Novel Network Intrusion Detection Method Based on IPSO-MTWSVM Model

Yangke Yuan, Hong Dai, Zijian Wu, and Di Meng

***Abstract*—Multi-classification Twin Support Vector Machine model (MTWSVM) is put forward in order to solve the issues of low detection rates and weaken practicability in current network intrusion detection models. In the first place, the Adaptive Synthetic sampling method (ADASYN) is used to balance the dataset. And in the second place, the numerical and normalized methods are used to process the dataset. Ultimately, the data is balanced and simple enough to improve the calculability and convergence speed of the model. For the multi-classification problem, the classification strategy is used to build multi-classification model based on parameter optimization because the parameters have a greater impact on the model. The Improved Particle Swarm Optimization algorithm (IPSO) is used to find the global optimal parameters iteratively in the paper. The KDD'99 dataset is used to perform the network intrusion detection experiments. The experimental results show that the Multi-classification Twin Support Vector Machine model based on parameter optimization can effectively improve the detection performance compared with other models. At the same time, the experimental results show the overall detection accuracy and the detection accuracy of various attacks have been significantly improved. Therefore, the intrusion detection model proposed in the paper has validity and practicability in the network intrusion detection.**

***Index Terms*—Intrusion Detection, Multi-classification, Twin Support Vector Machine, Particle Swarm Optimization**

I. Introduction

The investigation of internet security reports in recent years shows that the target network intrusion incidents are becoming more complex. The target areas and frequencies of attacks are also expanding. In the military aspect, the hacker invaded the enemy's network system to browse, steal and delete war strategy and technology, etc. In the government and enterprises, hackers illegally stole internal confidential database files and top secret documents through cyber-attacks [1]. In terms of individual users, security issues such as user privacy leaks are also emerging one after another. In order to avoid the network security issues, the importance of network intrusion detection is continuously and deeply understood by people. And it is of great significance for improving the detection accuracy of various attacks. Meanwhile, practical methods need to be studied so as to detect attacks more efficiently.

With the development of machine learning algorithms, researchers put forward a large number of intrusion detection technologies based on machine learning [2]. As a kind of machine learning algorithm, Support Vector Machine (SVM) has received widespread attention because of its superior performance. The related technologies have also been applied in the research of intrusion detection. However, it has some problems such as weak generalization ability and the low detection rate. In order to overcome these obstacles, the Twin Support Vector Machine (TWSVM) [3], as a derivative technology of the SVM was proposed and quickly come into notice of scholars [4]. In 2014, He and Zheng built a network intrusion detection model through TWSVM. The research proved that TWSVM saved nearly four times the training time compared with SVM, and the prediction accuracy of the model was also improved accordingly [5]. In 2015, Zhong established a network intrusion detection model using a regular term in the objective function based on improved least squares TWSVM. And it reduced the complexity of the model by regularizing the prediction coefficients of the penalty model to establish a simple and effective prediction model [6]. In 2019, Li, et al. proposed a TWSVM model based on artificial fish swarm algorithm. It used artificial fish swarm algorithm to optimize parameters and avoid the blindness of parameter selection [7]. In 2020, Zhou, et al. proposed the grey wolf optimized TWSVM model. The grey wolf optimization algorithm was used to give full play to the advantages of sub-classifiers by selecting appropriate parameters for different sub-classifiers. It improved the classification performance of the model [8].

In conclusion, more and more TWSVM models and their improved algorithms have been studied deeply. According to the current research progress, the existing network intrusion detection methods still have the following limitations. Firstly, the overall detection rate is low. Secondly, the User-to-Root (U2R) and Remote-to-Local (R2L) attacks detection rates are low. Moreover, some intrusion detection methods in application are not practical because they ignore the security risks brought by attacks using small samples. Finally, the process of parameter optimization is easy to trap into local optimization. The paper mainly focusses on these issues. At first, the sample number of various attacks is increased by oversampling technology to enhance the practicability of the model. Subsequently, the numerical, normalized and standardized methods are used to simplify the dataset and prevent the data gradient explosion. In the end, the TWSVM multi-classification model is built. The parameters are optimized by the Particle Swarm Optimization (PSO) algorithm. At the same time, the linear decreasing inertia

Manuscipt received June 11, 2021; revised March 29, 2022. This work is supported by Graduate Education Reform and Innovation Entrepreneurship Project, University of Science and Technology Liaoning (2021YJSCX09).

Yangke Yuan is Postgraduate of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (e-mail: 1056741362@qq.com).

Hong Dai is Professor of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (corresponding author: tel:+086-18642268599; fax: 0412-5929818; e-mail: dear_red9@163.com)

Zijian Wu is Postgraduate of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (e-mail: 939674778@qq.com).

Di Meng is Postgraduate of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (e-mail: 997184646@qq.com).

weight is introduced on the basis of the PSO algorithm to avoid the model trapping into local optimization in the optimization process. The convergent speed and detection performance of the model are improved.

II. METHODOLOGY

Sampling

The evaluation data originate from KDD'99 dataset [9]. However, there is an extremely unbalanced data distribution in the dataset. For instance, in 10% of the KDD'99 training set, the total number of Denial-of-Service (DoS) attacks can reach hundreds of thousands. There are only 21 land attacks in DoS attacks. The total number of U2R attacks is only 52. It contains a smaller sample of attacks. In practice, if the network protection system cannot detect the attacks using a small number of samples, it will seriously endanger the information system. In addition, the risk of over fitting will also exist due to the imbalanced dataset during the experiment. Therefore, it is necessary to increase the samples size of attacks using small samples by oversampling technology. After that, the practicability of the model will be improved. Oversampling work will use the Adaptive Synthetic sampling method (ADASYN) to balance small samples dataset [10]. The flow of ADASYN is described as follows.

Step1. The number of synthetic samples is defined as follows (formula 1):

$$G=(m_l-m_s)\times\beta \tag{1}$$

m_l is the number of samples in the majority class and m_s is the number of samples in the minority class. $\beta\in$ [0, 1] is a random number which is the sampling ratio. If β=1, the positive-negative ratio after sampling is 1:1.

Step2. The proportion of the majority class in k nearest neighbors is introduced below (formula 2):

$$r_i=\Delta_i/k \tag{2}$$

Where Δ_i is the number of samples of the majority class in k nearest neighbors, i=1, 2, 3..., m_s.

Step3. $\widehat{r_i}$ is defined as follow (formula 3):

$$\widehat{r_i}=r_i/\sum_{i=1}^{m_s} r_i \tag{3}$$

Where $\widehat{r_i}$ is the standardized result of r_i.

Step4. According to the sample weight, the number of new samples needed to generate minority samples is calculated as follow (formula 4):

$$g=\widehat{r_i}\times G. \tag{4}$$

Step5. The new samples are described below (formula 5):

$$x_{new}=x+rand(0,1)\cdot(x_i-x) \tag{5}$$

The under-sampling method also needs to be used after oversampling. The random sampling technology is the most common under-sampling method. Its working principle is that randomly selects n individuals from the population as samples so that each sample has an equal probability of being selected. The experiment in the paper uses random sampling without replacement in the sampling process. The number of samples for each type of attacks is sufficient to ensure the randomness of sampling after the KDD'99 dataset is over-sampled by ADASYN.

Twin Support Vector Machine

TWSVM was proposed by Jaya Deva in 2007 on the basis of Generalized Eigenvalue Proximal Support Vector Machine (GEPSVM). Therefore, TWSVM is essentially similar to GEPSVM. TWSVM is used in regression or classification problems. The paper mainly discusses the application of TWSVM in the latter. When solving two-classification problems, TWSVM constructs a hyperplane for each kind of samples so that each kind of samples is likely to be closed to the hyperplane of its own class and far from another hyperplane [11]. Two hyperplanes of TWSVM are obtained by solving two Quadratic Programming Problems (QPPs). The constraints of each QPP are only related to one type of samples. The solution process of TWSVM for linearly separable and non-linearly separable two-classification problems as follows.

i. *Linearly separable*

For linearly separable problems, two non-parallel hyperplanes of TWSVM model are given as follows (formula 6):

$$\begin{cases} x^{\mathrm{T}}w_{(1)}+b_{(1)}=0 \\ x^{\mathrm{T}}w_{(2)}+b_{(2)}=0 \end{cases} \tag{6}$$

x is the sample vector. $w_{(1)}$ and $w_{(2)}$ are the normal vectors of the hyperplanes respectively. $b_{(1)}$ and $b_{(2)}$ are the offsets of the two hyperplanes.

For making each hyperplane close to one type of samples and faring away from another, it is necessary to obtain the normal vector w and the offset b of the hyperplane which can be implemented by solving the QPPs as follows (formula 7 and 8):

$$\begin{cases} \underset{w_{(1)}b_{(1)}}{min}\ \frac{1}{2}\left\|Aw_{(1)}+e_1b_{(1)}\right\|^2+c_1e_2^T\xi_2 \\ S.t.\ -(Bw_{(1)}+e_2b_{(1)})+\xi_2\geq e_2,\xi_2\geq 0 \end{cases} \tag{7}$$

$$\begin{cases} \underset{w_{(2)}b_{(2)}}{min}\ \frac{1}{2}\left\|Bw_{(2)}+e_2b_{(2)}\right\|^2+c_2e_1^T\xi_1 \\ S.t.\ (Aw_{(2)}+e_1b_{(2)})+\xi_1\geq e_1,\xi_1\geq 0 \end{cases} \tag{8}$$

Among them, c_1, c_2>0 are the penalty parameters. e_1, e_2 are column vectors with elements equal to 1. ξ_1, ξ_2 are the slack variables of the hyperplanes respectively. **A** and **B** represent the positive and negative matrix of the samples. Each row represents a sample point. The QPPs are transformed into the dual problems as shown below (formulas 9 and 10):

$$\begin{cases} max_\alpha\ e_2^{\mathrm{T}}\alpha-\frac{1}{2}\alpha^{\mathrm{T}}P(Q^{\mathrm{T}}Q)^{-1}P^T\alpha \\ \quad S.t.\ 0\leq\alpha\leq c_1e_2 \end{cases} \tag{9}$$

$$\begin{cases} max_\beta\ e_1^T\beta-\frac{1}{2}\beta^{\mathrm{T}}Q(P^{\mathrm{T}}P)^{-1}Q^T\beta \\ \quad S.t.\ 0\leq\beta\leq c_2e_1 \end{cases} \tag{10}$$

Where $\mathbf{P}=[\mathbf{B},e_2]$, $\mathrm{Q}=[\mathbf{A},e_1]$. The α and β are Lagrange multipliers.

The hyperplanes of the TWSVM classifier can be obtained through solving the above dual problems. The classification of new samples can be solved by the decision function. The function is described below (formula 11):

$$\mathrm{Label}(x)=\underset{i=1,2}{argmin}\left(x^{\mathrm{T}}w_{(i)}+b_{(i)}\right)/\left\|w_{(i)}\right\| \tag{11}$$

The Label(x) is closer to hyperplanes. The label is the classification of new samples.

ii. *Non-linearly separable*

TWSVM algorithm is used for non-linearly separable problems. It is necessary to combine the kernel function

technology for processing. The samples in the original feature space are mapped to the high-dimensional regeneration space. So that the samples obtained by mapping are linearly separable. Then two hyperplanes of TWSVM algorithm are established in the high-dimensional space. The function *K (x, y)* is used to represent the kernel function. The hyperplane based on the kernel function is constructed below (formula 12) :

$$\begin{cases} K\left(x^{\mathrm{T}}, C^{\mathrm{T}}\right) w_1 + b_1 = 0 \\ K\left(x^{\mathrm{T}}, C^{\mathrm{T}}\right) w_2 + b_2 = 0 \end{cases} \tag{12}$$

Among them, $\mathbf{C}=\left[\boldsymbol{A}^{\mathrm{T}}, \boldsymbol{B}^{\mathrm{T}}\right]^{\mathrm{T}}$ is the samples matrix. The matrices of $\boldsymbol{A}$ and $\boldsymbol{B}$ represent the positive and negative class matrices, respectively. Each row represents a sample. x is a sample vector. w_1 and w_2 are used to represent normal vectors of the hyperplanes. b_1 and b_2 are used to represent offsets of the hyperplanes.

The values of normal vector w and offset b of the hyperplanes are obtained through the QPPs as shown below (formula 13 and 14):

$$\begin{cases} \min\limits_{w_1 b_1} \frac{1}{2} \left\| K\left(A, C^{\mathrm{T}}\right) w_1 + e_2 b_1 \right\|^2 + c_1 e_2^{\mathrm{T}} \xi_2 \\ S.t.\ -\left(K\left(B, C^{\mathrm{T}}\right) w_1 + e_2 b_1\right) + \xi_2 \geq e_2, \xi_2 \geq 0 \end{cases} \tag{13}$$

$$\begin{cases} \min\limits_{w_2 b_2} \frac{1}{2} \left\| K\left(B, C^{\mathrm{T}}\right) w_2 + e_1 b_2 \right\|^2 + c_2 e_1^{\mathrm{T}} \xi_1 \\ S.t.\ -\left(K\left(A, C^{\mathrm{T}}\right) w_2 + e_1 b_2\right) + \xi_1 \geq e_2, \xi_1 \geq 0 \end{cases} \tag{14}$$

The hyperplanes can be obtained based on the kernel function of the TWSVM classifier by solving the QPPs. The decision function solves the classification of the new samples. The function is defined as follow (formula 15):

$$\text{Label}(x)\ argmin_{i=1,2} = \left(K\left(x, C^{\mathrm{T}}\right) w_i + b_i\right) \tag{15}$$

The Label(x) is similar to the linearly separable case. The new samples are classified into a class which is closer to hyperplane.

Particle Swarm Optimization

PSO algorithm is an evolutionary computing technology. It is derived from the study of bird predation. Its basic idea is to find the optimal solution through the collaboration and information sharing among individuals in the group [12]. It is assumed that there is an N-dimensional space in which M particles are uniformly distributed in the PSO algorithm. Each particle has a fitness value determined by an optimization function. A speed determines its flight distance and direction. The PSO algorithm seeks for the optimal solution through the continuous iteration. Therefore, it first needs to initialize a group of random particles, and then iterates again. The optimal position of the particle itself should be recorded as the individual optimal position (*Pbest*) during iterations. And then all the particles should follow the current optimal particle to find the global optimal position (*Gbest*) in the space. *Pbest* and *Gbest* are continuously updated to find the optimal solution in each iteration process. The position and velocity of each particle in the current space can be marked as $X_i=(X_{i1}, X_{i2}, \ldots, X_{iN})$ and $V_i=(V_{i1}, V_{i2}, \ldots, V_{iN})$. The particles velocity update formula is described below (formula 16):

$$\begin{cases} V_i^{k+1} = w \cdot V_i^k + c_1 \cdot rand() \cdot \left(Pbest - X_i^k\right) \\ \qquad + c_2 \cdot rand() \cdot \left(Gbest - X_i^k\right) \\ X_i^{k+1} = X_i^k + V_i^{k+1} \end{cases} \tag{16}$$

w is an inertia weight. c_1 and c_2 are the learning factors. *rand* () is a random number between zero and one. It mainly used to reflect the randomness of PSO algorithm during the operation. k is the number of current iterations.

III. Build Model

A. Multi-classification Twin Support Vector Machine

The "One-Versus-One" (O-V-O) multi-classification strategy was initially applied to SVM model and achieved good results. The strategy was adopted to build the MTWSVM model after a great deal of researches. The MTWSVM model is formed by improving TWSVM model in the paper. For the M classification problem, a total of $M\cdot(M\text{-}1)/2$ sub-classifiers need to be constructed. And each sub-classifier is a two-classifier between two types of samples belongs to the TWSVM model. That is, a total of $M\cdot(M\text{-}1)$ classification hyperplanes need to be found [11]. Each sub-classifier of MTWSVM needs to use two types of samples and simplify the multi-classification problem for solving a two-classification problem. Since the actual experiment in the paper is facing a non-linearly separable problem, the kernel function *K (x, y)* needs to be introduced into MTWSVM. After that, the hyperplanes are defined as follows (formula 17):

$$\begin{cases} K\left(A_i, C^{\mathrm{T}}\right) w_{ij} + b_{ij} = 0 \\ K\left(A_j, C^{\mathrm{T}}\right) w_{ji} + b_{ji} = 0 \end{cases} \tag{17}$$

The QPPs need to be solved in MTWSVM by the followings (formula 18 and 19) when facing non-linearly separable problems.

$$\begin{cases} \min\limits_{w_{ij} b_{ij}} \frac{1}{2} \left\| K\left(A_i, C^{\mathrm{T}}\right) w_{ij} + e_{ij}^{(1)} b_{ij} \right\|^2 + \frac{c_{ij}}{2} \xi_{ij}^{\mathrm{T}} \xi_{ij} \\ S.t.\ \left(K\left(A_j, C^{\mathrm{T}}\right) w_{ij} + e_{ji}^{(2)} b_{ij}\right) + \xi_{ij} \geq e_{ji}^{(2)}, \xi_{ij} \geq 0 \end{cases} \tag{18}$$

$$\begin{cases} \min\limits_{w_{ji} b_{ji}} \frac{1}{2} \left\| K\left(A_j, C^{\mathrm{T}}\right) w_{ji} + e_{ji}^{(2)} b_{ji} \right\|^2 + \frac{c_{ji}}{2} \xi_{ji}^{\mathrm{T}} \xi_{ji} \\ S.t.\ \left(K\left(A_i, C^{\mathrm{T}}\right) w_{ji} + e_{ij}^{(1)} b_{ji}\right) + \xi_{ji} \geq e_{ij}^{(1)}, \xi_{ji} \geq 0 \end{cases} \tag{19}$$

MTWSVM must judge the classification after constructing all the classifiers. In the paper, the Majority Voting Method (MVM) is used to classify the samples in the experiment. MVM treats all classifications of samples as candidates and each sub-classifier of MTWSVM is a voter. Each voter has only one vote and can only vote for one candidate. MTWSVM contains $M\cdot(M\text{-}1)/2$ TWSVM two-classifiers are used to discriminate the classification sample x in the classification process. If classifier x is placed in the i-th class, the number of votes in the i-th class should be increased by 1. Otherwise, the number of votes for the j-th class is increased by 1. The sample x is classified as the class of the highest number votes after traversing all the classifiers.

B. Improved Particle Swarm Optimization

According to TWSVM algorithm principle, it can be seen that the kernel function parameter and the penalty factor have a vital influence on the final prediction effect and generalization ability of the model in the TWSVM classification process. Therefore, it is necessary to find the best parameters for the MTWSVM model to improve the predictive ability. PSO algorithm shows good optimization ability when solving the optimization function but the basic PSO algorithm is easy to trap into local optimization in the solving process. It leads to a large error in the result. The

Linear Decreasing Inertia Weight (LDIW) is introduced on the basis PSO algorithm in order to find the best parameters of the model and prevent the occurrence of local optimization [13]. After that, the algorithm's global search capability and the convergence rate can be effectively improved. LDIW is described below (formula 20):

$$w=[(w_s-w_e)(k_{max}-k)/k_{max}]+w_e \quad (20)$$

w_s is the maximum value of the initial inertia weight. w_e is the minimum value of the initial weight. k_{max} is the maximum iterations.

Intrusion Detection Method Based on IPSO-MTWSVM Model

The IPSO-MTWSVM model is built as final model after using the O-V-O classification strategy to construct the MTWSVM model. It's built by combining the IPSO algorithm and the MTWSVM algorithm. In order to introduce the network intrusion detection workflow of the IPSO-MTWSVM model in detail, the algorithm is described as Table I.

TABLE I
IPSO-MTWSVM MODEL

Input: Dataset X =KDD'99
Output: Output = (R1,R2,R3,R4,R5) // Classification results

```
s ← ADASYN(X)            // Oversampling
x← Numerical(s)          // Numerical
x' ← Z-score(x)          // Standardize
x" ← min-max(x')         // Normalize
init(maxgen,sizepop,pos,v)  // Initialize the IPSO parameters
While(i<maxgen*sizepop)
    f_i← fitness(x_i")
    individual_i ← calIndividual(f_i)
    global_i← calGlobal(f_i)
    refresh({pos_i,v_i})
    i ← i+1
End While
optimal ← save ({individual_i },{global_i})
Output ← MTWSVM (optimal)
return Output
```

IV. EXPERIMENTS

Data Preprocessing

i. *Sampling*

The 10% of KDD'99 training set are selected as the network intrusion detection dataset to detect the classification performance of the model. The ADASYN technology is used in the experiment to oversample the four types of attacks which contained too few samples in order to improve the practicability and prevent the problems caused by data imbalance. The number of samples has reached the requirements of random sampling after sampling. The experiment extracts 1000 entries from each of the four types of attack records after balanced treatment. The four types of attacks are Normal, DoS, Probe, U2R and R2L. The number of attack samples should be balanced in each type of attacks. The data subset is used in the final network intrusion detection experiments.

5000 records contain the four attack types and normal type samples in the experiment. Hence, experiments are multi-classification and need to be divided into DoS, Probe, U2R, R2L and normal. The detailed information of random sampling using ADASYN oversampling technology is shown in the Table II.

TABLE II
OVERSAMPLING AND RANDOM SAMPLING DETAILS

Types of Attacks		Oversampling		Sampling
		Before	After	
Normal		97278	97278	1000
DoS	back	2203	2203	167
	land	21	168	166
	neptune	107201	107201	167
	pod	264	264	166
	smurf	280790	280790	167
	teardop	979	979	167
Probe	ipsweep	1247	1247	250
	nmap	231	254	250
	portsweep	1040	1040	250
	satan	1589	1589	250
R2L	ftp_write	8	128	125
	guess_passwd	53	127	125
	imap	12	132	125
	multihop	7	126	125
	phf	4	128	125
	spy	2	126	125
	warezclient	1020	1020	125
	warezmaster	20	130	125
U2R	bufferoverflow	30	270	250
	loadmodule	9	252	250
	perl	3	252	250
	rookit	10	250	250

ii. *Digitization*

The classification model cannot identify the existence of the non-numerical data such as strings in the dataset. Therefore, these samples must be digitized to obtain data that can be used normally by the model.

iii. *Standardization*

The dataset needs to be standardized after digitization. Standardization mainly deal with the situation that the part of the feature vector is particularly scattered in the dataset. Standardization can prevent data gradient explosion and accelerate training. Z-score method is used for standardization. The method is introduced below (formula 21):

$$x'=(x-\mu)/\sigma \quad (21)$$

Among them, x is the original sample. x' is the new sample. μ is the mean of all samples. σ is the standard deviation of all samples. The standardization data obeys the standard normal distribution.

iv. *Normalization*

In order to further improve the accuracy and convergence speed of the model, the dataset needs to be normalized again. The *min-max* method is used for normalization. Then, the data is mapped to (0, 1). It is defined as follow (formula 22):

$$x''=(x'-min)/(max-min) \quad (22)$$

min and *max* are the minimum and maximum values of x', respectively. The dataset obtained by digitization. Standardization and normalization can fully meet the data type requirements of the model. They can be directly used to test the performance of the model in the paper.

Parameter Optimization

The dataset obtained after data preprocessing can be directly used in intrusion detection experiments. The model uses the best parameters so as to obtain the best performance. It is necessary to optimize the parameters through the IPSO algorithm before experiments.

The Mean-Square Error (MSE) is used as the objective function in experiments. The iterations of the IPSO algorithm and the population number are set to 50 and 20, respectively. The final iteration curve is shown in Fig.1.

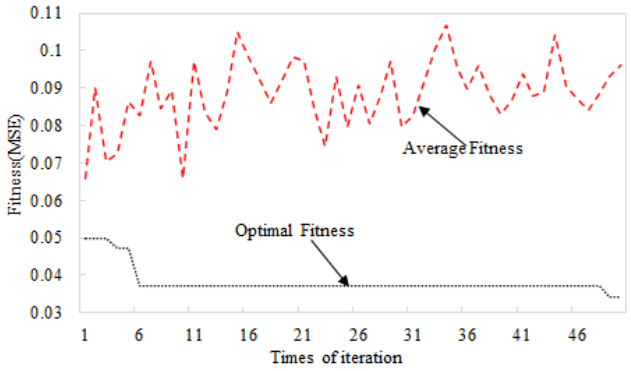

Fig.1. IPSO iteration diagram

The abscissa represents the iterations and the ordinate is the MSE value. The upper curve represents the average MSE value of each of iterations. The lower curve represents the optimal MSE value of each of iterations. The solution of MSE is used as follow (formula 23):

$$\mathrm{MSE}=\frac{1}{M}\sum_{m=1}^{M}\left(y_m-\widehat{y_m}\right)^2 \tag{23}$$

M represents the number of samples. y_m is the actual sample and $\widehat{y_m}$ is the predicted sample. It can be seen from formula (23) that the smaller the MSE value is, the smaller the error is. Therefore, it is necessary to find the parameters corresponding to the smallest MSE value through the iteration. It can be clearly seen from Fig.1 that the MSE value is the smallest at the forty-ninth iteration.

Result comparisons and discussion

In order to verify the effectiveness of the IPSO-MTWSVM model in network intrusion detection field, the experiments are used the same dataset. The SVM model is used for the comparative experiments with the default optimal parameters. The Table III is a confusion matrix. It is be used to compute the performance indicators.

TABLE III
CONFUSION MATRIX

Confusion matrix		Predicted class	
		Positive	Negative
Concrete class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Formulas (24)-(27) can be obtained by the confusion matrix shown in the Table III.

$$Accuracy=\frac{TP+TN}{TP+FP+TN+FN} \tag{24}$$

$$Precision=\frac{TP}{TP+FP} \tag{25}$$

$$Recall=\frac{TP}{TP+FN} \tag{26}$$

$$F\_measure=\frac{2*Precision*Recall}{Precision+Recall} \tag{27}$$

The optimal parameters obtained by IPSO algorithm are substituted into the MTWSVM model for network intrusion detection experiments. The dataset needs to be divided into the training set and the testing set during the experiments. Experiments show that the model has the best performance when the training set accounts for 70% and the testing set accounts for 30%. In this case, the training set has 3500 samples and the testing set has 1500 samples. The classification results of training set and testing set obtained through experiments are shown in Table IV and Table V.

TABLE IV
TRAINING SET CLASSIFICATION RESULT MATRIX

Concrete class / Predicted class	Normal	DoS	Probe	R2L	U2R
Normal	694	0	6	0	0
DoS	0	700	0	0	0
Probe	1	0	698	0	1
R2L	1	0	0	698	1
U2R	1	0	2	0	697

TABLE V
TESTING SET CLASSIFICATION RESULT MATRIX

Concrete class / Predicted class	Normal	DoS	Probe	R2L	U2R
Normal	297	0	2	1	0
DoS	0	300	0	0	0
Probe	4	0	296	0	0
R2L	1	0	0	298	1
U2R	1	0	0	0	299

The classification results matrices are shown as Table IV and Table V. They are obtained by IPSO-MTWSVM model using training set and testing set. The tables clearly show the number of correctly and incorrectly classified attacks for each type. The final performance indicators can be calculated from them by formulas (24)-(27). It can be clearly seen that the detection accuracy of R2L and U2R attack is increased by 1.18% and 4.19% when comparing IPO-MTWSVM with SVM in the Table VI. Meanwhile, the improved algorithm IPO-MTWSVM is also better than MTWSVM algorithm in the detection of four attacks in the Table VI. For the overall detection accuracy of the four attacks, IPO-MTWSVM algorithm is 0.73% higher than MTWSVM algorithm. The improved IPO-MTWSVM algorithm is 7.93% higher than SVM algorithm in Table VII. The experimental results prove that the optimal parameters of the IPSO algorithm are effective. The detection accuracy of various types of attacks has been significantly improved.

TABLE VI
IPSO-MTWSVM AND OTHER MODELS CLASSIFICATION PERFORMANCE

Model	Performance Index	Model Performance (%)				
		Normal	DoS	Probe	R2L	U2R
IPSO-MTWSVM	Accuracy	99.87	1	99.80	99.60	99.40
	Precision	99.6	1	99.7	99.3	98.1
	Recall	99.6	1	99.3	98.7	99
	F_measure	99.6	1	99.5	98.9	98.5
MTWSVM	Accuracy	99.53	99.73	99.79	98.99	99.13
	Precision	98.9	99.0	99.7	97.7	97.6
	Recall	98.6	99.7	99.3	97.3	98.0
	F_measure	98.8	99.3	99.5	97.5	97.8
SVM	Accuracy	96.21	95.61	96.75	98.42	95.21
	Precision	85.2	93.3	96.6	96.4	86.4
	Recall	98.3	84.8	88.2	96.4	89.7
	F_measure	91.3	88.8	92.2	96.4	88.0

The Table VI and Table VII are the comparison results of the network intrusion detection models built in the paper.

Next, we select the intrusion detection models of high-quality papers published in recent years as the comparison models. The same dataset is used in these models. Thus, FA-SVM, IE-DBN and DLELM are selected as the comparison models in the paper [14-16]. In order to more intuitively express the excellent detection accuracy of the models in the paper, the specific comparative analysis is shown in Fig.2.

TABLE VII
OVERALL ACCURACY OF THE INTRUSION DETECTION MODELS

Model	Overall Accuracy (%)
IPSO- MTWSVM	99.33
MTWSVM	98.6
SVM	91.4

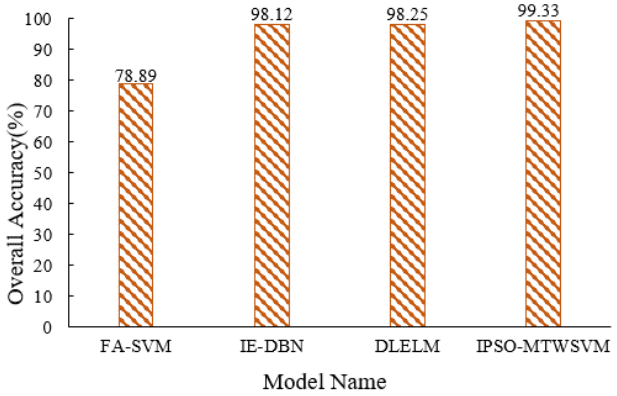

Fig.2. Overall accuracy comparison of the intrusion detection models

We can clearly draw the conclusion that the IPSO-MTWSVM network intrusion detection model is excellent through the comparison of Table VI, Table VII and Fig.2. Attacks can be analyzed and identified more effectively than other models in intrusion detection experiments.

V. CONCLUSION

In the paper, we propose an improved IPSO-MTWSVM algorithm for the network intrusion detection. The algorithm has better performance in accuracy, precision, recall and F1. It also shows the effectiveness and practicability of the model. The first applies ADASYN technology to balance the dataset in the data preprocessing stage. The number of U2R and R2L attacks is added to balance the number of attack samples. The second is numerical, standardized and normalized dataset so as to be suitable for the detection model processing. Data preprocessing can not only balance the samples to improve the practicability of the model, but also simplify the content of the samples and prevent data gradient explosions problem. The following is to use the O-V-O classification strategy on the basis of TWSVM. It makes the model achieve multi-classification function. The PSO algorithm for parameter optimization is introduced to build a PSO-MTWSVM model. Due to the PSO algorithm is easy to trap into the local optimization during the parameter optimization process, the linear decreasing inertia weight is introduced to improve the global search ability of the PSO algorithm and the model convergence speed. Finally, the final IPSO-MTWSVM model is built through parameter optimization. By comparing with the SVM and MTWSVM algorithm, the necessity and correctness of the improved IPSO-MTWSVM algorithm are verified. The proposed model is also better than those of new intrusion detection models in recent years. The proposed method improves the detection rate of U2R and R2L without reducing the detection rate of other attack types.

The following research work of the paper needs to study feature association rule mining or other feature engineering technologies so as to further improve the model detection performance. Shortening the calculation time of the model is still a problem worthy of further research. The proposed model used in the paper is relatively simple. Therefore, deep learning methods will be studied to build an efficient network intrusion detection model in future.

REFERENCES

[1] Laura Vegh, and Liviu Miclea, "A simple scheme for security and access control in cyber-physical systems," Proceedings of the 20th International Conference on Control Systems and Computer Science, 27-29 May, 2015, Bucharest, Romania, pp294-299.

[2] Eduardo Viegas, Altair Olivo Santin, and Vilmar Abreu Jr, "Machine Learning Intrusion Detection in Big Data Era:A Multi-Objective Approach for Longer Model Lifespans," IEEE Transactions on Network Science and Engineering, vol.8, no.1, pp366-376, 2021.

[3] Xiaobo Chen, and Yan Xiao, "Geometric projection twin support vector machine for pattern classification," Multimedia Tools and Applications, vol.80, no.15, pp23073-23089, 2021.

[4] Yuexuan An, Shifei Ding, and Jipu Hu, "Twin Support Vector Machine: A Review," Computer Science, vol.45, no.11, pp29-36, 2018.

[5] Jun He, and Shi-hui Zheng, "Intrusion detection model with twin support vector machines," Journal of Shanghai Jiao Tong University (Science), vol.19, no.4, pp448-454, 2014.

[6] Shengkai Zhong, "Intrusion detection based on improved least squares multi-classification twin support vector machine," Journal of Mathematical Sciences: Advances and Applications, vol.39, no.1, pp1-20, 2016.

[7] Jingcan Li, and Shifei Ding, "Twin support vector machine based on artificial fish swarm algorithm," CAAI Transactions on Intelligent Systems, vol.14, no.6, pp1121-1126, 2019.

[8] Guangyue Zhou, Kewen Li, Wenying Liu, and Zhaoxin Su, "Grey Wolf Optimizes Mixed Parameter Multi-Classification Twin Support Vector Machine," Journal of Frontiers of Computer Science and Technology, vol.14, no.4, pp628-636, 2020.

[9] The UCI KDD Archive Information and Computer Science University of California, Irvine. (1999, October, 28). KDD Cup 1999 Data. Available : http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[10] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," Proceedings of the International Joint Conference on Neural Networks, 1-8 June, 2008, Hong Kong, China, pp1322-1328.

[11] Shifei Ding, Jian Zhang, and Xiekai Zhang, "Survey on Multi Class Twin Support Vector Machines," Journal of Software, vol.29, no.1, pp89-108, 2018.

[12] Zhanfei Ma, hunian Chen, Jin Yang, Xuebao Li, and Qi Bian, "Novel Network Intrusion Detection Method Based on IPSO-SVM Algorithm," Computer Science, vol.45, no.2, pp231-235+260, 2018.

[13] Hindriyanto Dwi Purnomo, and Hui-Ming Wee, "Particle swarm optimization with adaptive selection of inertia weight strategy," International Journal of Computational Science and Engineering, vol.13, no.1, pp38-47, 2016.

[14] AI-Yaseen Wathiq Laftah, "Improving Intrusion Detection System by Developing Feature Selection Model Based on Firefly Algorithm and Support Vector Machine," IAENG International Journal of Computer Science, vol.46, no.4, pp534-540, 2019.

[15] Jia Huaping, Liu Jun, Zhang Min, He Xiaohu, and Sun Weixi, "Network intrusion detection based on IE-DBN model," Computer Communications, vol.178, pp131-140, 2021.

[16] Li Wuke, Yin Guangluan, and Chen Xiaoxiao, "Application of Deep Extreme Learning Machine in Network Intrusion Detection Systems," IAENG International Journal of Computer Science, vol.47, no.2, pp136-143, 2020.