IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 2, 2021, accepted June 24, 2021, date of publication June 29, 2021, date of current version July 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093313

An Online Network Intrusion Detection Model Based on Improved Regularized Extreme Learning Machine

YANQIANG TANG AND CHENGHAI LI, (Member, IEEE)
Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China

Corresponding author: Chenghai Li (lichenghai_ns@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703426, in part by the China Postdoctoral Science Foundation under Grant 2018M633680, and in part by the Young Talent Fund of University Association for Science and Technology in Shaanxi, China, under Grant 20190108.

ABSTRACT Extreme learning machine (ELM) is a novel single-hidden layer feedforward neural network to obtain fast learning speed by randomly initializing weights and deviations. Due to its extremely fast learning speed, it has been widely used in training of massive data in recent years. In order to adapt to the real network environment, based on the ELM, we propose an improved particle swarm optimized online regularized extreme learning machine (IPSO-IRELM) intrusion detection algorithm model. First, the model replaces the traditional batch learning with sequential learning by dynamically adapting the new data obtained in the training network instead of training all collected samples in an offline manner; second, we improve the particle swarm optimization algorithm and compare it with typical improved algorithms to prove its effectiveness; finally, to solve the random initialization problem of IRELM, we use IPSO to optimize the initial weights and deviations of IRELM to improve the classification ability of IRELM. The experimental results show that IPSO-IRELM algorithm has better generalization ability, which not only improves the accuracy of intrusion detection, but also has certain recognition ability for minority class samples.

INDEX TERMS Online regularized extreme learning machine, sequential learning, intrusion detection, improved particle swarm optimization, network security.

I. INTRODUCTION

The widespread use of information technology and the emergence and development of cyberspace have greatly contributed to economic and social prosperity and progress, but at the same time brought new security risks and challenges. Network intrusion detection system is the first step of network security situation awareness and an important part of comprehensive network security defense, which perceives whether there are behaviors and signs of network intrusion by analyzing various traffic data of key nodes in the network, so as to be fully prepared for network defense [1]. Traditional network intrusion detection systems build models based on pattern matching methods [2], [3], collecting attack samples first and then training all collected samples in an offline manner, which has an inherent drawback of not detecting emerging types of attacks and lacks adaptiveness and scalability. With the development of network automation and intelligence, new types of adaptive and dynamically scalable intrusion detection systems have emerged [4], [5]. With the advantages of adaptive, self-learning, self-organization, better fault tolerance and the ability to perform massively parallel computation and nonlinear mapping, neural networks are very suitable for variable intrusion detection systems [6]. Moreover, the most important thing in switching from offline learning to online learning is the time problem [7], if the learning time is too long, the online intrusion detection system will have the problem of not being able to detect the attack in time, which leads to the system paralysis. Therefore, this paper uses the extreme learning machine (ELM) with fast learning characteristics as the basis.

Extreme learning machine [8] is a typical classification algorithm in machine learning, characterized by randomly selected input layer weights and hidden layer deviations.

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang.

It is based on a single hidden layer feedforward neural network and computationally resolves the output layer weights according to Moore-Penrose generalized inverse matrix theory, and thus has the advantages of few training parameters, fast learning speed, and strong generalization ability. Its good generalization performance and extremely fast learning speed have been successfully applied to many real-world problems [9], [10]. However, ELM is vulnerable to the interference of outlier sample points during training, which affects the classification accuracy of the model. Therefore, Jose' *et al.* [11] proposed a regularized extreme learning machine (RELM) based on ELM. RELM considers the structural error while solving the least squares error, which effectively avoids the overfitting problem caused by the excessive number of hidden layers and can further improve the classification. Although RELM solves the overfitting problem compared to ELM, RELM also has some problems, such as the selection of the appropriate regularization factor is random and time-consuming. Several scholars have also studied the improvement of RELM: to automatically select a satisfactory regularization factor, Zhang *et al*. [12] proposed an adaptive RELM with a function instead of a regularization factor; Gautam *et al*. [13] proposed an ELM with a regularization kernel, used a single-class ELM classifier based on the regularization kernel to detect outliers, and extended it to adaptive online learning, whose experimental results show that the classifier has faster learning capability and is more suitable for real-time anomaly detection; reference [14] proposed a new binary grey wolf optimization-regularized extreme learning machine wrapper; reference [15] used adaptive whale optimization algorithm (AWOA) to determine the input weights and hidden layer deviations of ELM.; Kumar *et al.* [16]and Zhi *et al.* [17] proposed biogeography-based extreme learning machine (BBO-ELM)model and a GAPSO-Enhanced ELM method respectively. They also compared them with genetic algorithm (GA)-based ELM, and particle swarm optimization (PSO)-based ELM, to verify the effectiveness of the proposed methods. Kanimozhi and Singaravel [18] and Wang *et al.* [19] have combined PSO with ELM and applied it to different scenarios, both of which have yielded good results. However, the improved RELM, if used directly for intrusion detection, still uses batch learning and trains all samples obtained once, and subsequently does not learn new knowledge and still fails to detect new attacks in the network.

To address this problem, this paper continues to propose an online regularized ELM (IRELM) based on RELM, which has the ability of dynamic sequential learning and adaptively learns the constant flow of traffic in the network for intrusion detection; meanwhile, we improve the PSO algorithm to reduce the probability of falling into local extremum points, which in turn better optimizes IRELM. In summary, IPSO-IRELM has better classification performance in all experiments. The research in this paper will improve the adaptiveness and expandability of the intrusion detection system.

The main contributions of this paper are as follows:

1. Proposing an online regularized ELM based on the RELM, which dynamically trains the added data instead of the traditional batch learning.

2. Adding a perturbation mechanism to PSO, and the improved PSO is compared with other improved classical PSO to prove its effectiveness.

3. IPSO optimizes the initial weights and deviations of IRELM to avoid the impact of randomly generated initial weights and biases on the final results.

The complete model is used in intrusion detection and the model has better recognition capability.

The remainder of this paper is organized as follows.

In Section II, we present the current studies related to our work and summarize the parameters used by others' techniques and the limitations in Table 1. In Section III, we describe the IPSO-IRELM algorithm in detail. In Section IV, we derive the comparison results between IPSO-IRELM and other algorithms on the UCI dataset, NSL-KDD binary classification dataset, NSL-KDD multivariate classification dataset, and UNSW-NB15 multivariate classification dataset. In Section V, we summarize the research of this paper and point out the next research directions afterwards.

II. RELATED WORK

The RELM is proposed mainly to solve the problem that the standard ELM is affected by outlier points resulting in low generalization ability and lack of stability [20]. Regularization is essentially a structural risk minimization strategy that adds a regularization term representing the complexity of the model to the empirical risk. The standard mathematical model of RELM is represented as follows:

$$\min \frac{1}{2} \|\boldsymbol{\beta}\|_p^{\sigma_1} + \frac{C}{2} \|\boldsymbol{\xi}\|_q^{\sigma_2} \tag{1}$$

where $\sigma_1 > 0, \sigma_2 > 0; p, q = 0, \frac{1}{2}, 1, 2, \ldots, +\infty$; $\|\cdot\|_p$ is the L_p norm of the vector or matrix; $\|\boldsymbol{\beta}\|_p^{\sigma_1}$ is the regularization term, indicating the complexity of the model; $\|\boldsymbol{\xi}\|^{\sigma_2}$ is the total error of training and represents the empirical risk; C is a regularization parameter to balance the empirical risk and model complexity. When $\sigma_1 = \sigma_2 = 2$, (1) is a quadratic programming problem under the equation constraint. When the regularization term is the L_2 norm and L_1 norm of the parameter vector, it is called ridge regression (L_2 regularization) and LASSO (L_1 regularization), respectively, which are the two most typical regularization methods. Deng *et al*. [21] studied L_2 RELM with hidden layer neurons as Sigmoid functions, and proposed the Unweighted RELM and Weighted RELM (WRELM) algorithms for the presence of noise in the dataset. WRELM uses weighted least squares to calculate the output weights, which has some noise anti-interference ability, but the training process adds the process of calculating the weights of errors, which leads to increased time consumption when the training data is large. Huang *et al*. [22] proposed Semi-Supervised ELM (SS-ELM) and Unsupervised

ELM (US-ELM) algorithms based on stream regularization theory to deal with the relationship between unlabeled samples, which greatly extended the applicability of ELM. Yu and Sun [23] proposed a Sparse coding ELM algorithm (ScELM), which uses a sparse coding technique instead of random mapping to map the input feature vector to the hidden layer to improve the classification accuracy. An optimization method based on gradient projection and L_2 norm is used in the coding stage, while the output weights are derived by the Lagrange multiplier method. Zhao *et al*. [24] proposed the Robust ELM algorithm (RRELM) by introducing both the deviation and variance of the model into the objective function for optimization and keeping the L_2 penalty term constant. RRELM considers both the variance and deviation of the model and seeks to achieve the best compromise between them to enhance the network generalization performance and robustness. For the classification problem of imbalanced data, Xiao *et al*. [25] proposed the Class-specific Cost Regulation ELM algorithm (CCR-ELM), which achieves a compromise between the number of misclassified samples and the generalization ability of the model by applying different penalty factors to misclassified samples of different classes. However, since the number of hidden layer nodes, positive and negative sample weights, and kernel parameters in CCR-ELM have a large impact on the performance of the model, how to develop a more effective method to determine these parameters needs to be further investigated.

To address the problem that the input weights and hidden layer deviations of ELM are randomly generated, some scholars have also optimized the network structure of ELM. Zhu *et al*. [26] introduced differential evolution to ELM and proposed an Evolutionary ELM algorithm (E-ELM), which uses differential variation and crossover operators with simple structure and searches for optimal input weights and hidden layer deviations according to the dynamic adjustment of the population to obtain a more compact network structure. Xu and Shu [27] used the good global search ability of particle swarm optimization algorithm to optimize the number of hidden layer neurons of ELM, and proposed a PSO-ELM algorithm, which encodes the input weights and implied layer deviations of ELM as particles in the PSO search space, and in each iteration all particles update their positions by their own historical optimal solutions and the current global optimal solutions of the whole population to achieve the search for the optimal value in the solution space. The effectiveness of PSO optimization depends on its topology. Figueiredo and Ludermir [28] studied the effect of eight different topologies on PSO-ELM performance, and the results also did not find the best topology for all problems.

The above research on RELM shows that introducing regularization into ELM can solve the overfitting problem to a certain extent and improve the robustness and generalization ability of the model. However, the learning efficiency of the algorithm is reduced due to the addition of the regularization parameter in the objective function that needs to be optimized. Studies on the optimization of network structures for ELM have shown that there is no optimal topology for all problems, but there is always a topology that is suitable for a particular problem. Meanwhile, all the above studies do not adopt a dynamic way to train the data. Therefore, this paper combines both RELM and optimization of ELM network structure, and proposes an online regularized extreme learning machine model, which not only solves the initialization problem of ELM and improves the generalization ability of the model, but also transforms offline learning into online learning without reducing the learning efficiency of the algorithm, and has the ability to be more adaptable to modern intrusion detection. We have summarized the current partial RELM models, the optimized ELM network structure models, and our proposed model in Table. 1, and compared the algorithmic ideas, regularization methods used, feature mapping, robustness, evaluation data, and the drawbacks.

III. IPSO-IRELM ALGORITHM

The idea of RELM is to solve the pathological problem of the hidden layer matrix when ELM fails by limiting the parity of the output weights of ELM, while sacrificing the bias to solve the overfitting problem, thus improving the overall generalization ability [29]. However, RELM still suffers from the generic problem of ELM, where random input weights and hidden layer deviations can be potentially unreliable and unstable, affecting the classification performance. Therefore, we use the improved classical PSO algorithm for the initialization of RELM, and combine it with the actual situation by replacing the batch learning idea with the sequential learning idea, so that the matrix multiplication and inverse decomposition in the output weight calculation process are gradually updated, while adding various mechanisms to reduce the computational load and maintain the efficiency of RELM.

A. IRELM BASED ON SEQUENTIAL LEARNING

RELM adds a regularization parameter to ELM to adjust the coefficient β. The objective function is as follows:

$$\min \|\boldsymbol{H}\boldsymbol{\beta} - \boldsymbol{T}\|^2 + \frac{1}{C}\|\boldsymbol{\beta}\|^2 \tag{2}$$

where $\boldsymbol{H}$ is the hidden layer output matrix; $\boldsymbol{\beta}$ is the output weight matrix;$\boldsymbol{T}$ is the objective matrix. Therefore, the output weight can be expressed as:

$$\begin{aligned}\boldsymbol{\beta} &= \boldsymbol{H}^{\dagger}\boldsymbol{T} = (\boldsymbol{H}^T\boldsymbol{H} + \frac{\boldsymbol{I}}{C})^{-1}\boldsymbol{H}^T\boldsymbol{T} \\ &= \boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T + \frac{\boldsymbol{I}}{C})^{-1}\boldsymbol{T}\end{aligned} \tag{3}$$

When the input data size n is less than the number of neurons in the hidden layer L, because the dimension of $(\boldsymbol{H}\boldsymbol{H}^T + \frac{\boldsymbol{I}}{C})^{-1}$ is $\boldsymbol{n} \times \boldsymbol{n}$ and the dimension of $(\boldsymbol{H}^T\boldsymbol{H} + \frac{\boldsymbol{I}}{C})^{-1}$ is $L \times L$. Therefore, when $n < L$, the IRELM is constructed by $\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T + \frac{\boldsymbol{I}}{C})^{-1}\boldsymbol{T}$.

Define a recursive hidden layer output matrix $\boldsymbol{H}_k$, where k is 0 or any positive integer, representing the number of sequential updates in the matrix. Assuming that the first

TABLE 1. Comparison of various improved ELM models.

Model	Core Ideas	Regularization methods			Feature Mapping	Robustness	Evaluation datasets	Defects
		L_1	L_2	Manifold				
WRELM	Structural risk Weighted residuals		√		Sigmoid	√	"SinC" Benchmark	Too long training time
SS-ELM US-ELM	Graph embedding		√	√	Sigmoid	√	5 semi-supervised	Increased algorithm complexity
ScELM	Sparse coding Gradient projection	√			Sigmoid		8 classified	Less robustness
RRELM	Deviation variance minimization		√		Sigmoid Fourier	√	5 UCI	Inability to handle large data sets efficiently
CCR-ELM	Loss penalty for misclassified samples		√		Sigmoid Gaussian	√	19 UCI	The parameters of the model have too much influence on the performance
E-ELM	Differential evolution				Sigmoid		4 benchmark	Parameters need to be set manually
PSO-ELM	Particle swarm optimization				Sigmoid		9 benchmark	No topology suitable for all problems
IPSO-IRELM	Improved particle swarm optimization Online Learning		√		Sigmoid	√	2UCI 2 classified	The addition of the network topology causes a slight increase in training time

output matrix is $\boldsymbol{H}_0$, according to (3), the inverse matrix is obtained as follows:

$$\boldsymbol{K}_0^{-1} = (\boldsymbol{H}_0\boldsymbol{H}_0^T + \frac{\boldsymbol{I}}{C})^{-1} \tag{4}$$

herefore, the output weight matrix $\boldsymbol{\beta}_0$ is:

$$\boldsymbol{\beta}_0 = \boldsymbol{H}_0\boldsymbol{K}_0^{-1}\boldsymbol{T}_0 \tag{5}$$

When γN_1 number of new samples arrives, $\gamma N_1 = 1, 2, 3\ldots$, the new hidden layer output matrix $\boldsymbol{H}_1$ becomes:

$$\boldsymbol{H}_1 = \begin{bmatrix} \boldsymbol{H}_0 \\ \gamma\boldsymbol{H}_1 \end{bmatrix} \tag{6}$$

where $\gamma\boldsymbol{H}_1$ is the γ number of output matrix. Similarly, according to (3), the inverse matrix $\boldsymbol{K}_1^{-1}$ can be constructed as:

$$\begin{aligned} \boldsymbol{K}_1^{-1} &= (\boldsymbol{H}_1\boldsymbol{H}_1^{-1} + \frac{\boldsymbol{I}}{C})^{-1} \\ &= \begin{bmatrix} \boldsymbol{H}_0\boldsymbol{H}_0^{-1} + \dfrac{\boldsymbol{I}}{C} & \boldsymbol{H}_0\gamma\boldsymbol{H}_1^T \\ \gamma\boldsymbol{H}_1\boldsymbol{H}_0^T & \gamma\boldsymbol{H}_1\gamma\boldsymbol{H}_1^T + \dfrac{\boldsymbol{I}}{C} \end{bmatrix}^{-1} \end{aligned} \tag{7}$$

Therefore, the output weight matrix can be expressed as $\boldsymbol{\beta}_1 = \boldsymbol{H}_1\boldsymbol{K}_1^{-1}\boldsymbol{T}_1$. With the successive arrival of block data, the update of $\boldsymbol{K}$ and $\boldsymbol{\beta}$ can be implemented in Algorithm 1.

With the continuous learning, the dimension of $\boldsymbol{K}$ becomes larger and larger. When $n > L$, the speed advantage brought by $(\boldsymbol{H}\boldsymbol{H}^T + \frac{\boldsymbol{I}}{C})^{-1}$ disappears. Therefore, the update scheme is changed to $(\boldsymbol{H}^T\boldsymbol{H} + \frac{\boldsymbol{I}}{C})^{-1}$. Assuming that $n \geq L$ in step m, $\boldsymbol{K}_{m+1}^{-1}$ can be expressed as:

$$\boldsymbol{K}_{m+1}^{-1} = (\boldsymbol{H}_{m+1}^T\boldsymbol{H}_{m+1} + \frac{\boldsymbol{I}}{C})^{-1} \tag{8}$$

where:

$$\boldsymbol{H}_{m+1} = \begin{bmatrix} \boldsymbol{H}_m \\ \gamma\boldsymbol{H}_{m+1} \end{bmatrix} \tag{9}$$

When new data is input in Step $i+1$ and $i > m$, then $\boldsymbol{H}_{i+1}$ is:

$$\boldsymbol{H}_{i+1} = \begin{bmatrix} \boldsymbol{H}_i \\ \gamma\boldsymbol{H}_{i+1} \end{bmatrix} \tag{10}$$

Algorithm 1 IRELM Update Formula When $n < L$

1. Input the first batch of samples;
2. $\boldsymbol{K}_0^{-1} = (\boldsymbol{H}_0\boldsymbol{H}_0^T + \frac{\boldsymbol{I}}{C})^{-1}$;$\boldsymbol{\beta}_0 = \boldsymbol{H}_0\boldsymbol{K}_0^{-1}\boldsymbol{T}_0$ (if necessary);
3. Assuming that $n \geq L$ in step

for $i = 1 : m-1$ **do**

$$\boldsymbol{S}_{i+1} = \gamma\boldsymbol{H}_{i+1}\gamma\boldsymbol{H}_{i+1}^T + \frac{\boldsymbol{I}}{C} - \gamma\boldsymbol{H}_{i+1}\boldsymbol{H}_i^T\boldsymbol{K}_i^{-1}\boldsymbol{H}_i\gamma\boldsymbol{H}_{i+1}^T;$$

$$\boldsymbol{A} = \boldsymbol{K}_i^{-1} + \boldsymbol{K}_i^{-1}\boldsymbol{H}_i\gamma\boldsymbol{H}_{i+1}^T\boldsymbol{S}_{i+1}^{-1}\gamma\boldsymbol{H}_{i+1}\boldsymbol{H}_i^T\boldsymbol{K}_i^{-1};$$

$$\boldsymbol{B} = -\boldsymbol{K}_i^{-1}\boldsymbol{H}_i\gamma\boldsymbol{H}_{i+1}^T\boldsymbol{S}_{i+1}^{-1};$$

$$\boldsymbol{C} = -\boldsymbol{S}_{i+1}^{-1}\gamma\boldsymbol{H}_i\boldsymbol{H}_i^T\boldsymbol{K}_i^{-1};$$

$$\boldsymbol{D} = \boldsymbol{S}_{i+1}^{-1};$$

$$\boldsymbol{K}_{i+1}^{-1} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix};$$

$$\boldsymbol{\beta}_{i+1} = \boldsymbol{H}_{i+1}\boldsymbol{K}_{i+1}^{-1}\boldsymbol{T}_{i+1} \quad \text{(if necessary);}$$

end

Therefore, the inverse matrix $\boldsymbol{K}_{i+1}^{-1}$ can be calculated as follows:

$$\begin{aligned} &\boldsymbol{K}_{i+1}^{-1} \\ &= \left[\boldsymbol{K}_i + \gamma\boldsymbol{H}_{i+1}^T\boldsymbol{I}\gamma\boldsymbol{H}_{i+1}\right]^{-1} \\ &= \boldsymbol{K}_i^{-1} - \boldsymbol{K}_i^{-1}\gamma\boldsymbol{H}_{i+1}^T(\boldsymbol{I} + \gamma\boldsymbol{H}_{i+1}\boldsymbol{K}_i^{-1}\gamma\boldsymbol{H}_{i+1}^T)\gamma\boldsymbol{H}_{i+1}\boldsymbol{K}_i^{-1} \end{aligned} \tag{11}$$

When $n > L$, the sequential updating formula of IRELM can be implemented according to Algorithm 2.

According to Algorithm 1 and Algorithm 2, the advantages of IRELM are as follows:

1. Dimension reduction of inverse matrix. In RELM, the process that consumes the most computing power is the calculation of $(\boldsymbol{H}^T\boldsymbol{H})^{-1}$ and $\boldsymbol{K}_{i+1}^{-1}$. In IRELM's updating scheme, the dimension of inverse matrix $\boldsymbol{S}_{i+1}^{-1}$ is only $\gamma N_i \times \gamma N_i$, which is far less than $L \times L$, avoiding the most complex part of RELM.

Algorithm 2 IRELM Update Formula When $n > L$

1. Connect Algorithm1;
2. Assuming that $n \geq L$ in step;
3. $\boldsymbol{H}_{m+1} = \left[\boldsymbol{H}_m \ \gamma \boldsymbol{H}_{m+1}\right]$;
4. $\boldsymbol{K}_{m+1}^{-1} = (\boldsymbol{H}_{m+1}^T \boldsymbol{H}_{m+1} + \frac{\boldsymbol{I}}{C})^{-1}$;
5. $i = m + 1$;
6. **while** No termination **do**

$$\boldsymbol{H}_{i+1} = \left[\boldsymbol{H}_i \quad \gamma \boldsymbol{H}_{i+1}\right];$$
$$\boldsymbol{K}_{i+1}^{-1} = \boldsymbol{K}_i^{-1} - \boldsymbol{K}_i^{-1}\gamma \boldsymbol{H}_{i+1}^T(\boldsymbol{I} + \gamma \boldsymbol{H}_{i+1}\boldsymbol{K}_i^{-1}\gamma \boldsymbol{H}_{i+1}^T) \gamma \boldsymbol{H}_{i+1}\boldsymbol{K}_i^{-1};$$
$$\boldsymbol{\beta}_{i+1} = \boldsymbol{H}_{i+1}\boldsymbol{K}_{i+1}^{-1}\boldsymbol{T}_{i+1} \quad \text{(if necessary)};$$
$$i = i + 1;$$

end

2. Calculate the value of $\boldsymbol{\beta}$ if necessary. The calculation of $\boldsymbol{\beta}_{i+1}$ value does not depend on the previous value, so it needs to be updated when necessary, and does not need to be updated when unnecessary, which reduces the calculation cost.
3. The batch learning is replaced by sequential learning, which is more suitable for the actual situation, not only maintains the generalization ability of RELM, but also maintains the efficiency of the algorithm.

B. DESIGN PSO ALGORITHM

Particle swarm optimization [30] is one of the classic heuristic algorithms, and its main idea comes from the predatory behavior of a flock of birds. The core is to make use of the information sharing of individuals in the group so that the movement of the whole group has an evolutionary process from disorder to order in the problem solving space, so as to get the optimal solution of the problem. Suppose that there is a group of particles in a D-dimensional search space, and each particle has an initial velocity v_k, initial position s_k and fitness value g_k. In each iteration, each particle constantly updates its own velocity and position. At the same time, the fitness value is used to determine the optimal position p_k of the updated individual and the optimal position p_g of the population. Assuming that the optimal position is the initial position of particles in the first iteration, the update formula of the velocity and position of particles in the population are as follows:

$$v_{kd}(u+1) = zv_{kd}(u) + c_1 r_1(p_{kd}(u) - s_{kd}(u)) + c_2 r_2(p_{gd}(u) - s_{kd}(u)) \tag{12}$$
$$s_{kd}(u+1) = s_{kd}(u) + v_{kd}(u+1) \tag{13}$$

where k represents the kth particle in the population; u represents the current iteration times; z is the inertia factor, whose value is nonnegative. When z is large, the global search ability is strong; when z is small, the global search ability becomes weak; c_1 and c_2 are the individual learning factors and social learning factors of particles, whose values are nonnegative

Algorithm 3 The Framework of IPSO

1. **for** each particle k **do**
   Initialize velocity v_{kd} and position s_{kd} for particle k;
   Evaluate particle k and set $p_{kd} = s_{kd}$;
   end
   $p_{gd} = \min\{p_{kd}\}$;
2. **while** not stop
   for $k = 1 : a$
   Update the velocity and position of particle k using (14) and (13);
   Evaluate particle k;
   if fit (s_{kd}) < fit (p_{kd})
   $p_{kd} = s_{kd}$;
   if fit (p_{kd}) < fit (p_{gd})
   $p_{gd} = p_{kd}$;
   end
3. Select the top S elite particle and implement adaptive mutation for them by (16);
4. Get the current new position;
5. If the new position is better than before, update it;
   end

constants; r_1 and r_2 are independent random numbers in the range of [0,1].

In order to make the parameters of PSO adjust adaptively with the number of iterations, many scholars have proposed a variety of autonomous particle swarm optimization algorithms, which effectively balance the global and local search ability of particles, but still cannot solve the defect that particles are easy to fall into local optimum. Therefore, this paper proposes a Cauchy-Gaussian mutation strategy to make the particles which fall into local optimum get rid of stagnation and continue to search. Secondly, according to Clerc's idea of shrinkage factor [31], only use a global contraction factor to replace other adaptive parameters. The improved particle velocity update formula and Cauchy-Gaussian mutation strategy are as follows:

$$v_{kd}(u+1) = K[v_{kd}(u) + c_1 r_1(p_{kd}(u) - s_{kd}(u)) + c_2 r_2(p_{gd}(u) - s_{kd}(u))] \tag{14}$$
$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|},$$
$$where\ \varphi = c_1 + c_2, \quad \varphi > 4 \tag{15}$$
$$x_{kd}(u) = s_{kd}(u) \times \left[1 + \lambda_1 cauchy(0, \sigma^2) + \lambda_2 Gauss(0, \sigma^2)\right], \tag{16}$$
$$\sigma = \begin{cases} 1, & f(s_{best}) < f(s_{kd}(u)) \\ \exp\left(\dfrac{f(s_{best}) - f(s_{kd})}{|f(s_{best})|}\right), & otherwise \end{cases} \tag{17}$$

where s_{best} is the optimal individual position; $x_{kd}(u)$ represents the position of the optimal individual after mutation;

σ^2 is the standard deviation; λ_1 and λ_2 are the dynamic parameters of adaptive adjustment. In the search process, λ_1 is larger in the initial stage, and then gradually decreases, so that the algorithm can explore in a larger range with a larger mutation step; λ_2 is increasing, which is conducive to the algorithm to search the near optimal solution.

IPSO pseudo code is as follows:

TABLE 2. Updating strategies.

Algorithm	Updating formula	
	C1	C2
PSO	2	2
MPSO	(-2.05/T) t+2.55	(1/T) t+1.25
TACPSO	0.5+2exp [-(4t/T)^2]	2.2-2exp [-(4t/T)^2]
AGPSO1		
Group1	(-2.05/T) t+2.55	(1/T) t+1.25
Group2	(-2.05/T) t+2.55	($2t^3$/T) +0.5
Group3	(-$2t^3$/T^3) +2.5	(1/T) t+1.25
Group4	(-$2t^3$/T^3) +2.5	($2t^3$/T^3) +0.5
AGPSO2		
Group1	2.5-(2log(t)/log(T))	(2log(t)/log(T)) +0.5
Group2	(-$2t^3$/T^3) +2.5	($2t^3$/T^3) +0.5
Group3	0.5+2exp [-(4t/T)^2]	2.2-2exp [-(4t/T)^2]
Group4	2.5+2(t/T)^2-2(2t/T)	0.5-2(t/T)^2-2(2t/T)
AGPSO3		
Group1	1.95-2$t^{1/3}$/$T^{1/3}$	2$t^{1/3}$/$T^{1/3}$+0.05
Group2	(-$2t^3$/T^3) +2.5	($2t^3$/T^3) +0.5
Group3	1.95-2$t^{1/3}$/$T^{1/3}$	($2t^3$/T^3) +0.5
Group4	(-$2t^3$/T^3) +2.5	2$t^{1/3}$/$T^{1/3}$+0.05

C. INTRUSION DETECTION ALGORITHM BASED ON IPSO-IRELM

Based on the above analysis and derivation, we introduce IPSO into IRELM algorithm and establish an intrusion detection model based on IPSO-IRELM. The role of IPSO is to find a group of particle swarm positions with the best fitness function, and assign them to the optimal initial weight and deviation of IRELM after the iteration to establish an intrusion detection model. The model description is shown in the Fig.1:

IV. EXPERIMENTAL RESULTS

A. PARAMETER SETTING

Experiment 1: The parameters of classic PSO and its derivative algorithms (MPSO [32], TACPSO [33], AGPSO [34]) compared with IPSO are as follows:

Among them, MPSO is to incorporate an asymmetric time-varying acceleration coefficients adjusting strategy, which maintains the balance between global and local search with the great advantages of convergence property and robustness compared with basic PSO algorithm; In TACPSO, random velocities are added to reinitialize the velocities of particles in order to avoid searching for particles with zero

TABLE 3. Benchmark functions.

Function	D	range	optimal
$F_1(x)=\sum_{i=1}^{n}x_i^2$	30 100	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30 100	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{n}x_j\right)^2$	30 100	[-100,100]	0
$F_4(x)=\max\{\lvert x_i\rvert,1\le i\le n\}$	30 100	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$	30 100	[-30,30]	0
$F_6(x)=\sum_{i=1}^{n}(\lvert x_i+0.5\rvert)^2$	30 100	[-100,100]	0
$F_7(x)=\sum_{i=1}^{n}ix_i^4+random[0,1]$	30 100	[-1.28,1.28]	0
$F_8(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	30 100	[-500,500]	-418.9829n
$F_9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30 100	[-5.12,5.12]	0
$F_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30 100	[-32,32]	0
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30 100	[-600,600]	0
$F_{12}(x)=\frac{\pi}{n}\left\{10\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]\right\}+(y_n-1)^2$	30 100	[-50,50]	0
$F_{13}(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+10\sin^2(3\pi x_i+1)\right]+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\}+\sum_{i=1}^{n}u(x_i,5,100,4)$	30 100	[-50,50]	0

velocity prematurely. Also, to enhance early exploration and later exploitation, exponential time-varying acceleration coefficients are introduced, and the algorithm has a better probability of finding the global optimum and the average optimum than other algorithms; The main idea of the AGPSO algorithm is inspired by the diversity of individuals in bird or insect flocks, using different functions with different slopes, curvatures and intercept points to tune the social and cognitive parameters of the particle swarm algorithm in order to endow the particles with behaviors different from those of the natural population, further alleviating the problem of getting stuck in local minima and slow convergence of high-dimensional problems. The MPSO, TACPSO and AGPSO were chosen for comparison because these three improvement methods are more classical and more effective.

When doing the experiment, the number of particles is set to 30; the number of iterations is 1000. Take the average of 50 runs of the experiment.

Experiment 2-4: The parameters of the IRELM neural network model are: 41—39—1, and the number of iterations is 100. The model parameters of IRELM are determined by the two-classification experiment of GA-IRELM. First, set the number of hidden layer nodes to 10, 20, 30, 40, and 50 respectively. The experimental results show that the model with 40 hidden layer units has the best detection accuracy. Then set the number of hidden layer nodes to 36, 37, 38, 39, 40, 41, 42. The results show that the model with 39 hidden layer units has the best detection accuracy. When the number of hidden layer nodes is increased from 39 to 42, the accuracy of intrusion detection decreases. Keeping the hidden layer nodes unchanged, increasing the number of iterations will only cause overfitting and fluctuate the detection efficiency of the model.

B. IRELM BASED ON SEQUENTIAL LEARNING

Experiment1: We choose 13 benchmark functions for testing, among which F1-F7 are single-peak benchmark functions, and F8-F13 are multimodal benchmark functions [35].

Experiment2: The UCI data sets [36] are as follows:

TABLE 4. UCI datasets details.

Dataset	Dimension	Category	Total sample	Test sample
Iris	4	3	150	30
wine	13	3	178	89

Experiment3: The NSL-KDD data set [37] is as follows:

TABLE 5. NSL-KDD classification situation.

Dataset	Normal	DoS	U2R	R2L	Probe
Training set	13449	9234	11	209	2289
Test set	9711	7458	200	2754	2421

Experiment4: The UNSW-NB15 data set [38] is as follows:

TABLE 6. UNSW-NB15 classification situation.

Dataset	All	Normal	Analysis	Backdoor
Training set	175341	56000	2000	1746
Test set	82332	37000	7458	583
Dataset	Dos	Exploits	Fuzzers	Generic
Training set	12264	33393	18184	40000
Test set	4089	11132	6062	18871
Dataset	Reconnaissance		shellcode	Worms
Training set	10491		1133	130
Test set	3496		378	44

C. IPSO EXPERIMENT RESULTS

In order to verify that IPSO has a better optimization effect, we use the 13 benchmark functions in Table. 3 to do experiments, and verify whether IPSO is applicable to any dimension from the perspective of low-dimensional and high-dimensional. Fig.2 shows the images of thirteen benchmark functions and the convergence curves of the algorithms in 30 dimensions. Table. 7 and Table. 8 respectively show the best optimization result, average optimization result and standard deviation of optimization result of unimodal function and multimodal function in low dimension. Table. 9 and Table. 10 respectively show the best optimization result, average optimization result and standard deviation of optimization result of unimodal function and multimodal function in high dimension respectively. The best result represents the optimization ability of the algorithm, and the average result and standard deviation of the optimization results reflect the stability and robustness of the algorithm. For attention, the best values in Table. 7-10 are marked in bold.

According to Table. 7-10, when the dimension of the benchmark function is 30, first, IPSO obtains 11 minimum values of 13 benchmark functions except F8 and F9. Secondly, 8 average optimal values are obtained in 13 benchmark functions. At the same time, IPSO's standard deviation of the optimal results is also significantly better than other algorithms. When the dimension of benchmark function is 300, IPSO obtains 6 minimum values, 3 average optimal values and 2 standard deviations of optimal results, while PSO obtains 5 minimum values, 8 average optimal values and 7 standard deviations of optimal results. This shows that through the comparison of Table. 7 and Table. 8, IPSO can obtain better optimization results in low-dimensional benchmark functions, but the effect on multimodal functions is not as good as that of unimodal functions; through the comparison of Table. 9 and Table. 10, in high-dimensional benchmark functions, other improved PSO algorithms are inferior to the original PSO, and in the improved PSO, the comprehensive effect of IPSO is relatively the best; through the comparison of Table. 7 and Table. 9, the comparison of Table. 8 and Table. 10, we can conclude that the higher the dimensionality of the benchmark function, the worse the optimization effect. Therefore, the original PSO has a better effect on the

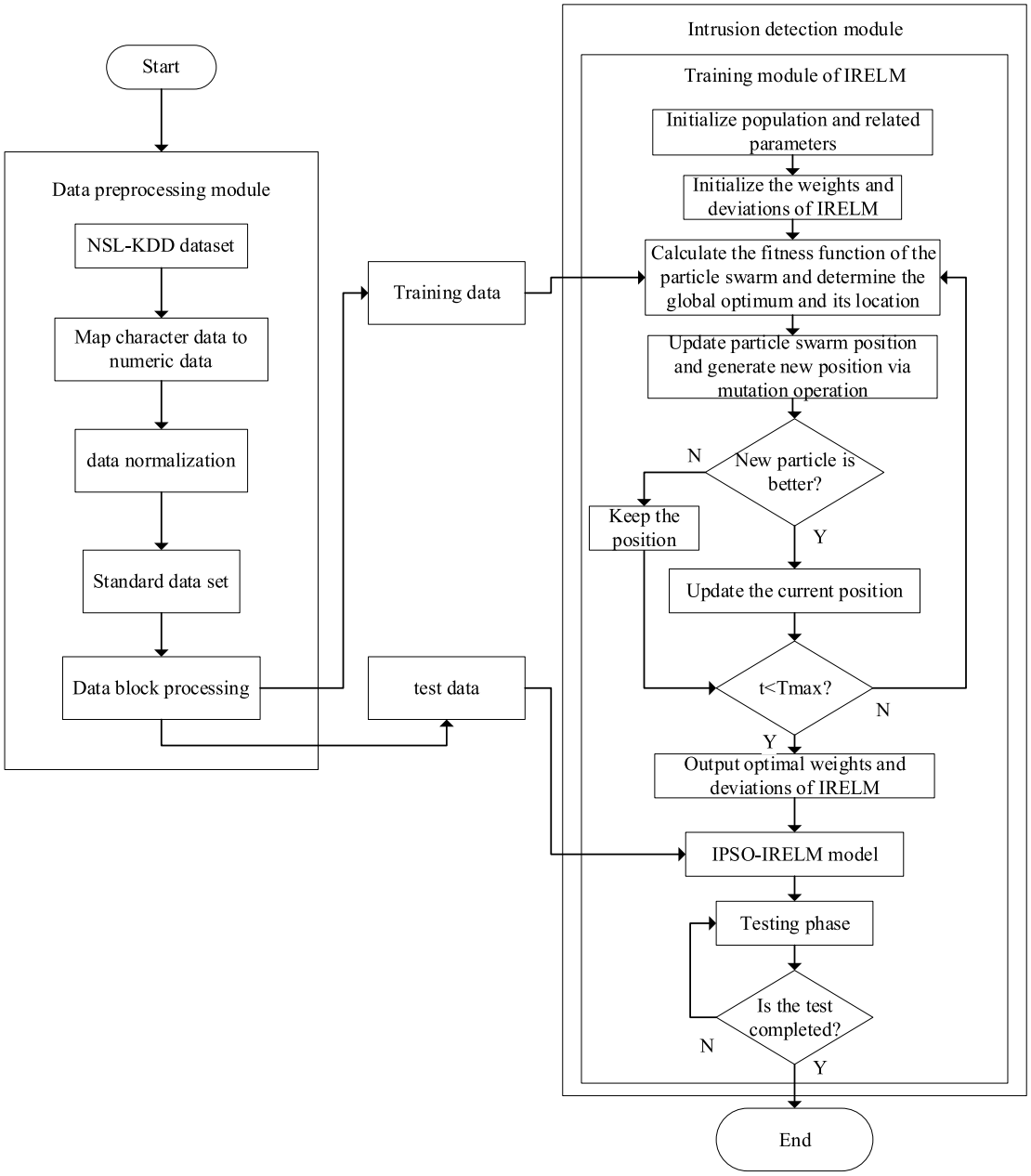

FIGURE 1. IPSO-IRELM algorithm intrusion detection framework.

high-dimensional benchmark function, and other improved PSOs are slightly inferior.

D. UCI DATA SET EXPERIMENTAL RESULTS

In order to verify the effectiveness of the proposed algorithm on the real classification data set, IPSO-IRELM was compared with PSO-IRELM, GA-IRELM, IRELM and RELM on the UCI data set. The performance evaluation indexes of each algorithm are listed in Table. 11-13, and the detection results of each algorithm are given in Fig.3. It can be seen intuitively from Fig.3 that whether it is the Iris data set or the Wine data set, the predicted value of IPSO-IRELM coincides perfectly with the true value, and the classification accuracy rate reaches 100%. It can be seen from Table. 11-13 that the performance evaluation indicators of other comparison algorithms are better, but not as good as IPSO-IRELM. This is because the two selected data sets are balanced data sets, so the detection results are better. IPSO-IRELM has the best performance evaluation index, which indicates that IPSO has better optimization ability than PSO and GA algorithm, and the necessity of introducing IPSO into IRELM; it also verifies that IPSO-IRELM algorithm has better classification performance. Therefore, it is further used in network intrusion detection to verify the feasibility of IPSO-IRELM.

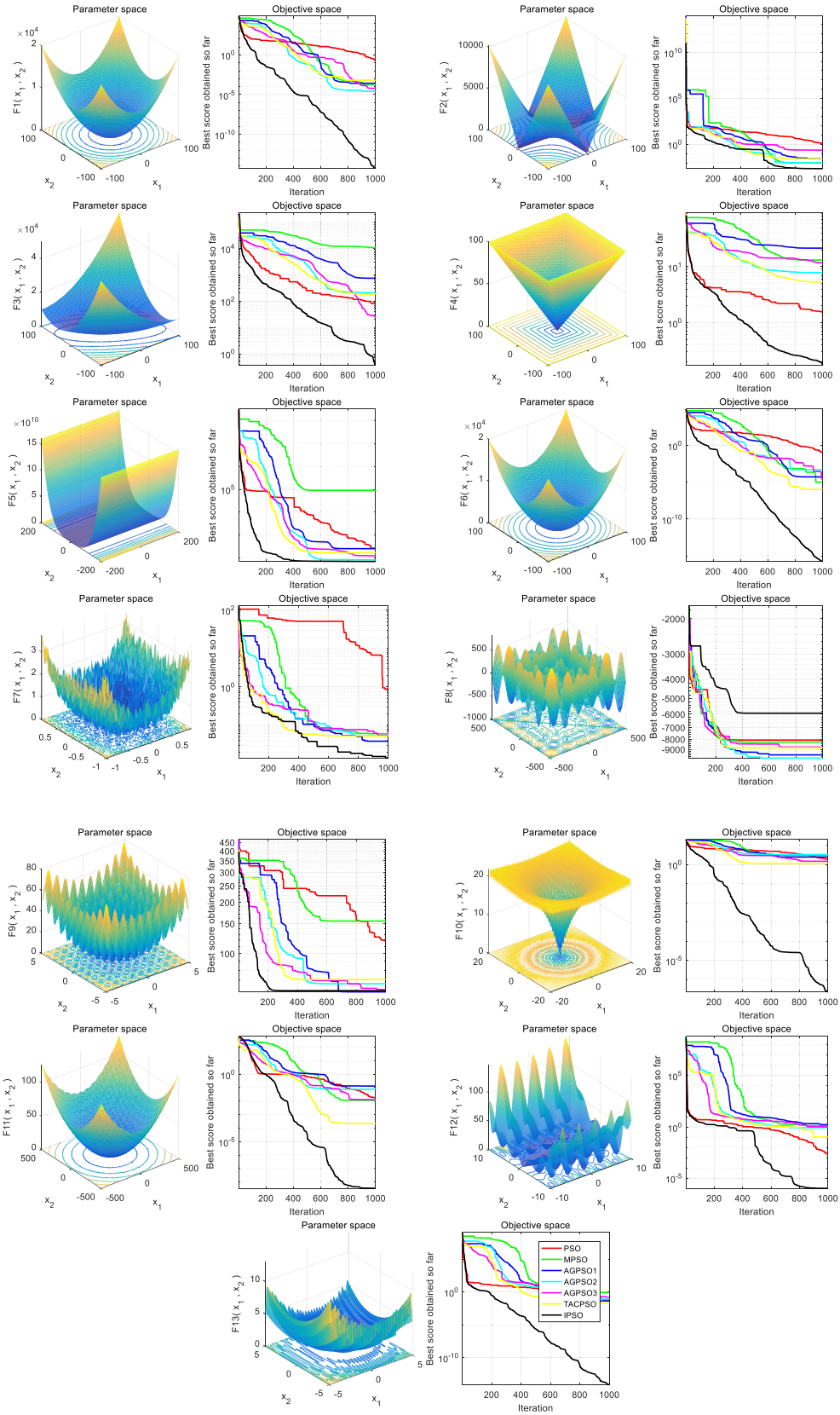

FIGURE 2. Convergence curve.

TABLE 7. The experimental results of single peak when dim = 30.

F		PSO	MPSO	AGPSO1	AGPSO2	AGPSO3	TACPSO	IPSO
F1	MINBEST	5.9673e-02	3.3629e-07	2.7168e-06	1.5206e-07	1.9610e-07	1.2425e-07	**1.4878e-17**
	MEANBEST	3.1037E-01	3.5051E-03	6.5684E-03	5.3615E-03	4.5723E-05	2.4345E-04	**4.5503E-12**
	STDBEST	1.9301E-01	9.0293E-03	2.5917E-02	2.5961E-02	8.0874E-05	8.8848E-04	**2.3702E-11**
F2	MINBEST	4.1157e-01	2.0349e-02	9.6988e-04	1.9317e-03	9.4268e-04	1.1852e-03	**2.1913e-06**
	MEANBEST	9.3152E-01	2.4266E+01	1.8893E+00	1.8176E+00	2.2387E+00	9.2332E-01	**1.4186E-02**
	STDBEST	3.6868E-01	1.4143E+01	3.8527E+00	3.7209E+00	4.6466E+00	2.7503E+00	**3.4066E-02**
F3	MINBEST	2.6674e+01	3.4040e+02	9.8346e+01	3.5596e+01	2.9571e+01	2.3462e+01	**4.2874e-01**
	MEANBEST	7.7519E+01	1.1952E+04	1.6198E+03	1.1588E+03	9.8490E+02	5.3551E+02	**3.2033E+00**
	STDBEST	2.4524E+01	8.7722E+03	2.4439E+03	1.8380E+03	1.8151E+03	1.2363E+03	**2.9167E+00**
F4	MINBEST	8.1923e-01	3.5182e+00	7.5414e+00	4.8334e+00	6.8907e+00	1.6553e+00	**1.1040e-01**
	MEANBEST	1.5509E+00	1.1428E+01	1.4991E+01	1.0965E+01	1.3808E+01	5.6322E+00	**6.6699E-01**
	STDBEST	**2.5775E-01**	4.1245E+00	3.4921E+00	2.9325E+00	3.7780E+00	2.3329E+00	4.4094E-01
F5	MINBEST	6.1565e+01	2.2089e+01	8.0851e+00	7.1807e+00	1.1890e+01	1.5799e+01	**2.2429e-01**
	MEANBEST	3.1694E+02	1.8652E+04	2.0451E+03	2.1365E+03	3.8745E+03	1.5626E+02	**4.4016E+01**
	STDBEST	2.7924E+02	3.6082E+04	1.2711E+04	1.2711E+04	1.7793E+04	4.3157E+02	**5.5992E+01**
F6	MINBEST	5.5816e-02	4.6986e-07	3.7805e-06	7.1311e-07	2.6101e-07	6.1234e-09	**5.1209e-19**
	MEANBEST	2.5130E-02	9.3297E-04	1.8982E-03	1.6529E-03	1.4774E-04	5.6502E-04	**2.3863E-12**
	STDBEST	1.2544E-01	3.3092E-03	4.9435E-03	5.3435E-03	3.7819E-04	1.4996E-03	**1.5603E-11**
F7	MINBEST	5.5612e-01	1.8962e-02	1.7902e-02	2.2176e-02	3.4248e-02	1.3020e-02	**6.5945e-03**
	MEANBEST	3.4397E+00	1.4997E+00	5.0724E-02	4.5951E-02	6.5855E-02	5.2682E-02	**3.4418E-02**
	STDBEST	6.9762E+00	3.3049E+00	2.0420E-02	**1.8327E-02**	1.8728E-02	2.9696E-02	3.0003E-02

TABLE 8. The experimental results of multimodal when dim = 30.

F		PSO	MPSO	AGPSO1	AGPSO2	AGPSO3	TACPSO	IPSO
F8	MINBEST	-9.095e+03	-1.019e+04	-1.079e+04	-1.069e+04	**-1.126e+04**	-1.049e+04	-8.265e+03
	MEANBEST	-6.215E+03	-8.784E+03	-9.610E+03	-9.685E+03	**-9.878E+03**	-9.088E+03	-5.872E+03
	STDBEST	1.316E+03	6.269E+02	**6.160E+02**	6.474E+02	6.761E+02	6.404E+02	1.365E+03
F9	MINBEST	6.5020e+01	5.1738e+01	2.7864e+01	**2.6876e+01**	2.9849e+01	3.5818e+01	2.7859e+01
	MEANBEST	1.1235E+02	**1.2168E+01**	5.7324E+01	6.8128E+01	6.1476E+01	7.1654E+01	5.9339E+01
	STDBEST	2.7929E+01	2.7533E+01	1.8443E+01	1.9848E+01	1.7648E+01	1.9009E+01	**1.6107E+01**
F10	MINBEST	2.0240e-01	4.7901e-03	1.3404e+00	3.3400e-03	1.5019e+00	9.6066e-04	**8.1559e-09**
	MEANBEST	**9.7292E-01**	2.0708E+00	2.3792E+00	2.4063E+00	2.6227E+00	1.9445E+00	1.8710E+00
	STDBEST	**5.3318E-01**	1.9522E+00	5.7264E-01	9.0893E-01	6.2229E-01	7.9240E-01	6.8567E-01
F11	MINBEST	3.0716e-03	1.3359e-06	5.4697e-06	2.1643e-06	3.1092e-07	6.1879e-07	**0**
	MEANBEST	2.2704E-02	1.8282E+00	5.5472E-02	3.5947E-02	3.9066E-02	**2.0706E-02**	5.3595E-02
	STDBEST	**1.3059E-02**	1.2725E+01	4.6587E-02	3.5280E-02	3.7912E-02	2.0636E-02	7.8357E-02
F12	MINBEST	4.4403e-04	6.6592e-05	1.1129e-01	1.4342e-06	3.8373e-04	1.0410e-05	**2.1840e-16**
	MEANBEST	**2.4507E-03**	1.1071E+00	2.0172E+00	9.6968E-01	1.3017E+00	6.1583E-01	2.4732E-01
	STDBEST	**1.9265E-03**	1.1897E+00	1.4934E+00	9.6465E-01	1.3597E+00	7.4619E-01	4.8974E-01
F13	MINBEST	2.0406e-02	4.2910e-06	3.3343e-03	3.5921e-04	7.6683e-04	4.3173e-05	**1.3899e-17**
	MEANBEST	1.0236E-01	1.1215E+00	2.3573E+00	9.7076E-01	6.7627E-01	7.6972E-01	**7.1812E-02**
	STDBEST	**7.5228E-02**	2.2498E+00	2.9466E+00	1.6146E+00	1.1258E+00	1.8656E+00	1.6483E-01

TABLE 9. The experimental results of single peak when dim = 300.

F		PSO	MPSO	AGPSO1	AGPSO2	AGPSO3	TACPSO	IPSO
F1	MINBEST	**1.714e+03**	2.333e+05	8.130e+04	6.299e+04	4.723e+04	6.937e+04	2.632e+03
	MEANBEST	**2.042E+03**	2.835E+05	1.057E+05	9.556E+04	7.217E+04	9.422E+04	4.123E+03
	STDBEST	**1.671E+02**	2.594E+04	1.159E+04	1.413E+04	1.349E+04	9.092E+03	8.877E+02
F2	MINBEST	1.692e+06	9.991e+02	7.025e+02	6.112e+02	7.267e+02	4.945e+02	**3.732e+02**
	MEANBEST	1.111E+63	5.771E+07	9.025E+02	7.989E+02	9.289E+02	6.132E+02	**4.828E+02**
	STDBEST	7.857E+63	4.081E+08	8.870E+01	9.795E+01	**5.763E+01**	8.108E+01	7.461E+01
F3	MINBEST	1.249e+05	9.746e+05	5.064e+05	3.472e+05	4.178e+05	3.025e+05	**6.766e+04**
	MEANBEST	2.020E+05	1.641E+06	9.509E+05	7.379E+05	6.577E+05	6.286E+05	**1.393E+05**
	STDBEST	4.805E+04	3.291E+05	1.912E+05	2.177E+05	1.355E+05	1.485E+05	**3.325E+04**
F4	MINBEST	2.087e+01	8.327e+01	6.357e+01	6.347e+01	6.552e+01	5.953e+01	**2.053e+01**
	MEANBEST	**2.334E+01**	9.714E+01	6.891E+01	6.945E+01	7.101E+01	6.489E+01	2.345E+01
	STDBEST	**1.375E+00**	3.115E+00	2.502E+00	3.151E+00	2.767E+00	2.373E+00	1.604E+00
F5	MINBEST	5.902e+06	5.045e+08	3.771e+07	3.426e+07	2.618e+07	5.552e+07	**3.579e+06**
	MEANBEST	**7.940E+06**	7.530E+08	7.852E+07	8.174E+07	4.723E+07	9.570E+07	8.397E+06
	STDBEST	**1.048E+06**	1.384E+08	2.071E+07	2.750E+07	1.375E+07	1.781E+07	2.629E+06
F6	MINBEST	**1.655e+03**	2.182e+05	7.826e+04	6.728e+04	4.131e+04	6.827e+04	2.368e+03
	MEANBEST	**2.006E+03**	2.827E+05	1.024E+05	9.661E+04	7.704E+04	9.724E+04	4.287E+03
	STDBEST	**1.685E+02**	2.633E+04	1.270E+04	1.425E+04	1.483E+04	1.233E+04	1.006E+03
F7	MINBEST	1.680e+04	2.706e+03	3.896e+02	6.849e+02	2.854e+02	**2.580e+02**	1.621e+04
	MEANBEST	1.894E+04	4.418E+03	9.068E+02	1.502E+03	**7.035E+02**	8.049E+02	1.909E+04
	STDBEST	8.664E+02	8.732E+02	**2.899E+02**	4.180E+02	3.495E+02	3.580E+02	1.168E+03

E. NSL-KDD DATA SET 2-ELEMENT CLASSIFICATION EXPERIMENT RESULTS

We merged the four types of attacks in the NSL-KDD dataset into Abnormal and denoted as 2 and Normal as 1. The experiment changed from a multi-classification problem to a two-element classification problem. The experimental results of each algorithm are shown in Table. 14 and Table. 15. Fig.4 is a comparison diagram of the confusion matrix

TABLE 10. The experimental results of multimodal when dim = 300.

F		PSO	MPSO	AGPSO1	AGPSO2	AGPSO3	TACPSO	IPSO
F8	MINBEST	**-7.747e+04**	-5.741e+04	-6.335e+04	-5.968e+04	-6.460e+04	-6.058e+04	-6.305e+04
	MEANBEST	**-6.079E+04**	-4.715E+04	-5.691E+04	-5.384E+04	-5.627E+04	-5.474E+04	-4.174E+04
	STDBEST	**1.453E+04**	4.378E+03	3.186E+03	3.036E+03	3.208E+03	2.840E+03	1.613E+04
F9	MINBEST	3.929e+03	2.602e+03	1.770e+03	1.737e+03	**1.557e+03**	1.848e+03	1.830e+03
	MEANBEST	4.381E+03	2.936E+03	2.072E+03	2.011E+03	**1.886E+03**	2.040E+03	2.123E+03
	STDBEST	2.317E+02	1.358E+02	1.218E+02	1.016E+02	1.281E+02	**8.993E+01**	1.443E+02
F10	MINBEST	**9.641e+00**	1.965e+01	1.766e+01	1.738e+01	1.796e+01	1.625e+01	1.187e+01
	MEANBEST	**1.011E+01**	1.987E+01	1.828E+01	1.814E+01	1.842E+01	1.717E+01	1.306E+01
	STDBEST	2.054E-01	**8.085E-02**	2.500E-01	3.416E-01	1.831E-01	4.230E-01	8.181E-01
F11	MINBEST	**1.465e+00**	2.010e+03	7.955e+02	6.149e+02	4.167e+02	6.714e+02	4.210e+00
	MEANBEST	**1.572E+00**	2.576E+03	9.548E+02	8.584E+02	6.494E+02	8.598E+02	9.870E+00
	STDBEST	**4.824E-02**	2.226E+02	1.149E+02	1.529E+02	1.046E+02	9.997E+01	4.628E+00
F12	MINBEST	1.330e+03	5.900e+08	8.855e+06	1.384e+07	5.626e+06	1.757e+07	**2.260e+01**
	MEANBEST	1.612E+04	1.293E+09	2.944E+07	3.707E+07	1.704E+07	4.468E+07	**4.559E+02**
	STDBEST	1.254E+04	4.168E+08	1.477E+07	2.247E+07	1.003E+07	1.528E+07	**1.299E+03**
F13	MINBEST	1.996e+05	1.786e+09	8.407e+07	8.575e+07	3.000e+07	9.071e+07	**3.008e+04**
	MEANBEST	**4.483E+05**	2.866E+09	1.605E+08	1.777E+08	9.507E+07	2.165E+08	5.285E+05
	STDBEST	**1.378E+05**	6.145E+08	4.900E+07	9.348E+07	3.568E+07	5.733E+07	4.675E+05

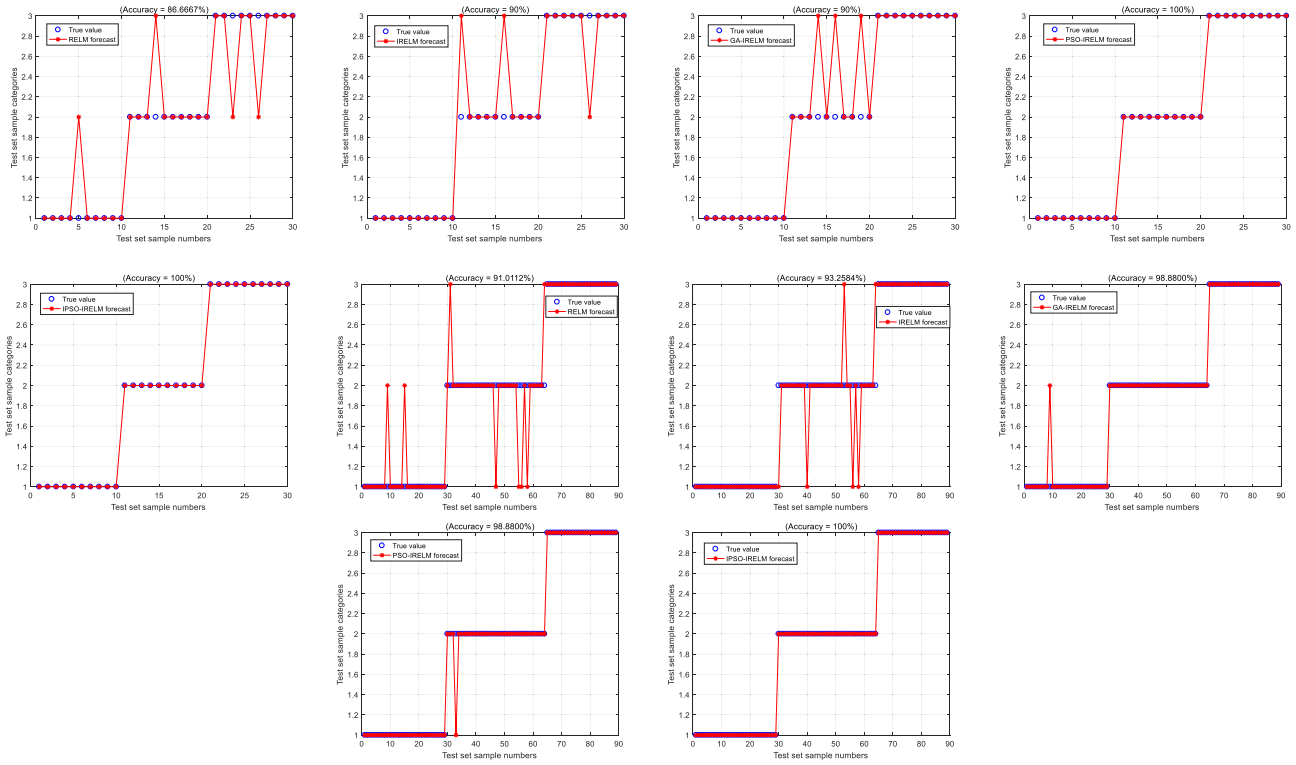

FIGURE 3. The detection results of each algorithm on UCI datasets.

TABLE 11. Accuracy of each algorithm on UCI dataset (%).

Dataset	Algorithm	RELM	IRELM	GA-IRELM	PSO-IRELM	IPSO-IRELM
Iris	Accuracy	86.6667	90.0000	90.0000	100.000	100.000
Wine	Accuracy	91.0112	93.2584	98.8800	98.8800	100.000

and ROC curve of the binary classification results. From Table. 14, compared with other algorithms, IPSO-IRELM has the highest accuracy rate of up to 91.13%, but the NSL-KDD data set is an unbalanced data set. Therefore, in addition to the accuracy rate, the precision, the true positive rate (TPR), the false positive rate (FPR), the F-score, and the area under curve (AUC) are used to evaluate the classification [39]. It can be seen from Table. 15 that for category 1, the TPR

TABLE 12. Performance evaluation index of each algorithm on Iris dataset.

Algorithm	Category	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
RELM	1	100.0	90.00	0.000	94.74	0.9500
	2	75.00	90.00	15.00	81.82	0.8750
	3	88.89	80.00	5.000	84.21	0.8750
IRELM	1	100.0	100.0	0.000	100.0	1.0000
	2	88.89	80.00	5.000	84.21	0.8750
	3	81.82	90.00	10.00	85.71	0.9000
GA-IRELM	1	100.0	100.0	0.000	100.0	1.0000
	2	100.0	70.00	0.000	82.35	0.8500
	3	76.92	100.0	15.00	86.96	0.9250
PSO-IRLEM	1	100.0	100.0	0.000	100.0	1.0000
	2	100.0	100.0	0.000	100.0	1.0000
	3	100.0	100.0	0.000	100.0	1.0000
IPSO-IRELM	1	100.0	100.0	0.000	100.0	1.0000
	2	100.0	100.0	0.000	100.0	1.0000
	3	100.0	100.0	0.000	100.0	1.0000

TABLE 13. Performance evaluation index of each algorithm on wine dataset.

Algorithm	Category	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
RELM	1	87.10	93.10	6.670	90.00	0.9322
	2	93.55	82.86	3.700	87.88	0.8958
	3	92.59	100.0	3.130	96.15	0.9844
IRELM	1	87.88	100.0	6.670	93.55	0.9667
	2	100.0	82.86	0.000	90.63	0.9143
	3	92.59	100.0	3.130	96.15	0.9844
GA-IRELM	1	100.0	96.55	0.000	98.25	0.9828
	2	97.22	100.0	1.850	98.59	0.9907
	3	100.0	100.0	0.000	100.0	1.0000
PSO-IRLEM	1	96.67	100.0	1.670	98.31	0.9917
	2	100.0	97.14	0.000	98.55	0.9857
	3	100.0	100.0	0.000	100.0	1.0000
IPSO-IRELM	1	100.0	100.0	0.000	100.0	1.0000
	2	100.0	100.0	0.000	100.0	1.0000
	3	100.0	100.0	0.000	100.0	1.0000

TABLE 14. Accuracy of each algorithm on NSL-KDD dataset (%).

Algorithm	SVM	ELM	RELM	IRELM	GA-IRELM	PSO-IRELM	IPSO-IRELM
Accuracy	76.2331	76.1489	79.3249	81.0814	83.7200	86.9200	91.1300

TABLE 15. Performance evaluation index of each algorithm on NSL-KDD dataset.

Algorithm	Category	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	1	64.89	97.66	39.98	77.97	0.7884
	2	97.14	60.02	2.340	74.19	0.7884
ELM	1	64.84	97.50	40.01	77.88	0.7875
	2	96.94	59.99	2.500	74.12	0.7875
RELM	1	68.25	97.22	34.22	80.20	0.8150
	2	96.90	65.78	2.780	78.37	0.8150
IRELM	1	70.21	97.43	31.29	81.86	0.8307
	2	97.24	68.71	2.570	80.53	0.8307
GA-IRELM	1	73.57	97.09	26.39	83.71	0.8535
	2	97.09	73.61	2.910	83.73	0.8535
PSO-IRELM	1	84.31	85.57	12.05	84.93	0.8667
	2	88.96	87.95	14.43	88.45	0.8667
IPSO-IRELM	1	91.96	87.03	5.760	89.42	0.9063
	2	90.56	94.24	12.97	92.37	0.9063

of SVM, ELM, RELM, IRELM, and GA-IRELM is better, reaching more than 97%, while the TPR of PSO-IRELM and IPSO-IRELM is a little worse. This is because the first five algorithms misjudge a large amount of attack data as normal data, so for category 2, their TPR is very low compared to IPSO-IRELM, and IPSO-IRELM's TPR can reach 94.24%.

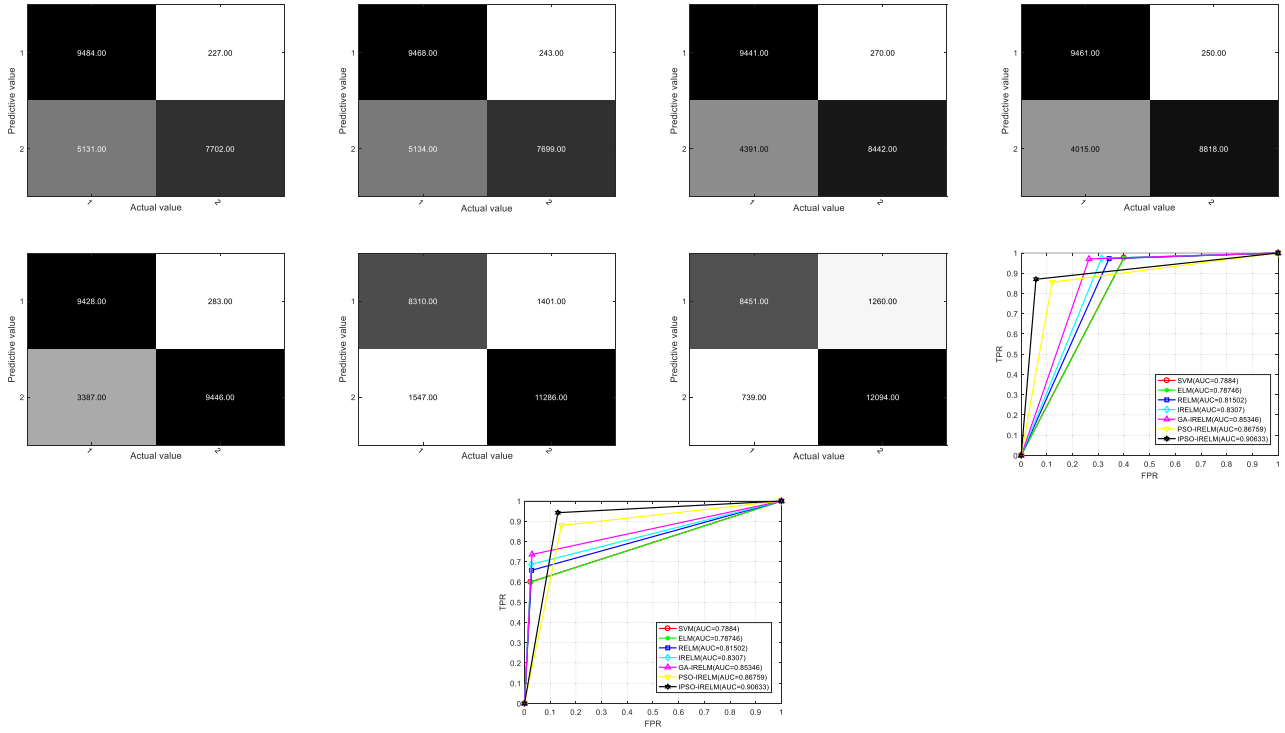

FIGURE 4. Binary classification confusion matrix and ROC curve comparison diagram.

TABLE 16. Accuracy of different algorithms (%).

Algorithm	SVM	ELM	RELM	IRELM	GA-IRELM	PSO-IRELM	IPSO-IRELM
Accuracy	75.16	78.23	81.56	81.32	82.10	84.37	85.58

TABLE 17. Performance evaluation index of each algorithm on normal.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	63.76	98.11	42.20	77.29	0.7795
ELM	67.00	97.48	36.33	79.42	0.8057
RELM	70.81	97.30	30.36	81.97	0.8347
IRELM	70.44	97.58	30.99	81.82	0.8329
GA-IRELM	71.71	96.52	28.82	82.28	0.8385
PSO-IRELM	74.45	97.02	25.20	84.25	0.8591
IPSO-IRELM	76.31	97.48	22.89	85.61	0.8729

Looking at the F-score and AUC, IPSO-IRELM is the best. In terms of precision, for category 2, IPSO-IRELM is a bit worse than other algorithms, but the difference is not big.

The confusion matrix is used to summarize the records in the data set according to the actual results and prediction results to realize visualization. As can be seen from Fig.4, IPSO-IRELM has the best performance in predicting category 2. At the time, the value corresponding to the second quadrant is the largest, which also corresponds to the TPR value of IPSO-IRELM corresponding to category 2 in Table. 15. ROC curve is a curve reflecting the relationship between TPR and FPR. The curve divides the graph into two parts, and the part below the curve is expressed as AUC, which is used to illustrate the accuracy of prediction. It can be seen from Fig.4 that the AUC of IPSO-IRELM is the highest in both category 1 and category 2, which fully indicates the excellent performance of IPSO-IRELM and verifies that IPSO-IRELM has better binary classification detection effect than other algorithms. Overall, IPSO-IRELM has the best performance.

F. NSL-KDD DATA SET MULTIVARIATE CLASSIFICATION EXPERIMENT RESULTS

Divide Normal, Dos, Probe, U2R, and R2L into five categories, denoted as 1, 2, 3, 4, and 5 respectively. The experiment has changed from two classifications to multiple classifications. The experimental results are shown in Table. 16-21. The confusion matrix and ROC curve diagram of multiclass classification are given in Fig.5.

From Table. 16, IPSO-IRELM has the highest accuracy rate of 85.58%. Since the NSL-KDD multi-classification data set is still an unbalanced data set, the intrusion detection

TABLE 18. Performance evaluation index of each algorithm on Dos.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	95.63	72.11	12.93	82.22	0.7959
ELM	96.59	79.06	13.89	86.95	0.8258
RELM	90.30	82.53	16.68	86.24	0.8293
IRELM	90.67	83.75	14.92	87.07	0.8441
GA-IRELM	82.79	72.49	25.48	77.29	0.7350
PSO-IRELM	95.26	81.86	21.96	88.05	0.7995
IPSO-IRELM	95.65	88.11	21.37	91.72	0.8337

TABLE 19. Performance evaluation index of each algorithm on probe.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	81.08	66.21	28.08	72.90	0.6907
ELM	77.30	66.25	31.37	71.35	0.6744
RELM	74.54	62.99	33.18	64.87	0.6491
IRELM	76.53	56.30	33.92	63.16	0.6119
GA-IRELM	55.15	67.04	27.97	60.51	0.6954
PSO-IRELM	59.19	84.96	32.66	69.78	0.7615
IPSO-IRELM	69.64	87.15	34.33	77.42	0.7641

TABLE 20. Performance evaluation index of each algorithm on U2R.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	NaN	0.000	0.000	NaN	0.5000
ELM	16.67	1.000	0.0089509	1.890	0.5050
RELM	55.56	2.500	0.0044755	4.780	0.5125
IRELM	NaN	0.000	0.0044755	NaN	0.5000
GA-IRELM	NaN	0.000	0.000	NaN	0.5000
PSO-IRELM	NaN	0.000	0.000	NaN	0.5000
IPSO-IRELM	76.92	5.000	0.0044755	9.390	0.5250

TABLE 21. Performance evaluation index of each algorithm on R2L.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	NaN	0.000	0.000	NaN	0.5000
ELM	59.11	4.830	0.035371	8.930	0.5240
RELM	73.48	8.750	0.030318	15.64	0.5436
IRELM	32.85	4.970	0.070743	8.640	0.5245
GA-IRELM	NaN	0.000	0.000	NaN	0.5000
PSO-IRELM	25.00	0.036311	0.015159	0.072516	0.5001
IPSO-IRELM	94.71	7.810	0.030318	14.42	0.5389

capabilities of each algorithm are further analyzed in terms of precision, TPR, FPR, F-score and AUC.

It can be seen from Table. 17 that for the Normal type data, the TPR of each algorithm is better, all above 97%. For other performance indicators, the precision, FPR, F-score and AUC of IPSO-IRELM are the highest among the comparison algorithms, indicating that IPSO-IRELM has better classification performance for Normal type data.

It can be seen from Table. 18 that for DOS data, IPSO-IRELM has the highest precision, TPR and F-score, but FPR is a little worse than SVM, ELM, RELM and IRELM. This is because in IPSO-IRELM, other types of data are predicted to have too many DOS data, resulting in slightly poor FPR. For AUC, IPSO-IRELM is second only to IRELM. In general, IPSO-IRELM has a strong ability to identify DOS type attacks.

From Table. 19, for Probe type data, the TPR of IPSO-IRELM is the highest, but the FPR is also the highest. This is because in addition to the better detection ability of IPSO-IRELM for Probe type data, other types of data will also be mistakenly detected as Probe type data, resulting in the highest FPR and low precision, but the F-score and AUC are still the highest.

It can be seen from Table. 20 that for U2R data, there are only 11 training data and 200 test data, so SVM, ELM, RELM, IRELM, GA-IRELM and PSO-IRELM all have poor recognition effect for U2R data. Although the TPR of IPSO-IRELM is only 5%, it is still the best, and the F-score and AUC are also the highest, with an precision rate of 76.92%, which other algorithms cannot do, indicating that IPSO-IRELM also has a certain ability to detect a few types of data, but it needs to be improved.

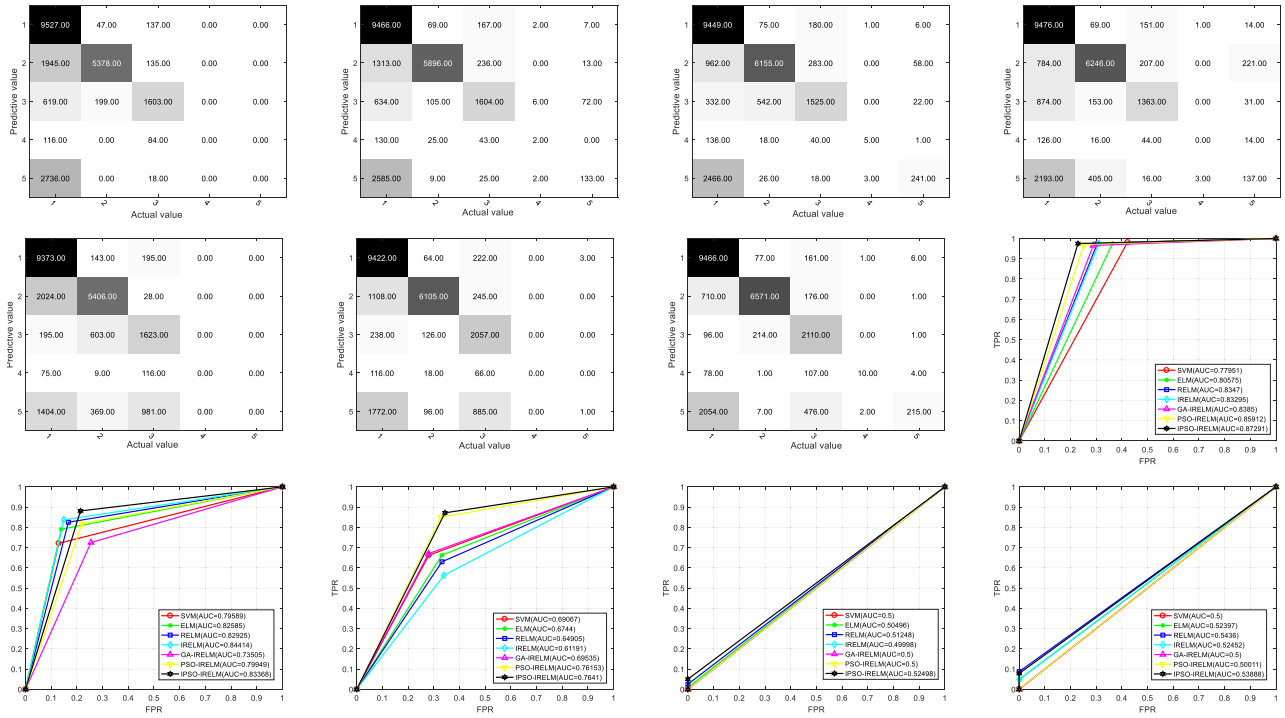

FIGURE 5. Multiple classification confusion matrix and ROC curve comparison diagram.

From Table. 21, for R2L type data, SVM and GA-IRELM still cannot recognize R2L data because it is still a minority type of data. Other algorithms have recognition capabilities, but they are still not prominent. IPSO-IRELM's TPR, F-score and AUC are second only to RELM, but the precision rate is higher than RELM. This is because RELM recognizes that other types of data are more R2L data during classification, which leads to a higher TPR than IPSO-IRELM, but the accuracy rate is not as good as IPSO-IRELM.

Through the analysis of Table. 17-21, we can conclude that in the multivariate classification problem, it is not only necessary to consider some performance indicators. When the TPR conflicts with the precision rate, the comparison between models is relatively complicated. The F-score just reconciles the TPR and precision rate. In Table. 17-21, the F value is better, indicating that IPSO-IRELM has a better classification effect than other algorithms.

In the analysis of Table. 17-21, we found that when the classification error occurs, only looking at the precision rate and TPR can potentially see which category is more, which category is less. We do not know how many types of specific errors are classified into. Therefore, we still use the confusion matrix to show the difference of the data with its color difference, brightness, etc., which is easy to understand. The dark area indicates that the true value and the predicted value overlap more, and the light area is the opposite. It can be seen from Fig.5 that the dark areas of the first three categories of each algorithm are concentrated on the diagonal, and the latter two categories are concentrated on the lower left corner, which is caused by the imbalance of the data. Because the latter two types of data are too small, the algorithm's recognition ability is not sufficient for high-level recognition, which leads to the recognition result tends to the category 1. From the comparison between the algorithms, for category 1, the classification accuracy of each algorithm is not much different. For category 2 and category 3, the good classification performance of the IPSO-IRELM algorithm is reflected, which is the best among all the comparison algorithms. As for the minority categories 4 and 5, IPSO-IRELM also has a certain recognition ability, while the recognition ability of other algorithms is basically zero.

The ROC curve has a huge advantage. When the distribution of positive and negative samples changes, its shape can remain basically unchanged. Therefore, the ROC curve can reduce the interference caused by different test sets and more objectively measure the performance of the model itself. It can be seen from Fig.5 that IPSO-IRELM has the best AUC for Normal, Dos, Probe and U2R. For R2L type, it is second only to RELM with a difference of 0.00472.

In summary, on the UCI balanced data set, the performance of IPSO-IRELM is better than that of IRELM and the traditional RELM algorithm, which shows the necessity of the improved method. On the NSL-KDD binary classification and multivariate classification data sets, it is also verified that the IPSO-IRELM algorithm has better classification performance.

G. UNSW-NB15 DATA SET MULTIVARIATE CLASSIFICATION EXPERIMENT RESULTS

The UNSW-NB15 dataset contains new patterns of modern network features with nine types of attacks, so this paper also uses the UNSW-NB15 dataset to verify whether the proposed

TABLE 22. Accuracy of different algorithms (%) and time(s).

Algorithm	SVM	ELM	RELM	IRELM	GA-IRELM	PSO-IRELM	IPSO-IRELM
Accuracy	68.84	70.29	70.28	70.30	83.49	84.56	88.53
Time	6096	263	251	220	474	452	436

TABLE 23. Performance evaluation index of each algorithm on normal.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	99.06	60.56	0.470	75.17	0.8005
ELM	95.33	65.21	2.610	77.45	0.8130
RELM	95.31	65.21	2.620	77.44	0.8130
IRELM	95.35	65.20	2.600	77.45	0.8130
GA-IRELM	85.19	72.30	10.26	78.22	0.8102
PSO-IRELM	92.59	67.46	4.410	78.05	0.8153
IPSO-IRELM	91.58	77.01	5.780	83.67	0.8562

TABLE 24. Performance evaluation index of each algorithm on Dos.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	48.15	2.230	0	4.250	0.5111
ELM	29.03	7.040	1.66e-02	11.34	0.5351
RELM	29.00	7.040	1.66e-02	11.33	0.5351
IRELM	29.24	6.990	1.53e-02	11.29	0.5349
GA-IRELM	32.73	0.440	3.83e-03	0.870	0.5022
PSO-IRELM	33.73	4.160	0	7.400	0.5208
IPSO-IRELM	31.25	2.690	0	4.950	0.5135

TABLE 25. Performance evaluation index of each algorithm on exploits.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	50.67	87.28	2.760	64.12	0.9226
ELM	54.36	85.80	1.890	66.55	0.9196
RELM	54.26	85.82	1.890	66.48	0.9196
IRELM	54.32	85.91	1.890	66.56	0.9201
GA-IRELM	27.02	74.01	13.94	39.59	0.8004
PSO-IRELM	49.77	83.29	3.500	62.31	0.8990
IPSO-IRELM	46.48	84.15	4.420	59.89	0.8987

TABLE 26. Performance evaluation index of each algorithm on fuzzers.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	23.51	69.76	14.88	35.16	0.7744
ELM	23.65	56.40	12.50	33.33	0.7195
RELM	23.61	56.14	12.51	33.24	0.7182
IRELM	23.65	56.38	12.51	33.32	0.7194
GA-IRELM	22.68	2.010	0.240	3.700	0.5089
PSO-IRELM	19.31	55.30	12.20	28.26	0.7155
IPSO-IRELM	24.51	43.68	6.440	31.40	0.6862

algorithm can have a better classification effect in the current network environment. Since the UNSW-NB15 dataset is also a non-equilibrium dataset, neither the proposed algorithm nor the comparison algorithms can identify the four attacks: Analysis, Backdoor, shellcode and Worms in the experiment, so the final experimental results are shown in Table. 22-28 and Fig.6.

From Table. 22, it is concluded that IPSO-IRELM has the highest accuracy rate of 88.53%. Also, IRELM has the shortest training time, which indicates that IRELM is more efficient. The training time of IPSO-IRELM increases due to the optimization of IPSO, but it does not increase compared with GA-IRELM and PSO-IRELM, which indicates that our improvement of PSO does not increase its complexity. Similarly, since the UNSW-NB15 multiclassification dataset is also an unbalanced dataset, the intrusion detection capability of each algorithm is further analyzed in terms of precision, TPR, FPR, F-score, and AUC.

From Fig.6, a portion of the Exploits data is mistaken for Dos; a portion of the Fuzzers data is mistaken for Exploits;

TABLE 27. Performance evaluation index of each algorithm on generic.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	99.79	96.24	1.58e-03	97.98	0.9812
ELM	99.81	96.29	1.58e-03	98.01	0.9814
RELM	99.81	96.29	1.58e-03	98.01	0.9814
IRELM	99.81	96.29	1.58e-03	98.01	0.9814
GA-IRELM	91.64	96.27	0.220	93.90	0.9803
PSO-IRELM	98.47	96.24	4.57e-02	97.34	0.9810
IPSO-IRELM	98.86	96.24	3.94e-04	97.54	0.9810

TABLE 28. Performance evaluation index of each algorithm on reconnaissance.

Algorithm	Precision(%)	TPR(%)	FPR(%)	F-score (%)	AUC
SVM	50.37	59.18	1.580	54.42	0.7880
ELM	41.45	66.10	2.260	50.95	0.8192
RELM	41.45	66.16	2.260	50.97	0.8195
IRELM	41.50	66.08	2.260	50.98	0.8191
GA-IRELM	5.260	2.86e-02	0	5.69e-02	0.5001
PSO-IRELM	25.35	3.150	0.270	5.600	0.5144
IPSO-IRELM	57.63	25.29	0.530	35.15	0.6238

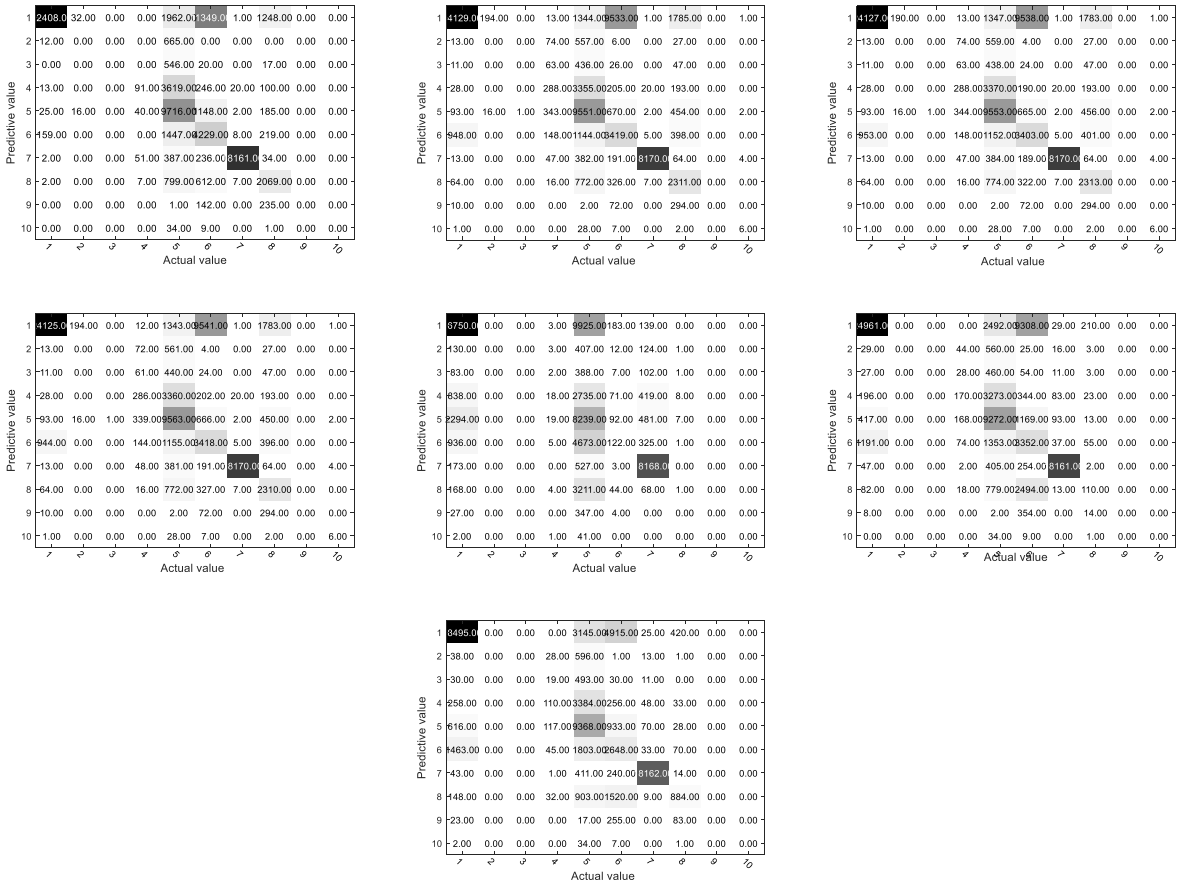

FIGURE 6. UNSW-NB15 dataset multiple classification confusion matrix.

and a portion of the Normal data is mistaken for Fuzzers in the final classification results obtained by all algorithms, which leads to the low precision rates in Table. 24-26. For Generic data, as in Table. 27, each algorithm can detect it well, but IPSO-IRELM has the smallest FPR, indicating that IPSO-IRELM can detect Generic while not mistaking Generic for other classes. For Normal data, as shown in Table. 23, the TPR, F-value and AUC of IPSO-IRELM are the highest, which indicates that IPSO-IRELM has better classification performance for Normal data. In summary,

IPSO-IRELM not only has the highest accuracy rate, but also the training time does not increase. Although some classifications are not recognized for the UNSW-NB15 dataset, it has the best classification results for the remaining six classifications, indicating that IPSO-IRELM still has better classification ability in modern network environments as well.

V. CONCLUSION

We propose an IPSO-IRELM intrusion detection model, which overcomes the shortcomings of offline learning of traditional intrusion detection models. It uses sequential learning to classify the types of attacks in the network, and jointly optimizes the initial weights and deviations of IRELM through improved particle swarms. The experimental results show that IPSO-IRELM has obvious advantages in all evaluation indicators, no matter on the UCI balance data set or the NSL-KDD intrusion detection data set, or the UNSW-NB15 data set, which contains many new types of attacks. The next step will be to study how to have a higher recognition rate for minority samples on unbalanced data sets, and apply the IPSO-IRELM algorithm to actual dynamic intrusion detection networks to test its classification effect in the real environment.

REFERENCES

[1] J. Gong, X. D. Zang, Q. Su, X. Y. Hu, and J. Xu, "Survey of network security situation awareness," *J. Softw.*, vol. 28, no. 4, pp. 1010–1026, 2017.

[2] M. H. Haghighat and J. Li, "Intrusion detection system using voting-based neural network," *Tsinghua Sci. Technol.*, vol. 26, no. 4, pp. 484–495, Aug. 2021.

[3] T. Museba, F. Nelwamondo, and K. Ouahada, "An adaptive heterogeneous online learning ensemble classifier for nonstationary environments," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–11, Mar. 2021.

[4] K. Zheng, Z. Cai, X. Zhang, Z. Wang, and B. Yang, "Algorithms to speedup pattern matching for network intrusion detection systems," *Comput. Commun.*, vol. 62, pp. 47–58, May 2015.

[5] D. Zhao, X. Zhu, and T. Xu, "Improvement of algorithm for pattern matching in intrusion detection," in *Proc. 5th IEEE Int. Conf. Broadband Netw. Multimedia Technol.*, Guilin, China, Nov. 2013, pp. 281–284.

[6] Z. Lin and D. Hongle, "Research on SDN intrusion detection based on online ensemble learning algorithm," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Haikou City, China, Dec. 2020, pp. 114–118.

[7] C. Nixon, M. Sedky, and M. Hassan, "Practical application of machine learning based online intrusion detection to Internet of Things networks," in *Proc. IEEE Global Conf. Internet Things (GCIoT)*, Dubai, United Arab Emirates, Dec. 2019, pp. 1–5.

[8] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[9] V. Kuppili, M. Biswas, A. Sreekumar, H. S. Suri, L. Saba, D. R. Edla, R. T. Marinhoe, J. M. Sanches, and J. S. Suri, "Extreme learning machine framework for risk stratification of fatty liver disease using ultrasound tissue characterization," *J. Med. Syst.*, vol. 41, no. 10, pp. 1–20, Oct. 2017.

[10] C. Cervellera and D. Maccio, "An extreme learning machine approach to density estimation problems," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3254–3265, Oct. 2017.

[11] J. M. Martínez-Martínez, P. Escandell-Montero, E. Soria-Olivas, J. D. Martín-Guerrero, R. Magdalena-Benedito, and J. Gómez-Sanchis, "Regularized extreme learning machine for regression problems," *Neurocomputing*, vol. 74, no. 17, pp. 3716–3721, Oct. 2011.

[12] Y. Zhang, Q. Wu, and J. Hu, "An adaptive learning algorithm for regularized extreme learning machine," *IEEE Access*, vol. 9, pp. 20736–20745, 2021.

[13] C. Gautam, A. Tiwari, S. Suresh, and K. Ahuja, "Adaptive online learning with regularized kernel for one-class classification," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 3, pp. 1917–1932, Mar. 2021.

[14] J. Qu, K. Ren, and X. Shi, "Binary grey wolf optimization-regularized extreme learning machine wrapper coupled with the boruta algorithm for monthly streamflow forecasting," *Water Resour. Manage.*, vol. 35, no. 3, pp. 1029–1045, Feb. 2021.

[15] W. Dong, S. Zhang, A. Jiang, W. Jiang, L. Zhang, and M. Hu, "Intelligent fault diagnosis of rolling bearings based on refined composite multi-scale dispersion q-complexity and adaptive whale algorithm-extreme learning machine," *Measurement*, vol. 176, May 2021, Art. no. 108977.

[16] R. Kumar, M. P. Singh, B. Roy, and A. H. Shahid, "A comparative assessment of metaheuristic optimized extreme learning machine and deep neural network in multi-step-ahead long-term rainfall prediction for all-Indian regions," *Water Resour. Manage.*, vol. 35, no. 6, pp. 1927–1960, Apr. 2021.

[17] Z. Lei, Q. Zhu, Y. Zhou, B. Sun, W. Sun, and X. Pan, "A GAPSO-enhanced extreme learning machine method for tool wear estimation in milling processes based on vibration signals," *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 8, no. 3, pp. 745–759, Apr. 2021.

[18] N. Kanimozhi and G. Singaravel, "Hybrid artificial fish particle swarm optimizer and kernel extreme learning machine for type-II diabetes predictive model," *Med. Biol. Eng. Comput.*, vol. 59, no. 4, pp. 841–867, Mar. 2021.

[19] Y. Wang, R. Li, and Y. Chen, "Accurate elemental analysis of alloy samples with high repetition rate laser-ablation spark-induced breakdown spectroscopy coupled with particle swarm optimization-extreme learning machine," *Spectrochimica Acta B, At. Spectrosc.*, vol. 177, Mar. 2021, Art. no. 106077.

[20] L. Zhang and D. Zhang, "Evolutionary cost-sensitive extreme learning machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 3045–3060, Dec. 2017.

[21] W.-Y. Deng, Q.-H. Zheng, L. Chen, and X.-B. Xu, "Research on extreme learning of neural networks," *Chin. J. Comput.*, vol. 33, no. 2, pp. 279–287, Apr. 2010.

[22] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[23] Y. Yu and Z. Sun, "Sparse coding extreme learning machine for classification," *Neurocomputing*, vol. 261, pp. 50–56, Oct. 2017.

[24] Y.-P. Zhao, Q.-K. Hu, J.-G. Xu, B. Li, G. Huang, and Y.-T. Pan, "A robust extreme learning machine for modeling a small-scale turbojet engine," *Appl. Energy*, vol. 218, pp. 22–35, May 2018.

[25] W. Xiao, J. Zhang, Y. Li, S. Zhang, and W. Yang, "Class-specific cost regulation extreme learning machine for imbalanced classification," *Neurocomputing*, vol. 261, pp. 70–82, Oct. 2017.

[26] Q.-Y. Zhu, A. K. Qin, P. N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognit.*, vol. 38, no. 10, pp. 1759–1763, Oct. 2005.

[27] Y. Xu and Y. Shu, "Evolutionary extreme learning machine-based on particle swarm optimization," in *Proc. 3rd Int. Symp. Neural Netw.*, Chengdu, China, 2006, pp. 1917–1932.

[28] E. M. N. Figueiredo and T. B. Ludermir, "Investigating the use of alternative topologies on performance of the PSO-ELM," *Neurocomputing*, vol. 127, pp. 4–12, Mar. 2014.

[29] W. Zheng, Y. Qian, and H. Lu, "Text categorization based on regularization extreme learning machine," *Neural Comput. Appl.*, vol. 22, nos. 3–4, pp. 447–456, Mar. 2013.

[30] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, Piscataway, NJ, USA, Nov. 1995, pp. 1942–1948.

[31] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congr. Evol. Comput.*, La Jolla, CA, USA, Jul. 2000, pp. 84–88.

[32] G. Q. Bao and K. F. Mao, "Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Guilin, China, Dec. 2009, pp. 2134–2139.

[33] T. Ziyu and Z. Dingxue, "A modified particle swarm optimization with an adaptive acceleration coefficients," in *Proc. Asia–Pacific Conf. Inf. Process.*, Jul. 2009, pp. 330–332.
[34] S. Mirjalili, A. Lewis, and A. S. Sadiq, "Autonomous particles groups for particle swarm optimization," *Arabian J. Sci. Eng.*, vol. 39, no. 6, pp. 4683–4697, Jun. 2014.
[35] C. L. Zhang and S. F. Ding, "A stochastic configuration network based on chaotic sparrow search algorithm," *Knowl.-Based Syst.*, vol. 220, May 2021, Art. no. 106924.
[36] Machine Learning Repository. *Center for Machine Learning and Intelligent Systems*. Accessed: Mar. 22, 2021. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.php
[37] J. Sonchack, A. Dubey, A. J. Aviv, J. M. Smith, and E. Keller, "Timing-based reconnaissance and defense in software-defined networks," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl.*, New York, NY, USA, Dec. 2016, pp. 89–100.
[38] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Jan. 2016.
[39] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

YANQIANG TANG was born in Shandong, China, in 1997. He received the B.S. degree from the Wuhan University of Technology, in 2019. He is currently pursuing the M.S. degree with Air Force Engineering University. His research interests include research on intelligent algorithm, machine learning, and network security situation prediction.

CHENGHAI LI (Member, IEEE) was born in Shandong, China, in 1966. He received the Ph.D. degree from Air Force Engineering University, China. He is currently a Professor with Air Force Engineering University. His research interests include evidence theory, embedded systems, and network security.