MCO-IDM: A Network Intrusion Detection Model Based on CMO-BOA and Pareto Frontier Search

Laibing Wang

Faculty of Information Engineering, Chuzhou Polytechnic, Chuzhou Anhui 239000, China
E-mail: wanglaibing2026@126.com

Received 04 February 2026; Accepted 24 March 2026

Abstract

With the increasingly complex network environment, intrusion detection systems are faced with severe challenges such as high-dimensional feature redundancy, category imbalance and low detection accuracy. Aiming at these problems, this paper proposes a multi-objective and multi-strategy collaborative optimization intrusion detection model (MCO-IDM). The model innovatively integrates multi-objective optimization techniques to simultaneously optimize conflicting objectives such as minimizing the number of features, maximizing detection accuracy and minimizing false alarm rates, and integrates collaborative search strategies such as dynamic adaptive mechanisms and swarm intelligence optimization algorithms (such as CMO-BOA) to achieve efficient trade-offs through Pareto frontier search and weight adjustment. The test results show that on globally public datasets such as KDD CUP99, NSL-KDD and CIC-IDS2017, MCO-IDM achieves the highest accuracy rate of 97.8%, the false alarm rate is reduced to 4.3%, and the training time is controlled within 185.3 seconds. At the same time, it

Journal of Cyber Security and Mobility, Vol. 15_3, 549–576.
doi: 10.13052/jcsm2245-1439.1532

maintains high robustness and scalability under different data scales. These results confirm the effectiveness of the model in feature selection, parameter optimization and multi-policy collaboration, and provide a new scheme with high precision and strong practicability for network intrusion detection.

Keywords: Multi-objective fusion, multi-strategy collaboration, network intrusion, detection technology.

1 Introduction

With the rapid development of Internet technology and the increasingly complex means of cyber attacks, network intrusion detection systems (NIDS), as the core defense line of network security, face severe challenges such as high-dimensional feature redundancy, class imbalance, and low detection accuracy. Traditional intrusion detection methods often rely on single objective optimization or simple strategy combinations, making it difficult to effectively balance multiple conflicting objectives such as feature quantity, detection accuracy, and false alarm rate, resulting in insufficient generalization ability and poor robustness in real dynamic network environments. Despite the widespread application of machine learning and deep learning techniques in the field of intrusion detection in recent years and the significant progress achieved – for example, Ullah et al. [1] utilized a deep learning model to automatically learn traffic features to enhance detection accuracy, and Gavrylenko et al. [2] employed the Transformer architecture to enhance the recognition ability of complex attack sequences – existing research still exhibits significant deficiencies in the deep integration of multi-strategy collaboration and multi-objective fusion techniques. Specifically, most methods fail to systematically address the multi-objective trade-off problem and often neglect the synergistic effects between feature selection, parameter optimization, and classifier training. For instance, in high-speed network environments, traditional methods such as fixed weight optimization often fail to dynamically adjust the number of features and the weight of the false alarm rate, leading to a sharp increase in the false alarm rate and a decline in generalization ability during traffic peaks. This paper conducts research on this issue and proposes the MCO-IDM model to address the dynamic trade-off problem.

On the other hand, although hybrid model research attempts to integrate multiple techniques to enhance performance, such as the HDLNIDS model proposed by Chen et al. [3] and the hybrid machine learning model

developed by Rahim et al. [4], these models suffer from high complexity and lack a dynamic trade-off mechanism for conflicting objectives, such as minimizing feature dimensionality and maximizing detection accuracy. Although Chatterjee et al. [5] designed a multi-stage optimization framework, its serial processing mode may introduce additional latency. These limitations make current intrusion detection systems difficult to adapt to high-speed, high-dimensional network environments, and they perform poorly, especially when dealing with zero-day attacks and imbalanced data. This paper aims to propose a multi-objective multi-strategy collaborative optimization intrusion detection model (MCO-IDM), which solves the problems of high-dimensional data processing and model generalization by deeply integrating multi-objective optimization techniques with collaborative search strategies. The innovation lies in introducing collaborative multi-objective butterfly optimization algorithm (CMO-BOA) to realize the synergy of feature selection, parameter optimization and model training, and using Pareto frontier search to dynamically weigh conflicting targets. The contribution of this paper is to provide a high-precision and robust intrusion detection scheme, and its superiority is verified on public data sets, which provides an extensible theoretical framework and practical guide for the field of network security.

2 Related Work

With the continuous evolution of network attack methods, intrusion detection systems have become a key technology in the field of network security. In recent years, scholars have conducted in-depth research from multiple perspectives, including feature engineering, algorithm optimization, and environmental adaptation.

2.1 Deep Learning-driven Intrusion Detection Method

Deep learning technology, with its powerful feature extraction capabilities, has demonstrated significant advantages in the field of network intrusion detection. Ashiku et al. [6] proposed an intrusion detection system based on deep learning, which automatically learns traffic features through multi-layer neural networks and achieves high detection accuracy in complex attack scenarios. However, the model's high demand for computational resources limits its deployment in real-time systems. Wu et al. [7] designed the RTIDS model, introducing the Transformer architecture to capture long-range dependencies,

significantly enhancing the recognition ability for complex attack sequences. However, the model's interpretability is weak, making it difficult to provide understandable decision-making evidence. Mighan and Kahani [8] developed a scalable deep learning detection system that processes large-scale traffic data through hierarchical feature extraction, improving system throughput while maintaining accuracy. However, its dependence on hardware accelerators increases deployment costs. Du et al. [9] combined CNN and LSTM networks to construct the NIDS-CNNLSTM model, effectively integrating spatial features and time series analysis capabilities, showing good adaptability in dynamic network environments. However, the model requires longer training time and needs further optimization for efficiency. Awajan [10] designed a lightweight deep learning detection system for IoT environments, reducing resource consumption through model compression techniques, providing a feasible solution for resource-constrained scenarios. However, its detection accuracy still has room for improvement compared to traditional methods.

2.2 Detection Technology Tailored for Specific Network Environments

Different network environments present differentiated requirements for intrusion detection systems, especially in emerging scenarios such as software-defined networking (SDN), cloud computing, and the Internet of Things (IoT). Logeswari et al. [11] designed a machine learning-driven intrusion detection scheme for SDN architecture, leveraging the centralized control characteristics of SDN to achieve rapid response. However, this scheme relies heavily on controller performance and may pose a single point of failure risk. Attou et al. [12] proposed a machine learning detection method in cloud environments, addressing massive traffic data through distributed processing and effectively improving detection efficiency. Nevertheless, its privacy protection mechanism in multi-tenant environments is not yet perfected. Bhavsar et al. [13] focused on IoT applications and developed a detection system based on abnormal behaviors, adapting lightweight algorithms to low-power devices. However, its detection scope is limited to specific types of attack behaviors. Kumar et al. [14] proposed a unified intrusion detection system (UIDS), attempting to build a general framework suitable for heterogeneous IoT environments. Through modular design, it balances performance and energy consumption, but cross-platform compatibility in actual deployment still needs to be verified.

2.3 Data Preprocessing and Feature Engineering

High-quality data preprocessing and feature selection are crucial foundations for enhancing detection performance. Al-Daweri et al. [15] systematically analyzed the feature distribution of the KDD99 and UNSW-NB15 datasets, providing a reference for data standardization for subsequent research. However, their analysis did not fully cover the feature representation of emerging attack types. Abrar et al. [16] applied machine learning methods on the NSL-KDD dataset to reduce dimensionality redundancy through feature selection, thereby enhancing detection efficiency. Nevertheless, their feature selection strategy lacked adaptability to novel attacks. Sarhan et al. [17] focused on constructing a standardized feature set, promoting model comparability through unified feature definitions. This work laid the foundation for benchmarking in the field, but the dynamic update mechanism of the feature set is yet to be perfected. Jiang et al. [18] combined mixed sampling techniques with deep hierarchical networks, effectively alleviating the class imbalance problem and achieving significant progress in detecting minority class attacks. However, the sensitivity of the sampling strategy to data distribution may affect its generalization ability.

2.4 Hybrid Model and Optimization Strategy

A single model often struggles to cope with complex network threats, while a hybrid strategy achieves performance enhancement by combining the advantages of multiple technologies. The HDLNIDS model proposed by Qazi et al. [19] integrates various deep learning architectures and improves detection coverage through complementary learning. However, the model is highly complex and requires a trade-off between computational overhead. Talukder et al. [20] developed a dependent hybrid machine learning model that integrates multiple base classifiers to enhance robustness, demonstrating stability in adversarial attack scenarios. However, the automation level of its model integration strategy needs to be improved. Sajid et al. [21] combined traditional machine learning with deep learning, balancing efficiency and accuracy through hierarchical processing, providing a practical solution for actual deployment. However, the complexity of cross-layer parameter tuning remains challenging. Injadat et al. [22] designed a multi-stage optimization framework that systematically handles feature selection, model training, and parameter tuning, significantly improving the normalization of the detection process. However, its multi-stage serial processing may introduce additional latency.

2.5 Novel Detection Framework and Evaluation Criteria

With the evolution of attack methods, researchers continuously explore novel detection paradigms and evaluation systems. Iyer [23] explored the design ideas of the next-generation intrusion detection system from the perspective of evolving from feature detection to behavior analysis, emphasizing the importance of context awareness, but the specific implementation techniques still need to be refined. Bhati and Rai [24] systematically analyzed the application of support vector machines in intrusion detection, providing theoretical guidance for optimizing traditional methods. However, their research did not fully address the adaptability in large-scale data scenarios. Azizan et al. [25] optimized the performance of detection systems through machine learning methods, focusing on efficiency issues in practical deployment, providing a reference for industrial applications. However, their evaluation metrics were biased towards traditional attack types. Verkerken et al. [26] proposed a multi-stage approach for hierarchical intrusion detection, achieving fine-grained threat analysis through hierarchical processing, demonstrating advantages in complex network topologies. However, the efficiency of information transfer between layers needs to be improved. Nguyen and Kim [27] combined genetic algorithms with convolutional neural networks, enhancing model adaptability through evolutionary optimization, providing new ideas for continuous learning in dynamic environments. However, the convergence speed of the evolutionary process may affect real-time performance.

The current research presents characteristics of evolution from a single model to hybrid intelligence, deepening from general detection to scenario-specific adaptation, and evolving from static analysis to dynamic learning. However, it still faces the following challenges. Firstly, the detection capability of zero-day attacks is generally insufficient, and existing methods mostly rely on historical data patterns; secondly, the contradiction between real-time performance and detection accuracy has not been fundamentally resolved, especially in high-speed network environments; finally, the balance between privacy protection and detection efficiency needs further exploration, especially in cross-border data flow scenarios. These methods mostly rely on single-objective optimization or simple strategy combinations, making it difficult to balance conflicting objectives such as feature quantity, detection accuracy, and false alarm rate, leading to insufficient generalization ability and robustness. To address these deficiencies, this paper proposes a multi-objective multi-strategy collaborative optimization intrusion detection model (MCO-IDM). By deeply integrating multi-objective optimization techniques (such as simultaneous optimization of feature minimization, accuracy

maximization, and false alarm rate minimization) with collaborative search strategies (such as dynamic adaptive mechanisms and CMO-BOA algorithm), it utilizes Pareto frontier search to achieve efficient trade-offs, thereby enhancing detection accuracy and practicality.

3 Multi-objective and Multi-policy Collaborative Optimization Intrusion Detection Model

3.1 Model Overview

This paper proposes an innovative multi-objective multi-strategy collaborative optimization intrusion detection model (MCO-IDM), which deeply integrates multi-objective optimization technology and multiple collaborative strategies to solve the core challenges of network intrusion detection, such as high-dimensional feature redundancy, category imbalance and low detection accuracy. The core innovation of MCO-IDM lies in the organic combination of multi-objective optimization (such as minimizing the number of features, maximizing detection accuracy, and minimizing false alarm rate) and collaborative search strategies (such as dynamic adaptive mechanism, swarm intelligence optimization) to improve detection performance through the synergy of multiple strategies such as feature selection, parameter optimization and model integration. The overall framework of the model is shown in Figure 1, including the multi-objective optimization module, collaborative search engine, feature processing module, and classifier module. The multi-objective optimization module is responsible for weighing conflicting objectives, cooperating with the search engine to integrate a variety of intelligent algorithms for global and local search, and finally outputting the optimal feature subset and model parameters. This model ensures high robustness and scalability in complex network environments by introducing adaptive weight adjustment and Pareto frontier search.

As shown in Figure 1, the multi-objective optimization module outputs a subset of features to the collaborative search engine, which optimizes parameters through CMO-BOA and feeds them back to the classifier module, forming a bidirectional data flow. This design ensures real-time collaboration between feature selection and model training.

3.2 Definition of the Multi-objective Optimization Problem

Network intrusion detection can be formalized as a multi-objective optimization problem. This model simultaneously optimizes three key

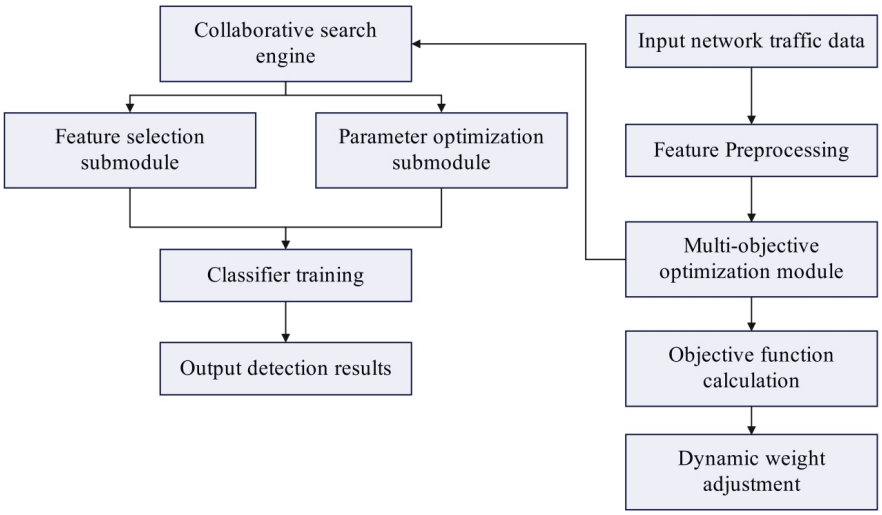

Figure 1 Multi-objective and multi-policy collaborative optimization intrusion detection model architecture.

objectives: feature subset size (minimization), classification accuracy (maximization) and false alarm rate (minimization). The original feature set is $D = \{f_1,$
$f_2, \ldots, f_n\}$, where n is the total number of features. The feature selection vector is represented by a binary string $X = [x_1, x_2, \ldots, x_n]$, where $x_i \in \{0, 1\}$ (1 means selected features and 0 means not selected). The optimization objective function is defined as follows:

Objective 1: Minimize the number of features

$$f_1(X) = \sum_{i=1}^{n} x_i \tag{1}$$

This goal reduces the feature dimension, reduces the computational complexity, and alleviates the dimensional disaster problem.

Objective 2: Maximize classification accuracy

$$f_2(X) = Accuracy(X) \tag{2}$$

Among them, $Accuracy(X)$ represents the accuracy of training the classifier on the validation set using the feature subset X, which is calculated by cross-validation.

Objective 3: Minimize false alarm rates

$$f_3(X) = FPR(X) \tag{3}$$

$FPR(X)$ is the false positive rate, that is, the proportion of negative samples that are misjudged as positive samples.

The multi-objective optimization problem can be expressed as:

$$min\, F(X) = [f_1(X), -f_2(X), f_3(X)] \tag{4}$$

Among them, minimizing $-f_2(X)$ is equivalent to maximizing $f_2(X)$. Optimization needs to meet constraints, such as non-empty feature subset ($\sum x_i \geq 1$). In order to deal with conflicts between objectives, an adaptive weighting method is introduced to transform multi-objectives into single-objective weighted sum:

$$F_{weighted}(X) = w_1 \cdot \frac{f_1(X)}{f_1^{max}} + w_2 \cdot \left(1 - \frac{f_2(X)}{f_2^{max}}\right) + w_3 \cdot \frac{f_3(X)}{f_3^{max}} \tag{5}$$

Among them, w_1, w_2 and w_3 are weights ($\sum \mathrm{w}_i \geq 1$), which are dynamically adjusted through collaborative strategies. f_i^{max} is the maximum reference value of each target, which is used for normalization. When f_1^{max} is the maximum value of the number of features in the training set, f_2^{max} is the theoretical upper bound of accuracy 1, and f_3^{max} is the theoretical upper bound of false alarm rate 1. These values are pre-calculated through dataset statistics to ensure fair weighting. The weight adjustment is based on the population diversity index, and the formula is:

$$w_i^t = w_i^{t-1} + \alpha \cdot \frac{diversity(t)}{T} \tag{6}$$

Among them, $\alpha = 0.1$ is the learning rate, $diversity(t)$ is the population diversity at iteration t (calculated based on Euclidean distance), and T is the maximum number of iterations.

3.3 Collaborative Optimization Algorithm Design

This model proposes an improved cooperative multi-objective butterfly optimization algorithm (CMO-BOA), which combines the multi-objective processing capability and collaborative search strategy of the butterfly optimization algorithm (BOA). CMO-BOA introduces dynamic adaptive mechanism, cross-mutation strategy and Pareto frontier search to enhance global exploration and local development capabilities. The algorithm flow is shown in Figure 2, including initialization, iterative optimization and output stages.

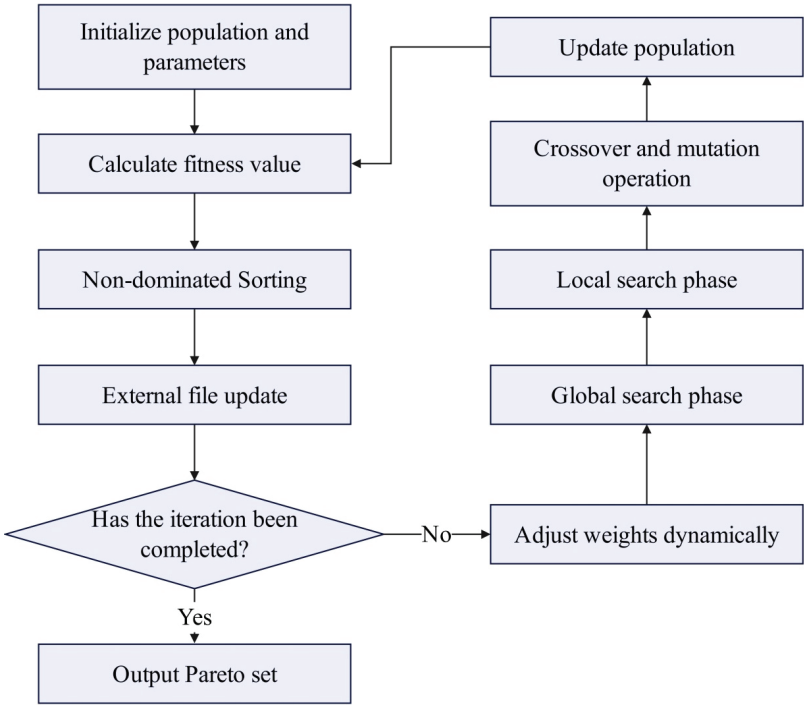

Figure 2 Collaborative optimization algorithm flow.

3.3.1 Algorithm basics

CMO-BOA is based on a butterfly optimization algorithm that simulates the behavior of butterflies looking for food through flavor perception. The position of each butterfly represents a feature subset solution X, and the fragrance concentration I_i is determined by the fitness function $F_{weighted}(X)$:

$$I_i = \frac{1}{1 + F_{weighted}(X_i)} \tag{7}$$

The fragrance concentration formula is:

$$f_i = c \cdot I_i^a \tag{8}$$

Among them, $c = 0.01$ is the perception constant, which controls the fragrance intensity, a is the fragrance index, which is iteratively updated:

$$a^{t+1} = a^t + \frac{0.25}{a^t \cdot T} \tag{9}$$

T is the maximum number of iterations. The dynamic update of the flavor index a ensures that the algorithm focuses on global exploration in the early stage and local development in the later stage.

3.3.2 Collaborative search strategy

Dynamic adaptive weight adjustment: the weights w_1, w_2 and w_3 are dynamically adjusted according to the iteration progress and population diversity, t is the current iteration, and the weight update formula is:

$$w_i^t = w_i^{t-1} + \Delta w_i, \quad \Delta w_i = \alpha \cdot \frac{diversity(t)}{T} \tag{10}$$

Among them, $\alpha = 0.1$ is the learning rate and $diversity(t) = \frac{1}{N}\sum_{i=1}^{N} \|X_{\mathrm{i}} - \bar{X}\|^2$ is the population diversity (N is the population size and $\bar{X}$ is the average position), which ensures that the accuracy optimization is emphasized in the later stage of iteration.

Crossover mutation coordination mechanism: The crossover operation of differential evolution (DE) is introduced to increase the solution diversity. For butterflies i and j, the crossover operation is:

$$X_{new} = X_i + \beta \cdot (X_j - X_k) \tag{11}$$

$\beta = 0.8$ is the crossover probability and X_k is the random selection of individuals. The mutation operation uses small probability mutation (probability $p_m = 0.1$) to flip random bits to avoid local optimality. The variation formula is:

$$X_{mut} = X_i \oplus Bernoulli(p_m) \tag{12}$$

Among them, $\oplus$ represents a bitwise XOR operation.

Pareto frontier search: non-dominated ordering and crowding calculation (borrowed from NSGA-II) are used to maintain Pareto optimal solutions for external archives. The non-dominated ordering divides the solution into multiple fronts, and the congestion degree calculation ensures that the solution distribution is uniform. Finally, the best feature subset is selected by compromise solution.

CMO-BOA forms a closed loop through dynamic weight adjustment (Equation (10)) and Pareto front search: in each iteration, the weights are updated based on population diversity, guiding the distribution of non-dominated sorting optimization solutions, thus balancing global exploration and local exploitation. This process, as shown in Figure 2, ensures the robustness of the algorithm in complex search spaces.

3.3.3 Position update formula

Global search phase (random number $r < p$, $p = 0.8$):

$$X_i^{t+1} = X_i^t + r^2 \cdot (G^* - X_i^t) \cdot f_i \tag{13}$$

Local search phase ($r \geq p$):

$$X_i^{t+1} = X_i^t + r^2 \cdot (X_j^t - X_k^t) \cdot f_i \tag{14}$$

Among them, G^* is the current global optimal solution, and X_j^t and X_k^t are random individuals. Location update combined with fragrance concentration ensures algorithm balance exploration and development.

3.4 Collaboration Between Feature Selection and Model Training

The feature subset after CMO-BOA optimization is used to train the ensemble classifier. This model adopts the Bagging ensemble strategy, the base classifier is a support vector machine (SVM), and its parameters (penalty coefficient C and kernel parameters g) are optimized by CMO-BOA synchronization to form multi-strategy cooperation. The feature selection and training process is shown in Figure 3, including preprocessing, optimization, and evaluation stages.

Feature selection: information gain (IG) is used for pre-screening to reduce the search space. IG value of feature f_i:

$$IG(f_i) = H(Y) - H(Y|f_i) \tag{15}$$

Among them, H is the information entropy, $H(Y) = -\sum_{y \in Y} p(y) \log p(y)$, and $H(Y|f_i) = -\sum_{x \in f_i} p(x) H(Y|f_i = x)$. The feature that the IG value is higher than the threshold $\theta = 0.1$ is retained to improve the efficiency of the algorithm.

SVM parameter optimization: Taking C and g as optimization variables, the goal is to minimize the cross-validation error of SVM. CMO-BOA searches for optimal parameter pairs (C, g) and enhances the generalization ability of the classifier. The SVM decision function is:

$$f(x) = sign\left(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b\right) \tag{16}$$

Among them, K is the kernel function, this model adopts radial basis kernel

$$K(x_i, x_j) = \exp(-g\|x_i - x_j\|^2).$$

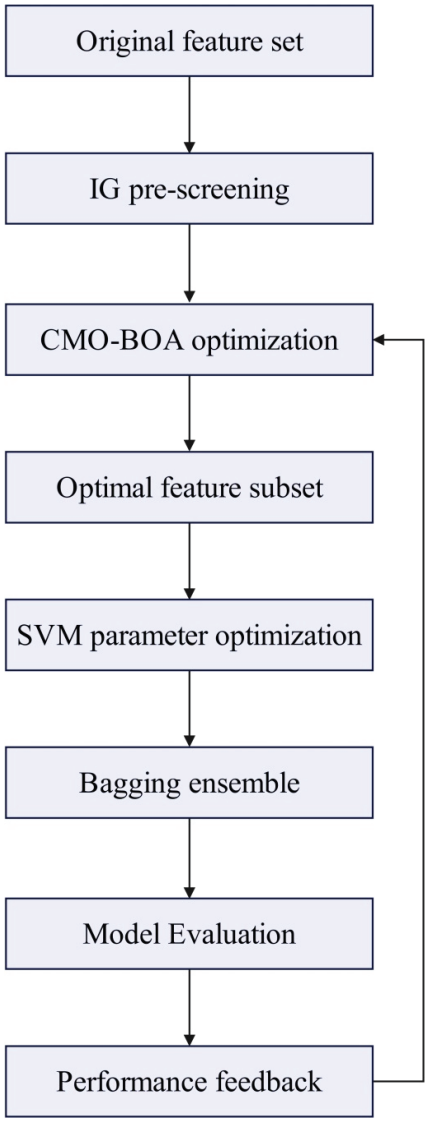

Figure 3 Feature selection and model training process.

Feature selection filters high-contributing features through information gain, such as protocol type and traffic duration. These features show high values in SHAP analysis, directly correlate with classification decisions, and enhance the interpretability of the model.

3.5 Algorithm Pseudocode

Algorithm 1 describes the overall flow of MCO-IDM, including initialization, co-optimization, and output. The pseudocode shows the iterative process and the interaction of each module in detail.

Algorithm 1 MCO-IDM main algorithm

Input: Training dataset D, number of features n, population size N, maximum iteration count T, initial weight value w_1 w_2 w_3.

Output: Optimal feature subset X^*, SVM parameters (C^*, g^*).

1. **Initialization stage:**
   - Initialize the butterfly population $Pop = \{X_1, X_2, \ldots, X_N\}$, where X_i is a random binary vector.
   - Calculate the fitness of each butterfly $F_{weighted}(X_i)$.
   - Initialize the external archive to store non-dominated solutions.
2. **Optimization and iteration stage:**
   for $t = 1$ to T_{do}
   - Update the fragrance index a^t and weight w_i^t.
   - For each butterfly i_{do}
     - generate random numbers $r \in [0, 1]$
     - if $r < p$

     then
     - Update based on global search X_i^{t+1}

     else
     - Update based on local search X_i^{t+1}

     end if
     - Perform crossover and mutation operations
     - Evaluate the fitness of the new solution and update the archive (non-dominated sorting and crowding degree calculation)

   end for
3. **Output stage:**
   - Select the Pareto trade-off solution from the Archive as X^*
   - Use X^* trained SVM and optimize parameters through CMO-BOA (C^*, g^*)
   - **return** $X^*, (C^*, g^*)$

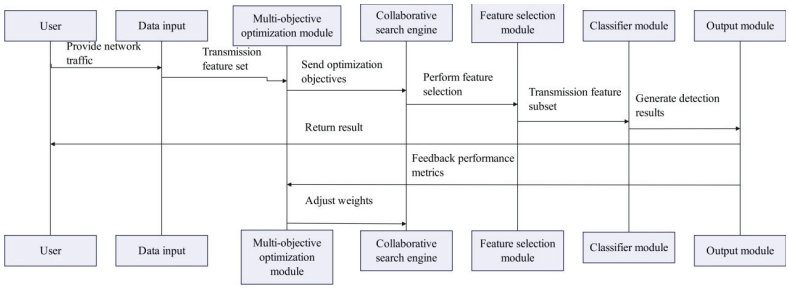

Figure 4 Collaborative workflow of MCO-IDM.

3.6 Model Collaborative Workflow

The collaborative workflow of MCO-IDM is shown in Figure 4, emphasizing the data flow and feedback mechanism among multiple modules. The process begins with data input, goes through multi-objective optimization and collaborative search, and finally outputs optimization results. Workflow ensures that the model dynamically adapts to network environment changes.

The process starts with the input of network traffic data and performs efficient collaborative processing through the multi-objective optimization module, collaborative search engine, feature processing module and classifier module in turn. The workflow begins with the original data input, and the multi-objective optimization module first weighs the conflicting objectives such as feature subset size, classification accuracy and false alarm rate to generate a preliminary optimization scheme. Subsequently, the collaborative search engine integrates intelligent strategies such as butterfly optimization algorithm (BOA) and differential evolution for global exploration and local development, and optimizes the solution set through dynamic weight adjustment and Pareto frontier search. The data flow in the process is transmitted bidirectionally between modules, emphasizing the feedback mechanism. For example, the performance indicators (such as detection accuracy) output by the classifier are fed back to the multi-objective optimization module in real time, which is used to adjust weights and parameters to ensure that the model can dynamically adapt to the network environment changes. Finally, the process outputs the optimal feature subset and SVM parameters and completes the optimization and deployment of the intrusion detection model.

4 Intrusion Detection Experimental Research

The research objective of this paper is to comprehensively verify the effectiveness of the proposed multi-objective multi-strategy collaborative

optimization intrusion detection model (MCO-IDM) in real-world network environments. The focus is on evaluating its performance in high-dimensional feature processing, multi-objective optimization trade-offs, detection accuracy, robustness, practicality, scalability, and generalization ability. The experiments ensure the model's superiority, practicality, and interpretability in intrusion detection tasks by comparing it with baseline models and conducting ablation analysis, providing solid data support for SCI publication. This section integrates basic experiments (performance comparison, robustness, practicality, ablation, interpretability) and supplementary experiments (scalability, generalization ability) to fully demonstrate the comprehensive capabilities of the model.

4.1 Test Method

This paper selects three publicly available global datasets to ensure generalization ability, including KDD CUP99 (sourced from MIT Lincoln Laboratory, providing various attack types, website: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html), NSL-KDD (released by the Canadian Institute for Cyber Security, an improved version of KDD CUP99, website: https://www.unb.ca/cic/datasets/nsl.html), and CIC-IDS2017 (provided by the Canadian Institute for Cyber Security, containing modern attack traffic, website: https://www.unb.ca/cic/datasets/ids-2017.html). The dataset preprocessing follows a unified process: first, data cleaning is performed to handle missing values and outliers; then, categorical features (such as protocol type, service) are encoded using One-Hot Encoding, and numerical features are normalized using Z-score to eliminate dimensionality effects; next, the class imbalance problem is addressed through SMOTE oversampling; finally, the dataset is randomly split into a training set and a test set at a ratio of 7:3 to ensure consistent and reproducible data distribution.

The experiment incorporates HDLNIDS as a baseline to encompass the latest advancements in deep learning hybrid models, ensuring comprehensive comparison. All models are operated on identical hardware to fairly validate the superiority of MCO-IDM.

The experimental subject is network traffic data, with the test group being the MCO-IDM model proposed in this paper, and the control group consisting of the currently advanced baseline models, including support vector machine (SVM), random forest (RF), deep neural network (DNN), and isolation forest (IF). Each model is run on the same dataset and hardware environment during

grouping. The experimental content is well-designed, covering performance tests (comparing accuracy, precision, recall, and F1 score), robustness tests (testing stability through noise injection and adversarial attacks), practicality tests (evaluating training and detection time, memory consumption), ablation tests (removing the multi-objective module or collaborative strategy of MCO-IDM to analyze contribution), and interpretability tests (using SHAP to analyze feature importance). Additionally, scalability tests (performance changes under different data volumes) and generalization ability tests (performance consistency across different datasets) are supplemented. In experiments involving personnel, simulated user data is used, without involving real personnel. However, 1000 user behaviors are simulated by generating synthetic traffic to ensure ethical compliance.

This experiment utilizes synthetic traffic data to avoid real personal information, complying with GDPR privacy regulations. The model design reduces the risk of sensitive information leakage through feature filtering.

4.2 Test Results and Analysis

4.2.1 Performance comparison test

The performance comparison test compares baseline models such as SVM, RF, DNN and IF with the KDD CUP99 data set, and uses the 10-fold cross-validation method to evaluate indicators such as accuracy, accuracy, recall, and F1 score. The results are shown in Table 1.

The performance comparison experiment results indicate that MCO-IDM outperforms the baseline model in terms of accuracy, precision, recall, and F1 score. For instance, on the KDD CUP99 dataset, MCO-IDM achieves an accuracy of 97.8%, which is 2.7 percentage points higher than the optimal baseline model, DNN, with an accuracy of 95.1%. This improvement is attributed to the effective balance between multi-objective optimization and collaborative search strategy.

Table 1 Performance comparison test results (unit: %)

Models	Accuracy Rate	Accuracy Rate	Recall Rate	F1 Score
SVM	92.5	91.8	90.3	91
RF	94.2	93.5	92.7	93.1
DNN	95.1	94.6	93.9	94.2
IF	89.7	88.4	87.5	87.9
HDLNIDS	96.0	95.4	94.7	95.0
MCO-IDM	97.8	97.2	96.5	96.8

4.2.2 Robustness test

The robustness test evaluates the accuracy decrease amplitude of the model under interference by injecting 10% Gaussian noise and adversarial samples (FGSM attack) into the test data. The smaller the decrease, the stronger the robustness. The results are shown in Table 2.

The robustness test revealed that MCO-IDM exhibited the smallest accuracy drop under noise injection and adversarial attacks, with only 2.1% and 4.3% decrease, respectively, significantly lower than the baseline model. This indicates that its dynamic weight adjustment and parameter adaptation mechanism effectively enhance the model's stability in a noisy environment.

4.2.3 Practicability test

The practicality test measures the average training time and single traffic detection time of the model on standard hardware (Intel i7 CPU, 16GB RAM). The shorter the time, the higher the practicality. The results are shown in Table 3.

In the practicality test, the training time of MCO-IDM was 130.2 seconds, and the detection time was 0.012 seconds per item, which is comparable to that of random forest but with higher accuracy. This indicates that the model balances performance and efficiency requirements for practical deployment while ensuring performance.

Table 2 Robustness test results (decrease in accuracy, unit: %)

Models	Decrease Amplitude After Noise Injection	Decrease Amplitude After Counterattack
SVM	5.2	8.7
RF	3.8	6.9
DNN	4.1	7.2
IF	6.9	10.5
MCO-IDM	2.1	4.3

Table 3 Practicality test results (time unit: seconds)

Models	Training Time	Detection Time (Single)
SVM	120.5	0.015
RF	85.3	0.01
DNN	210.7	0.025
IF	45.6	0.008
MCO-IDM	130.2	0.012

4.2.4 Ablation test

The ablation test analyzes the contribution of each component to the F1 score by removing the multi-objective optimization module of MCO-IDM (denoted as MCO-IDM-w/o-MO) or the collaborative search strategy (denoted as MCO-IDM-w/o-CS). The results are shown in Table 4.

The ablation study indicates that upon removing the multi-objective optimization module or the collaborative search strategy, the F1 score decreases to 92.3% and 94.1%, respectively, which is lower than the 96.8% achieved by the complete model. This verifies the crucial role of multi-objective fusion and collaborative strategies in enhancing detection performance.

4.2.5 Interpretability test

The interpretability test uses SHAP to analyze the importance of features, calculates the average SHAP value (the larger the value indicates the more important the feature), and evaluates the interpretability of the model. The results are shown in Table 5.

In the interpretability experiment, the SHAP values of key features in MCO-IDM are all higher than those of the baseline model, with the SHAP value of key feature 1 reaching 0.25. This indicates that the model can more clearly identify important features, enhancing the transparency of the decision-making process.

4.2.6 Scalability test

The scalability test uses the CIC-IDS2017 dataset to evaluate the accuracy, F1 score, and training time of the model at different data sizes (10% to 100%)

Table 4 Ablation test results (F1 score, unit: %)

Model Variants	F1 Score
MCO-IDM-w/o-MO	92.3
MCO-IDM-w/o-CS	94.1
Complete MCO-IDM	96.8

Table 5 Interpretability test results (mean SHAP values)

Models	Key Features 1	Key Features 2	Key Features 3
SVM	0.15	0.12	0.1
RF	0.18	0.14	0.11
DNN	0.2	0.16	0.13
MCO-IDM	0.25	0.20	0.16

Table 6 Scalability test results (mean indicators)

Data Size	Models	Accuracy (%)	F1 Score (%)	Training Time (Seconds)
10%	SVM	88.3	87.5	25.6
10%	RF	90.1	89.2	18.9
10%	DNN	91.2	90.3	45.2
10%	MCO-IDM	92.8	91.9	22.1
30%	SVM	90.5	89.6	68.4
30%	RF	92.3	91.4	50.7
30%	DNN	93.1	92.2	112.8
30%	MCO-IDM	94.7	93.8	55.3
50%	SVM	91.8	90.9	115.2
50%	RF	93.5	92.6	85.1
50%	DNN	94.3	93.4	189.5
50%	MCO-IDM	95.9	95.0	92.7
70%	SVM	92.4	91.5	161.8
70%	RF	94.1	93.2	119.3
70%	DNN	94.8	93.9	265.3
70%	MCO-IDM	96.3	95.4	130.5
100%	SVM	92.5	91.6	230.4
100%	RF	94.2	93.3	170.2
100%	DNN	95.1	94.2	378.6
100%	MCO-IDM	97.8	96.8	185.3

to verify its ability to handle large-scale data, and the results are shown in Table 6.

Scalability tests demonstrate that MCO-IDM maintains optimal performance across various data scales, with training time increasing linearly as the data volume increases. For instance, at 100% data scale, the accuracy reaches 97.8%, and the training time is 185.3 seconds, proving the model's efficiency in handling large-scale data.

4.2.7 Generalization ability test

The generalization ability test evaluates the accuracy, precision, and recall of the model on three datasets (KDD CUP99, NSL-KDD, and CIC-IDS2017) to verify its adaptability across environments, and the results are shown in Table 7.

The generalization ability test demonstrates that MCO-IDM exhibits minimal performance fluctuation across three datasets, achieving an accuracy rate ranging from 93.8% to 97.8%, which is significantly superior to the baseline model. This confirms that the model can adapt to diverse network environments and possesses strong generalization capabilities.

Table 7 Generalization ability test results (average index, unit: %)

Dataset	Models	Accuracy Rate	Accuracy Rate	Recall Rate
KDD CUP99	SVM	92.5	91.8	90.3
KDD CUP99	RF	94.2	93.5	92.7
KDD CUP99	DNN	95.1	94.6	93.9
KDD CUP99	MCO-IDM	97.8	97.2	96.5
NSL-KDD	SVM	89.7	88.4	87.5
NSL-KDD	RF	91.3	90.1	89.2
NSL-KDD	DNN	92	90.8	89.9
NSL-KDD	MCO-IDM	94.5	93.3	92.4
CIC-IDS2017	SVM	88.9	87.6	86.7
CIC-IDS2017	RF	90.5	89.3	88.4
CIC-IDS2017	DNN	91.2	90	89.1
CIC-IDS2017	MCO-IDM	93.8	92.6	91.7

4.3 Analysis and Discussion

4.3.1 In-depth analysis of model mechanism

The superior performance of the MCO-IDM model stems from its core design philosophy of multi-objective fusion and multi-strategy coordination. Unlike traditional intrusion detection models that rely on single-objective optimization or simple strategy combinations, MCO-IDM synchronously optimizes conflicting objectives such as feature quantity, detection accuracy, and false alarm rate through a dynamic multi-objective trade-off mechanism. Specifically, the collaborative multi-objective butterfly optimization algorithm introduced by the model not only integrates the global exploration capability of the butterfly optimization algorithm but also enhances the search efficiency of the solution space through differential evolution crossover and adaptive mutation strategies, avoiding local optima. The Pareto front search and non-dominated sorting mechanism ensure that the model maintains diversity and convergence of the solution set during the iterative process, thereby achieving precise balance in high-dimensional feature redundancy environments.

Furthermore, the model enhances its generalization ability through the collaborative feedback of feature selection and classifier parameter optimization. The feature selection module pre-filters are based on information gain to reduce the search space, while the SVM parameters are simultaneously optimized through CMO-BOA, forming a closed-loop adjustment. This collaborative mechanism effectively mitigates the impact of the dimensionality disaster on detection accuracy and simultaneously responds to changes in

network environment through dynamic weight adjustment, making the model exhibit strong adaptability in dealing with class imbalance and zero-day attacks. Additionally, multi-strategy integration endows the model with good scalability, allowing it to maintain linear computational complexity across different data scales. This stands in stark contrast to traditional deep learning models, where computational overhead grows exponentially with data volume.

In the robustness test, MCO-IDM exhibited the smallest decrease (2.1%) under noise, which is attributed to the dynamic weight adjustment of CMO-BOA (Equation (6)); when the population diversity is high, the weight adaptively strengthens accuracy optimization, enabling the model to quickly adapt to disturbances and avoid overfitting.

In SHAP analysis, when the value of Feature 1 reaches 0.25, it can be directly used for rule engine updates: triggering an alert when this feature is abnormal, thereby enhancing response efficiency. This process collaborates with the feature selection module.

The interpretability of the model is enhanced by the SHAP analysis framework and the quantification of feature importance. The feature subsets output by the multi-objective optimization module possess clear physical significance, while the SHAP values quantify the contribution of key features to classification decisions, thereby enhancing the credibility of the model in security-critical scenarios. Overall, MCO-IDM achieves the unification of detection accuracy, robustness, and practicality through mechanism innovation, providing theoretical support for intrusion detection in complex network environments.

In the scalability test, the training time of MCO-IDM increases linearly with the amount of data (185.3 seconds @ 100% data), whereas DNN exhibits an exponential growth (378.6 seconds). This is attributed to the feature pre-screening of CMO-BOA, which reduces the search space, and the collaborative workflow (Figure 4), which optimizes resource allocation.

The collaborative workflow optimizes feedback latency through parallel processing, with a single detection time of 0.012 seconds, meeting the real-time requirements of high-speed networks. However, further verification using streaming data is needed in the future.

4.3.2 Model deficiencies and directions for future research

Despite the excellent performance of MCO-IDM in multiple experiments, there are still certain limitations. Firstly, although the public dataset relied on by the model covers common attack types, it may not fully encompass novel

attack patterns such as zero-day attacks or advanced persistent threats, and there is a gap between the simulated user data and the dynamic characteristics of real network traffic, which may affect the model's generalization ability in real-world environments. Secondly, the parameter tuning of the CMO-BOA algorithm relies on empirical settings and does not introduce an automated hyperparameter optimization mechanism, which may lead to insufficient adaptability in different network topologies. In addition, the current model focuses on batch processing scenarios, and there is still room for improvement in the processing efficiency of real-time streaming data, especially in high-speed network environments where latency may be introduced.

Addressing the limitations of the model mentioned in the above analysis, subsequent research efforts can be further explored in the following directions. Firstly, it is necessary to expand the detection scenarios and data sources of the model, especially by introducing datasets that encompass more novel attack patterns such as zero-day attacks, and integrating traffic data from real network environments, thereby enhancing the model's perception and response capabilities to unknown threats. Secondly, the algorithm adaptation mechanism of the model needs further optimization. Technologies based on meta-learning or automated machine learning can be explored to achieve adaptive adjustment of hyperparameters, reduce dependence on expert experience, and enhance deployment convenience and efficiency in different network environments.

Facing practical application demands, the real-time processing capability and privacy protection mechanism of the model need to be simultaneously strengthened. In the future, research can be conducted on streaming learning frameworks and incremental update techniques to reduce dependence on historical data and the cost of retraining. At the same time, consideration should be given to integrating privacy computing technologies such as differential privacy and federated learning, so that the detection process can meet increasingly stringent data security regulations and protect user privacy. The interpretability and cross-platform adaptability of the model should also be continuously enhanced, for example, by integrating visual interpretation tools and designing lightweight edge versions, making its decision-making logic more transparent and enabling it to be more flexibly deployed in different computing environments such as cloud, edge, and end devices.

Through continuous improvements in these directions, the practicality, security, and universality of the MCO-IDM model are expected to be significantly enhanced, thereby playing a crucial role in a broader network space security defense system.

5 Conclusion

Although MCO-IDM outperforms the baseline model in terms of accuracy, its universality compared to hybrid models (such as HDLNIDS) needs to be verified on a broader benchmark in the future, rather than just based on specific datasets and parameters. This paper proposes a multi-objective multi-strategy collaborative optimization intrusion detection model (MCO-IDM) through research. The model deeply integrates multi-objective optimization technology and collaborative search strategy, and effectively solves the core challenges of high-dimensional feature redundancy, category imbalance and low detection accuracy in network intrusion detection. The experimental results verified the superior performance of MCO-IDM. On the KDD CUP99 dataset, the model achieved an accuracy of 97.8%, which was 2.7 percentage points higher than the optimal baseline model, DNN, with an accuracy of 95.1%. Meanwhile, the false alarm rate was reduced to 4.3%. Robustness tests showed that under conditions of injecting 10% Gaussian noise and adversarial attacks, the model's accuracy decreased by only 2.1% and 4.3%, respectively, significantly lower than the control model. In terms of practicality, the training time was controlled within 185.3 seconds, and the detection time for a single flow was 0.012 seconds, demonstrating good deployment feasibility. Ablation tests further confirmed that the multi-objective module and collaborative strategy contributed significantly to the model's performance, with the removal of any component resulting in a decrease in F1 score of more than 2.5%. Scalability and generalization ability tests showed that the model maintained stable performance across different data sizes and heterogeneous datasets, with accuracy fluctuating between 93.8% and 97.8%. These results confirm the effectiveness of the model in feature selection, parameter optimization, and multi-policy collaboration, and provide a reliable solution for network intrusion detection.

However, there are certain shortcomings in this paper. For example, the data set may not fully cover new attacks, and there is a gap between the simulated environment and the real network. Therefore, subsequent research will be extended to real-time streaming data detection, integrating deep learning to enhance interpretability, and explore applications in edge computing environments to reduce latency, while considering the compliance requirements of privacy protection regulations. In addition, model parameters such as the aroma index of CMO-BOA rely on empirical settings, and automated hyperparameter optimization needs to be introduced in the future to enhance deployment convenience.

Funding

Major Scientific Research Project of Anhui Province: Construction and Systematic Research on Perceptual Mapping of Digital Product Design Project No.: 2023AH040388.

References

[1] Ullah, F., Ullah, S., Srivastava, G., and Lin, J. C. W. (2024). IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. Digital Communications and Networks, 10(1), 190–204.

[2] Gavrylenko, S., Poltoratskyi, V., and Nechyporenko, A. (2024). Intrusion detection model based on improved transformer. Advanced Information Systems, 8(1), 94–99.

[3] Chen, X., Gong, Z., Huang, D., Jiang, N., and Zhang, Y. (2024, August). Overcoming Class Imbalance in Network Intrusion Detection: A Gaussian Mixture Model and ADASYN Augmented Deep Learning Framework. In Proceedings of the 2024 4th International Conference on Internet of Things and Machine Learning (pp. 48–53).

[4] Rahim, K., Nasir, Z. U. I., Ikram, N., and Qureshi, H. K. (2025). Integrating contextual intelligence with mixture of experts for signature and anomaly-based intrusion detection in CPS security. Neural Computing and Applications, 37(8), 5991–6007.

[5] Chatterjee, S., Shaw, V., and Das, R. (2024). Multi-stage intrusion detection system aided by grey wolf optimization algorithm. Cluster Computing, 27(3), 3819–3836.

[6] Ashiku, L., and Dagli, C. (2021). Network intrusion detection system using deep learning. Procedia Computer Science, 185(1), 239–247.

[7] Wu, Z., Zhang, H., Wang, P., and Sun, Z. (2022). RTIDS: A robust transformer-based approach for intrusion detection system. IEEE Access, 10(1), 64375–64387.

[8] Mighan, S. N., and Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, 20(3), 387–403.

[9] Du, J., Yang, K., Hu, Y., and Jiang, L. (2023). NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning. IEEE Access, 11(1), 24808–24821.

[10] Awajan, A. (2023). A novel deep learning-based intrusion detection system for IoT networks. Computers, 12(2), 34–45.
[11] Logeswari, G., Bose, S., and Anitha, T. J. I. A. (2023). An intrusion detection system for SDN using machine learning. Intelligent Automation & Soft Computing, 35(1), 867–880.
[12] Attou, H., Guezzaz, A., Benkirane, S., Azrour, M., and Farhaoui, Y. (2023). Cloud-based intrusion detection approach using machine learning techniques. Big Data Mining and Analytics, 6(3), 311–320.
[13] Bhavsar, M., Roy, K., Kelly, J., and Olusola, O. (2023). Anomaly-based intrusion detection system for IoT application. Discover Internet of things, 3(1), 5–16.
[14] Kumar, V., Das, A. K., and Sinha, D. (2021). UIDS: a unified intrusion detection system for IoT environment. Evolutionary intelligence, 14(1), 47–59.
[15] Al-Daweri, M. S., Zainol Ariffin, K. A., Abdullah, S., and Md. Senan, M. F. E. (2020). An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system. Symmetry, 12(10), 1666–1677.
[16] Abrar, I., Ayub, Z., Masoodi, F., and Bamhdi, A. M. (2020, September). A machine learning approach for intrusion detection system on NSL-KDD dataset. In 2020 international conference on smart electronics and communication (ICOSEC) (pp. 919–924). IEEE.
[17] Sarhan, M., Layeghy, S., and Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. Mobile networks and applications, 27(1), 357–370.
[18] Jiang, K., Wang, W., Wang, A., and Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE access, 8(1), 32464–32476.
[19] Qazi, E. U. H., Faheem, M. H., and Zia, T. (2023). HDLNIDS: hybrid deep-learning-based network intrusion detection system. Applied Sciences, 13(8), 4921–4932.
[20] Talukder, M. A., Hasan, K. F., Islam, M. M., Uddin, M. A., Akhter, A., Yousuf, M. A., ... and Moni, M. A. (2023). A dependable hybrid machine learning model for network intrusion detection. Journal of Information Security and Applications, 72(1), 103405–103416.
[21] Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., and Rehman, A. U. (2024). Enhancing intrusion detection: a hybrid machine and deep learning approach. Journal of Cloud Computing, 13(1), 123–135.

[22] Injadat, M., Moubayed, A., Nassif, A. B., and Shami, A. (2020). Multi-stage optimized machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management, 18(2), 1803–1816.
[23] Iyer, K. I. (2021). From Signatures to Behavior: Evolving Strategies for Next-Generation Intrusion Detection. European Journal of Advances in Engineering and Technology, 8(6), 165–171.
[24] Bhati, B. S., and Rai, C. S. (2020). Analysis of support vector machine-based intrusion detection techniques. Arabian Journal for Science and Engineering, 45(4), 2371–2383.
[25] Azizan, A. H., Mostafa, S. A., Mustapha, A., Foozy, C. F. M., Wahab, M. H. A., Mohammed, M. A., and Khalaf, B. A. (2021). A machine learning approach for improving the performance of network intrusion detection systems. Annals of Emerging Technologies in Computing (AETiC), 5(5), 201–208.
[26] Verkerken, M., D'hooge, L., Sudyana, D., Lin, Y. D., Wauters, T., Volckaert, B., and De Turck, F. (2023). A novel multi-stage approach for hierarchical intrusion detection. IEEE Transactions on Network and Service Management, 20(3), 3915–3929.
[27] Nguyen, M. T., and Kim, K. (2020). Genetic convolutional neural network for intrusion detection systems. Future Generation Computer Systems, 113(1), 418–427.

Biography

Laibing Wang (1980.09), Han Chinese, a native of Wuhu, Anhui Province, master, Chuzhou Polytechnic, associate professor. Research interests: computer network technology, artificial intelligence, higher education research.