this increase was linear and did not lead to an exponential increase in computational time, indicating that the system could adapt to the increase in data volume. The research system showed the highest accuracy and the lowest error rate on large data sets, indicating that the combination of DT and FL could effectively improve the classification performance of the model. Even in the case of a large amount of data, the performance improvement indicated that the system could maintain a high detection accuracy and had good scalability when processing large-scale data sets. The privacy budget of the research system was 0.8, indicating that a good balance was achieved between privacy protection and performance. Even on large data sets, the system could provide effective privacy protection

while maintaining high detection performance. The system had good scalability in privacy protection and could adapt to scenarios with different privacy requirements.

In summary, the research system adopted an FL architecture, DT technology, privacy budget mechanism, and efficient encryption technology. These technologies and strategies protected privacy while minimizing the impact on detection performance, ensuring that the system had efficient, secure, and reliable threat detection capabilities in the IoT environment.

4.2 Conclusion

In response to the challenges posed by IoT network threats, an innovative IoT network threat detection system that integrated DTNs and FL support was proposed. Performance testing was conducted on the research algorithm, and in regard to classification performance, the accuracy of the research algorithm on the Circle dataset was as high as 97.3%. Even on the more challenging Pumpkin Seeds dataset, the accuracy reached 83.6%, indicating that the research algorithm could still maintain good monitoring performance when processing complex datasets. In addition, the error fluctuation range of the research algorithm on the four types of datasets was extremely small, ranging from -1×10^{-4} to 1×10^{-4}, further demonstrating the stability and reliability of its performance. In terms of privacy protection, as the privacy budget increased, the accuracy of the system substantially improved. When the privacy budget was 0.8, the system accuracy reached 92%, indicating that at a higher level of privacy protection, the system could not only effectively protect data privacy but also maintain excellent detection performance. In summary, the system proposed by the research showed excellent performance in classification performance, computational efficiency, and privacy protection, and could effectively respond to complex and changing network threats in the IoT environment, which had important practical application value. In the IoT environment, real-time data processing is very important. In the future, edge computing and real-time data analysis technology can be combined to further improve the real-time performance and response speed of the system.

Funding

This study was supported by the Science and Technology Planning Project of Henan Province, China (Grant No. 242102320167), and the research project “Design and Implementation of Real Time State Monitoring System for Distribution BoxBased on AIoT” of Henan Polytechnic in 2024 (2024ZK29).

References

[1] Sonthitham, P., Sonthitham, A., & Khuantham, C. (2021, May). Development of internet of things technology for control system mulberry smart farm: A case study at Surin Province. In 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 989-992. DOI:10.1109/ECTI-CON51831.2021.9454873.

[2] Sun J, Xu Q H, Yu F C, Zhu J, Zhang RX. (2020). Fault detection of sewage treatment hydraulic system based on dynamic GRNN model with internet of things technology. Chinese Hydraulics & Pneumatics, (11), 120-126. DOI:10.11832/j.issn.1000-4858.2020.11.019.

[3] Li, J., Zhi, J., Hu, W., Wang, L., & Yang, A. (2020). Research on the improvement of vision target tracking algorithm for Internet of things technology and Simple extended application in pellet ore phase. Future Generation Computer Systems, 110, 233-242. DOI:10.1016/j.future.2020.04.014.

[4] Sattar, K. A., & Baroudi, U. (2024). Performance evaluation of cluster-based federated machine learning. Neural Computing and Applications, 36(14), 7657-7668. DOI:10.1007/s00521-024-09487-3.

[5] Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., & Zhang, W. (2023). A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 14(2), 513-535. DOI:10.1007/s13042-022-01647-y.

[6] Ma, H. (2024). Development of a smart tourism service system based on the Internet of Things and machine learning. The Journal of Supercomputing, 80(5), 6725-6745. DOI:10.1007/s11227-023-05719-w.

[7] Zhang, Z. (2024). SD-WSN Network Security Detection Methods for Online Network Education. Informatica, 48(21), 51-66. DOI:10.31449/inf.v48i21.6257.

[8] Alcaraz, C., Meskini, I. H., & Lopez, J. (2025). Digital twin communities: an approach for secure DT data sharing. International Journal of Information Security, 24(1), 1-19. DOI:10.1007/s10207-024-00912-1.

[9] Zhao, Y., Li, L., Liu, Y., Fan, Y., & Lin, K. Y. (2022). Communication-efficient federated learning for digital twin systems of industrial Internet of Things. IFAC-PapersOnLine, 55(2), 433-438. DOI:10.1016/j.ifacol.2022.04.232.

[10] Sasikumar, A., Vairavasundaram, S., Kotecha, K., Indragandhi, V., Ravi, L., Selvachandran, G., & Abraham, A. (2023). Blockchain-based trust mechanism for digital twin empowered Industrial Internet of Things. Future Generation Computer Systems, 141, 16-27. DOI:10.1016/j.future.2022.11.002.

[11] Zheng, Q., Wang, J., Shen, Y., Ding, P., & Cheriet, M. (2022). Blockchain based trustworthy digital twin in the Internet of Things. In 2022 International Conference on Information Processing and Network Provisioning, 152-155. DOI:10.1109/ICIPNP57450.2022.00040.

[12] Aryavalli, S. N. G., & Kumar, G. H. (2024). Futuristic vigilance: Empowering chipko movement with cyber-savvy IoT to safeguard forests. Archives of

Advanced Engineering Science, 2(4), 215-223. DOI:10.47852/bonviewAAES32021480.

[13] Zhao, W., Zhang, S., Xue, L., Chang, T., & Wang, L. (2024). Research on model of micro-grid green power transaction based on blockchain technology and double auction mechanism. Journal of Electrical Engineering & Technology, 19(1), 133-145. DOI:10.1007/s42835-023-01541-9

[14] Alturki, N., Alharthi, R., Umer, M., Saidani, O., Alshardan, A., Alhebshi, R. M., et al. (2024). Efficient and Secure IoT Based Smart Home Automation UsingMulti-Model Learning and Blockchain Technology. CMES-Computer Modeling in Engineering & Sciences, 139(3). 3387-3415. DOI:10.32604/cmes.2023.044700.

[15] Ariansyah, D., Isnan, M., Rahutomo, R., & Pardamean, B. (2023). Digital Twin (DT) smart city for air quality management. Procedia Computer Science, 227, 524-533. DOI: 10.1016/j.procs.2023.10.554.

[16] Hakimi, O., Liu, H., & Abudayyeh, O. (2024). Digital twin-enabled smart facility management: A bibliometric review. Frontiers of Engineering Management, 11(1), 32-49. DOI:10.1007/s42524-023-0254-4.

[17] Choi, W., Hudachek, K., Koskey, S., Perullo, C., & Noble, D. (2024). Digital twin in the power generation industry. JMST Advances, 6(1), 103-119. DOI:10.1007/s42791-024-00065-1.

[18] Tellechea-Luzardo, J., Winterhalter, C., Widera, P., Kozyra, J., de Lorenzo, V., & Krasnogor, N. (2020). Linking engineered cells to their digital twins: a version control system for strain engineering. ACS Synthetic Biology, 9(3), 536-545. DOI:10.1101/786111.

[19] Harahsheh, K. M., & Chen, C. H. (2023). A survey of using machine learning in IoT security and the challenges faced by researchers. Informatica, 47(6), 1-54. DOI:10.31449/inf.v47i6.4635.

[20] Yuan, X., Chen, J., Zhang, N., Ye, Q. J., Li, C., Zhu, C., & Shen, X. S. (2024). Low-cost federated broad learning for privacy-preserved knowledge sharing in the RIS-aided internet of vehicles. Engineering, 33, 178-189. DOI:10.1016/j.eng.2023.04.015.