IKPCA-ELM-based Intrusion Detection Method

Hui Wang[1*], Chengjie Wang[1], Zihao Shen[1] and Dengwei Lin[2]
[1] School of Computer Science and Technology, Henan Polytechnic University
Jiaozuo 454000, P.R. China
[e-mail: wanghui_jsj@hpu.edu.cn]
[2] Office of Educational Administration, Jiaozuo University
Jiaozuo 454000, P.R. China
[e-mail: 103000768@qq.com]
*Corresponding author: Hui Wang

Received December 25, 2019; revised February 13, 2020; accepted May 12, 2020; published July 31, 2020

Abstract

An IKPCA-ELM-based intrusion detection method is developed to address the problem of the low accuracy and slow speed of intrusion detection caused by redundancies and high dimensions of data in the network. First, in order to reduce the effects of uneven sample distribution and sample attribute differences on the extraction of KPCA features, the sample attribute mean and mean square error are introduced into the Gaussian radial basis function and polynomial kernel function respectively, and the two improved kernel functions are combined to construct a hybrid kernel function. Second, an improved particle swarm optimization (IPSO) algorithm is proposed to determine the optimal hybrid kernel function for improved kernel principal component analysis (IKPCA). Finally, IKPCA is conducted to complete feature extraction, and an extreme learning machine (ELM) is applied to classify common attack type detection. The experimental results demonstrate the effectiveness of the constructed hybrid kernel function. Compared with other intrusion detection methods, IKPCA-ELM not only ensures high accuracy rates, but also reduces the detection time and false alarm rate, especially reducing the false alarm rate of small sample attacks.

Keywords: Mixed kernel function, particle swarm optimization, kernel principal component analysis, extreme learning machine, intrusion detection

This work was supported by the National Natural Science Foundation of China (61300216).

1. Introduction

With the rapid development of the Internet, network intrusion is increasingly frequent. As a widely applied precaution, intrusion detection has become an important research topic [1]. In order to efficiently identify various types of attacks on networks, researchers have applied machine learning to intrusion detection. Common machine learning methods include Naïve Bayesian (NB) [2], back propagation (BP) [3], support vector machines (SVM) [4] and extreme learning machines (ELM) [5]. However, the intrusion detection of network data featuring high dimensions and more redundancies using a single machine learning method will result in reduced detection speed and accuracy. Therefore, how to combine methods of reducing network data dimensions and decreasing redundancy features with machine learning has become a hotspot of present research.

An intrusion detection method combining principal component analysis (PCA) with NB is proposed in Reference [6]. However, the accuracy rate of this intrusion detection method is not high because the conditional independence assumption of NB does not match the complexity relationship of data in the network.

In Reference [7], KPCA and improved BP neutral network are combined to detect network intrusion. However, the detection results of U2R and R2L attacks are not ideal since the neural network requires a large data sample set, and the available training dataset for U2R and R2L attacks is small.

An algorithm combining KPCA with a SVM is designed in Reference [8], but SVM learning is slow; that is, the intrusion detection time is affected if there is a large amount of data in the network.

In Reference [9], the authors point out that the ELM algorithm showed faster detection speed and better generalization performance than the SVM algorithm. In Reference [10], an intrusion detection method combining PCA and ELM is proposed . But it fails to obtain better results when PCA is used to reduce the dimensions of non-linear dependent data.

KPCA and ELM algorithms are proposed for intrusion detection in Reference [11]. However, due to the limitations of the Gaussian radial basis kernel function, the effect of KPCA feature extraction is often not ideal when there are huge differences in sample attributes or uneven sample distribution, so the detection performance of ELM is affected.

According to the above analysis, an IKPCA-ELM-based intrusion detection method is proposed in this paper. First, the mean and mean square error of sample attributes are introduced to improve the Gaussian radial basis kernel function and polynomial kernel function respectively, and the two improved kernel functions are combined to construct a hybrid kernel function in order to cover the shortages of KPCA feature extraction using a single kernel function in the event of uneven sample distribution and large differences in sample attributes. Next, an improved particle swarm optimization (IPSO) algorithm is proposed to determine the optimal hybrid kernel function, which is conducive to the improved KPCA combined with an ELM to enhance intrusion detection performance.

2. IKPCA-based Feature Extraction

2.1 Construction of Hybrid Kernel Functions

The KPCA algorithm is a non-linear extension based on PCA which makes up for the deficiency of PCA in reducing the dimensions of non-linear data. In KPCA, a kernel function is introduced to extract the features of non-linear data and reduce the data dimensions [12].The feature extraction concept of KPCA is to use mapping function Φ to map a to a certain feature space F, then utilize PCA to analyze the principal component in feature space F if training sample a is $a_1, a_2, \ldots, a_s$.

Assuming that mapping function Φ meets $\sum_{i=1}^{m}\Phi(a_i)=0$, that is, the mapping function has been demeaned, covariance matrix C^F and its corresponding feature equation are solved in feature space F:

$$\begin{cases} C^F = \frac{1}{s}\sum_{i=1}^{s}\Phi(a_i)\Phi(a_i)^T \\ \overline{\lambda v} = C^F \bar{v} \end{cases} \tag{1}$$

Where $\bar{\lambda} \geq 0$ is the feature value corresponding to the covariance matrix, and $\bar{v}$ is the feature vector corresponding to feature value $\bar{\lambda}$. The covariance matrix and its corresponding feature value and feature vector are difficult to obtain due to the challenging direct calculation of $\Phi(a_i)\Phi(a_i)^T$. Therefore, kernel function k is introduced to define $s \times s$-dimensional matrix K:

$$K = [k]_{s\times s} \tag{2}$$

Where $k = k(x,y) =< \Phi(x), \Phi(y) > = \Phi(x)\Phi(y)^T$. The calculation of the covariance matrix, feature value and feature vector according to mapping function Φ in equation (1) is thereby converted into the calculation of kernel matrix K according to the kernel function, and the feature value and feature vector are identified by the kernel matrix accordingly. Therefore, the KPCA feature extraction method is greatly dependent on the kernel function. Common kernel functions are given in **Table 1**:

Table 1. Description of Common Kernel Functions.

Kernel function name	Kernel function expression
Linear kernel function	$k(x,y) =< x,y >$
d-order polynomial kernel function	$k(x,y) = [a < x,y > +b]^d$
Gaussian radial basis kernel function (RBF)	$k(x,y) = \exp(-\frac{\|x-y\|^2}{\sigma^2})$
Multilayer perceptron kernel function	$k(x,y) = \tanh[v < x,y > +c]$

The selection of different kernel functions brings different effects to the KPCA feature extraction results [13]. The Gaussian radial basis kernel function is commonly used as a kernel function in KPCA feature extraction by virtue of its strong generalization and prediction capabilities. The Gaussian radial basis kernel function is a local kernel function. The kernel function value is only affected when the distance between samples is close, and the kernel function value gradually approaches 0 when the distance between samples is far. Therefore, KPCA using the Gaussian radial basis kernel function is not ideal for feature extraction if the

samples are unevenly distributed. The d-order polynomial kernel function is a global kernel function. The kernel function value will be affected even though the distance between samples is far. Therefore, according to Mercer's theorem of kernel functions, the Gaussian radial basis kernel function can be combined with the global d-order polynomial kernel function to construct a hybrid kernel function that can address the effects of uneven sample distribution on KPCA feature extraction. The equation is given as follows:

$$k_{hyb} = \varepsilon k_{RBF} + (1-\varepsilon)k_{poly} \tag{3}$$

Where k_{ploy} represents the d-order polynomial function, k_{RBF} represents the Gaussian radial basis kernel function and $\varepsilon \in (0,1)$ represents the kernel function weight ratio.

The sample record may contain dozens of attributes, and there may be significant differences in the same attribute in different samples. The dimension reduction effect of KPCA will be affected when the same attribute in different samples is significantly different. The hybrid kernel function constructed in equation (3) cannot solve this problem perfectly. Therefore, in order to diminish the effects of differences in sample attributes on feature extraction, attribute mean and mean square error are introduced to the Gaussian kernel function and polynomial kernel function to homogenize the attributes. The equations of the improved Gaussian kernel function and polynomial kernel function are given as follows:

$$k_{I-RBF}(x,y) = \exp(-\frac{\|(x-u)/p-(y-u)/p\|^2}{\sigma^2}) \tag{4}$$

$$k_{I-ploy}(x,y) = [a < (x-u/p),(y-u/p) > +b]^d \tag{5}$$

Where $u = (u_1, u_2, \ldots, u_m)$ is the attribute mean, $p = (p_1, p_2, \ldots, p_m)$ is the attribute mean square error and m is the sample vector dimension. The values of u_j and p_j are calculated as follows:

$$u_j = \frac{1}{s}\sum_{i=1}^{s} L_{ij} \tag{6}$$

$$p_j = \sqrt{\frac{1}{s-1}\sum_{i=1}^{s}(L_{ij}-u_j)^2} \tag{7}$$

Where j=1, 2, …, m; s is the number of training samples and L_{ij} is the jth attribute of the ith sample.

From the properties of Mercer's theorem, the improved function can be proven to be a kernel function. If K1 and K2 in Rn×Rn are kernel functions and the constant λ≥ 0, the following functions are kernel functions:

$$K(x,y) = \lambda K_1(x,y) \tag{8}$$

$$K(x,y) = K_1(x,y) \times K_2(x,y) \tag{9}$$

$$K(x,y) = \exp(K_1(x,y)) \tag{10}$$

Proposition 1: $k_{I-RBF}(x,y)$ is a kernel function.

Proof:

$$k_{I-RBF}(x,y) = \exp(-\frac{\|(x-u)/p-(y-u)/p\|^2}{\sigma^2})$$

$$= \exp(-\frac{\|(x-u)/p\|^2}{\sigma^2}) \times \exp(-\frac{\|(y-u)/p\|^2}{\sigma^2})$$

$$\times \exp(\frac{2[(x-u)/p][(y-u)/p]}{\sigma^2})$$

Let $f(x)=\exp(-\frac{\|(x-u)/p\|^2}{\sigma^2})$ and $h(x,y)=\exp(\frac{2[(x-u)/p][(y-u)/p]}{\sigma^2})$,

then $k_{I-RBF}(x,y)=f(x)f(y)h(x,y)$.

According to Reference [14], $\psi(x)\psi(y)$ are positive definite kernels if function ψ in Rn is a real function. Therefore, $f(x)f(y)$ are positive definite kernels.

Let $\Omega(x)=(x-u)/p$ and $\Omega(y)=(y-u)/p$, then similarly $\Omega(x)\Omega(y)$ are positive definite kernels.

From equation (8), $\frac{2}{\sigma^2}\Omega(x)\Omega(y)$ is a positive definite kernel, and further, from equation (10), $h(x,y)$ is a positive definite kernel.

Since $f(x)f(y)$ and $h(x,y)$ are positive definite kernels, $k_{I-RBF}(x,y)$ is a kernel function from equation (9).

Proposition 2: $k_{I-ploy}(x,y)$ is a kernel function.

Proof: The dot product of vectors x,y is a kernel function if the vectors $x\in R^n, y\in R^n$ according to the properties of Mercer's theorem. Just like $(x-u/p)\in R^n, (y-u/p)\in R^n$, $k_{I-ploy}(x,y)$ is a kernel function.

In this paper, in order to reduce the effects of uneven sample distribution and sample attribute difference on KPCA feature extraction, and further improve the KPCA feature extraction results, the improved Gaussian radial base kernel function k_{I-RBF} is combined with the improved d-order polynomial kernel function k_{I-ploy} to construct a new hybrid kernel function, as shown in the equation below:

$$k_{N-hyb}=ck_{I-RBF}+(1-c)k_{I-poly} \tag{11}$$

Where $c\in(0,1)$ represents the kernel function weight ratio.

2.2 IKPCA Feature Extraction Process

The hybrid kernel function constructed in 1.1 is used to improve KPCA, and the IKPCA feature extraction process is described as follows:

1) A $s\times m$-order data matrix can be obtained given that there are s sample instances and each sample has m dimensional features:

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1\mathrm{m}} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{s1} & a_{s2} & \cdots & a_{sm} \end{bmatrix} \tag{12}$$

2) Kernel matrix K is calculated by equation (2) according to the hybrid kernel function.

3) Kernel matrix K is decentralized using equation $\overline{K}=K-l_sK-Kl_s+l_sKl_s$ to obtain decentralized kernel matrix $\overline{K}$, where each value of matrix l_s is $1/s$.

4) Feature values $\lambda_1,\lambda_2,\ldots,\lambda_s$ and feature vectors $v_1,v_2,\ldots,v_s$ of $\overline{K}$ are calculated using $\lambda v=\overline{K}v$.

5) Feature values $\lambda_1,\lambda_2,\ldots,\lambda_s$ are ranked in descending order to obtain feature value sequence $\lambda_1^{'},\lambda_2^{'},\ldots,\lambda_s^{'}$, and the feature vectors are adjusted to $v_1^{'},v_2^{'},\ldots,v_s^{'}$ accordingly.

6) Feature vectors $v_1^{'}, v_2^{'}, \ldots, v_s^{'}$ are subject to Schmidt orthogonalization to obtain unitized orthogonal feature vectors $\alpha_1, \alpha_2, \ldots, \alpha_s$.

7) Accumulative contribution rates $\eta_1, \eta_2, \cdots, \eta_s$ corresponding to feature values $\lambda_1^{'}, \lambda_2^{'}, \cdots, \lambda_s^{'}$ are calculated using $\frac{\sum_{i=1}^{k} \lambda_i^{'}}{\sum_{j=1}^{m} \lambda_j^{'}}$ and compared with predetermined accumulative contribution rate P that shall be at least 90% in this paper. If $\eta_m \geq P$, then m principal components $\alpha_1, \alpha_2, \cdots, \alpha_m$ are extracted.

8) Projection matrix $A^{'} = \overline{K}\alpha$ of kernel matrix $\overline{K}$ on the extracted feature vectors is calculated, where $\alpha = \{\alpha_1, \alpha_1, \ldots, \alpha_m\}$, and matrix $A^{'}$ represents the data upon IKPCA feature extraction.

3. Determination of Hybrid Kernel Function Parameters

3.1 Improved Particle Swarm Optimization (IPSO)

Particle Swarm Optimization (PSO) is a heuristic optimization algorithm based on random strategies. Known for its powerful global search capability and fast convergence rate, PSO has been widely used in parameter optimization [15].

In the PSO algorithm, if the number of iterations is t and the relevant information of particle i is described by particle position x_i^t and particle velocity v_i^t, then the position of particle t in dimension d is x_{id}^t, and the particle flight velocity is v_{id}^t. During the iteration process, the particle position is updated by particle velocity, and particle velocity is updated by the individual and global optimal values, so that the common position and velocity update equations are given as follows:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{13}$$

$$\begin{aligned} v_{id}^{t+1} = {} & w_i v_{id}^t + c_1 rand_1 (pbest_{id}^t - x_{id}^t) \\ & + c_2 rand_2 (gbest_d^t - x_{id}^t) \end{aligned} \tag{14}$$

Where c_1 and c_2 represent the individual and global learning rates of the particle respectively, $rand_1$ and $rand_2$ are random numbers between 0 and 1, $t_{\max}$ represents the maximum number of iterations, $pbest_{id}^t$ represents the individual optimal value of particle i after t iterations, and $gbest_d^t$ represents the global optimal value of the particle swarm after t iterations. w_i represents the inertia factor that affects particle velocity.

The value of inertia factor w_i has a critical impact on the convergence rate, so the selection of an appropriate inertia factor is of great significance [16]. In Reference [17], inertia factor w_i is studied and the PSO algorithm is found to have a faster convergence rate and better effect when a concave-convex function is used to decrease and quantify the inertia factor. Therefore, the inertia factor expression adopted in this paper is:

$$w_i = (w_{\max} - w_{\min})(1 - \frac{t}{t_{\max}})^3 + w_{\min} \tag{15}$$

Where $w_{\min}$ and $w_{\max}$ represent the minimum and maximum values of inertia factor w_i respectively.

At the same time, in order to avoid the premature convergence of the particle swarm in local areas and fall into local optimization, the diversity of the particle swarm is introduced in this paper and calculated as follows:

$$diversity = \frac{1}{L \times N} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{N} (x_{ij} - \overline{x_j})^2} \tag{16}$$

Where L is the maximum value of the diagonal of the search space, N represents the size of the particle swarm, x_{ij} represents the j component at the position of the i th particle, and $\overline{x_j}$ is the mean of the j th component of the particle.

The *diversity* value of the particle swarm is compared with predetermined threshold h to check that the particle falls into local optimization. Due to the capability of escaping local optimization, Gaussian disturbance is used in this paper to update the global optimal value when the particle is trapped in local optimization. The equations are given as follows:

$$\begin{cases} gbest_d^t = gbest_d^t + (x_{\max}^t - x_{\min}^t) \bullet Gaussian(0, \delta^2) \\ \delta = \delta_{\max} - (\delta_{\max} - \delta_{\min}) \times \dfrac{t}{t_{\max}} \end{cases} \tag{17}$$

Where $x_{\max}^t$ and $x_{\min}^t$ are the maximum and minimum values of all current particle positions, and $\delta_{\max}$ and $\delta_{\min}$ are the upper and lower limits of Gaussian distribution parameter δ.

3.2 Determination of Hybrid Kernel Function Parameters Based on IPSO

The selection of different parameters of the hybrid kernel function will affect feature extraction, so the IPSO algorithm proposed in 2.1 is used here to identify the optimal values of parameters in order to avoid randomly setting parameters c, a, b, d and σ of the hybrid kernel function constructed in 1.1. In the solution for the parameters of the hybrid kernel function, each particle is composed of (c, a, b, d and σ) to represent a potential solution. Parameter optimization is measured by a fitness function. The optimal parameters of the hybrid kernel function are determined so as to improve intrusion detection accuracy. Upon feature extraction, classification detection is performed using an ELM. Therefore, the error between the true value and predicted value of the test sample is taken as the fitness function:

$$f = \frac{\sum_{j=1}^{s} (y_j - \widehat{y_j})^2}{s} \tag{18}$$

Where s and y_j are the test sample size and true value respectively, and $\widehat{y_j}$ is the predicted value of the ELM. During the iteration process, the mean square error is minimized to identify the optimal parameters.

The specific procedures are described as follows:

1) Parameter initialization: There is a particle swarm with size N, maximum iterations $t_{\max}$, inertia factor $[w_{\min}, w_{\max}]$, particle learning rate c_1, c_2, particle swarm velocity $[v_{\min}, v_{\max}]$ and position $[x_{\min}, x_{\max}]$, diversity threshold h and Gaussian distribution parameters $[\delta_{\min}, \delta_{\max}]$. The particle velocity and position are set randomly within the range of $[v_{\min}, v_{\max}]$ and $[x_{\min}, x_{\max}]$.

2) For all particles, the fitness value is calculated according to equation (18) and used as the individual optimal value $pbest_{id}^{t}$ of these particles, and the minimum fitness value is used as the global optimal value $gbest_{d}^{t}$.

3) The position and velocity of particle i are updated according to equation (13) and equations (14) and (15) individually.

4) The *diversity* value of the particle swarm is calculated using equation (16). Step 5 is executed if the value is less than h; otherwise, step 6 is executed.

5) The global optimal value $gbest_{d}^{t}$ is updated using equation (17), and the fitness value of the particle is calculated using equation (18). The current individual and global optimal values of the particle are compared respectively with the historical individual and global optimal values to select the smaller values.

6) The number of iterations t is judged to reach the maximum $t_{\max}$. If not, steps 3 to 6 are repeated; otherwise, the algorithm ends.

7) The global optimal values of parameters c, a, b, d and σ of the hybrid kernel function are obtained.

4. An ELM Classifier for Intrusion Detection

ELM classifiers have been widely applied in classification, pattern recognition and other fields due to their excellent generalization performance and fast detection speed. In this paper, when an ELM is used for intrusion detection, IKPCA is conducted first to complete the feature extraction of the training dataset and reduce the data dimensions; then the extracted dataset $A^{'}=\{a_i, i=1,2,...,Q\}$ is used to train the ELM classifier; and finally, the test dataset is applied to check the ELM classification performance after training the ELM classifier. The specific procedures of using an ELM classifier for intrusion detection are as follows:

1) Threshold b of the ELM hidden layer and connection weight θ between the input layer and hidden layer are initialized randomly.

2) Activation function G is selected and the hidden layer is assumed to have L neurons.

3) If L is equal to the number of samples in training set $A^{'}$, the ELM can approach the training samples with zero error for any b and θ; and if the number Q of samples in the training set is large, the training error may approach any number meeting $\varepsilon>0$, and the output matrix of the ELM is given as follows:

$$H\beta=T \tag{19}$$

Where T represents the expected output matrix of the ELM, β is the connection weight between the hidden layer and output layer, and H is the output matrix of the hidden layer of the ELM network, expressed as follows:

$$H(\theta_1,...,\theta_L,b_1,...,b_L,a_1,...,a_Q)=\begin{bmatrix} G(\theta_1 a_1+b_1) & \cdots & G(\theta_L a_1+b_L) \\ \vdots & & \vdots \\ G(\theta_1 a_Q+b_1) & \cdots & G(\theta_L a_Q+b_L) \end{bmatrix} \tag{20}$$

4) Equation $\beta = H^{+}T$ is solved according to $\min_{\beta} \|H\beta - T\|$, where H^{+} is the generalized inverse of the *Moore – Penrose* of output matrix H. The values of β, b and θ are determined by following steps 1 to 4 to complete the training of the ELM classifier [18].

5) The test dataset is used to check the classification performance of the trained ELM.

5. Intrusion Detection Model

Intrusion detection is essentially a classification issue that can distinguish between normal and abnormal data. Most intrusion data in the network has relatively high dimensions, so IKPCA is conducted to extract the principal component and reduce the data dimensions, and an ELM classifier is applied for intrusion detection. The proposed model consists of three stages. In the first stage, the principal component is obtained based on IKPCA, irrelevant and redundant attributes are deleted, and the optimal attribute subset is found. In the second stage, the optimal attribute subset is used as the training dataset of the ELM. In the third stage, the ELM is used to classify the test dataset, and the performance of the model is evaluated in conjunction with the evaluation indicators. The intrusion detection model process is illustrated in **Fig. 1**:

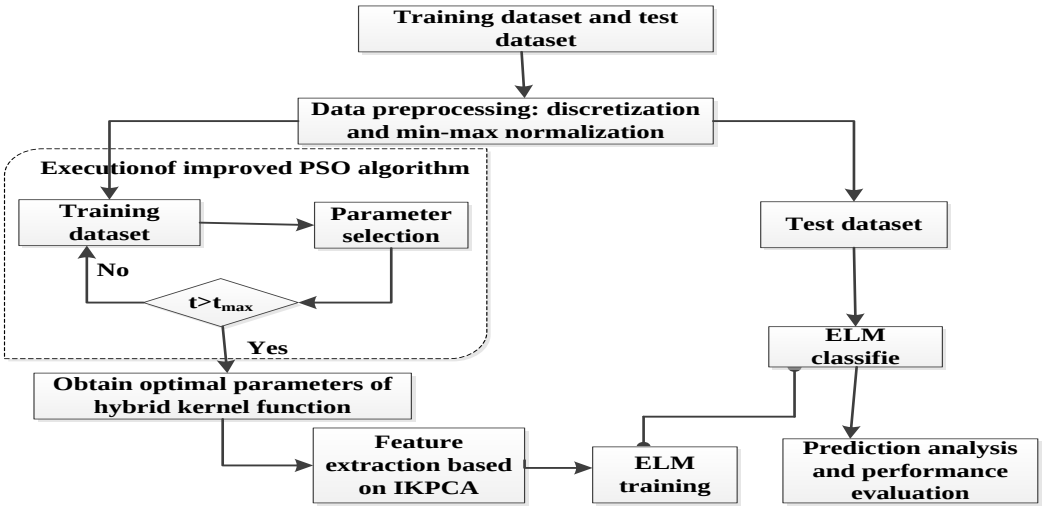

Fig. 1. IKPCA-ELM-based Intrusion Detection Model

6. Simulation Experiment

6.1 IPSO Verification

In order to verify the effectiveness of the IPSO algorithm proposed in this paper compared with the PSO algorithm in parameter optimization, a single-peak benchmark function Sphere and a multi-peak benchmark function Rastrigin are selected for testing. The function information is given in **Table 2**:

Table 2. Test function information.

Function name	Sphere	Rastrigin
Test function expression	$F_1(x)=\sum_{i=1}^{n} x_i^2$	$F_2(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$
Dimensions	60	60
Search range	[-100, 100]	[-5.12, 5.12]
Minimum value	0	0

For fairness, the basic parameter settings of the PSO and IPSO algorithms are the same, as follows: the size of particle swarm N = 60, maximum number of iterations $t_{\max}=200$, inertia factors $w_{\max}=0.9$ and $w_{\min}=0.4$, particle learning rate $c_1=c_2=2$ and maximum value 1 and minimum value -1 of particle swarm velocity and position. The IPSO algorithm has the diversity threshold $h=0.0001$ and Gaussian distribution parameters $\delta_{\max}=0.6$ and $\delta_{\min}=0.1$. The performance of the PSO and IPSO algorithms is compared in **Table 3**:

Table 3. Performance Information of Algorithms.

Function name	PSO		IPSO	
	Mean	Standard deviation	Mean	Standard deviation
Sphere	0.035642	0.075068	2.6278620×10^{-7}	4.3623930×10^{-7}
Rastrigin	0.985120	0.878032	-0.008934	0.257166

In **Table 3**, the mean and standard deviation of the IPSO algorithm are less than those of the PSO algorithm, indicating that IPSO searches more accurately and steadily in optimization.

For the two test functions, the relationship between the fitness value and number of iterations of PSO and IPSO is illustrated in **Fig. 2** and **Fig. 3**, in which IPSO shows a faster convergence rate and better convergence precision than PSO, thus the optimization performance of IPSO is better.

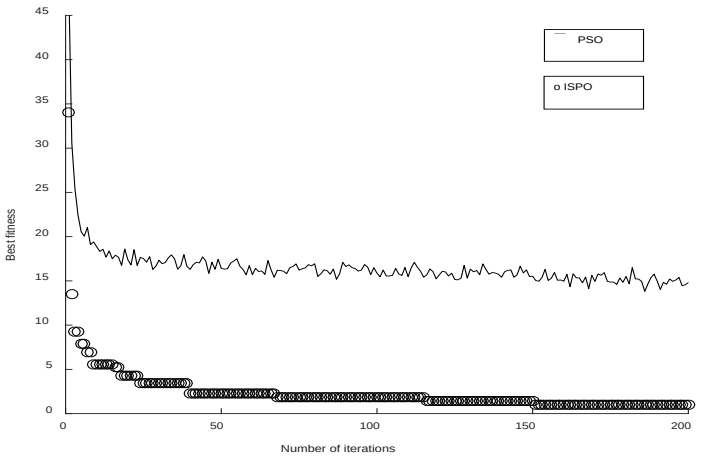

Fig. 2. Sphere function

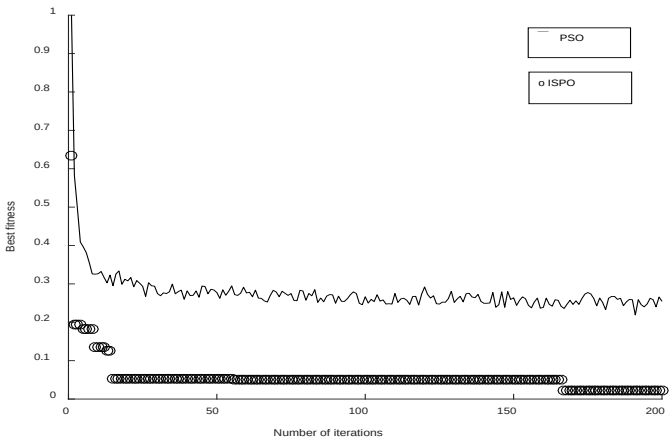

Fig. 3. Rastrigin function

6.2 Experimental Data and Environmental Platform

In order to check the performance of the IKPCA-ELM method proposed in this paper, 10% of the training dataset and test dataset in the KDD CUP 1999 dataset is used as the experimental data for the simulation experiment. In addition, an experimental environment platform is constructed with the operating system of Windows 10, CPU frequency of 2.60 GHz, memory of 4G, programming language of Python3.6 and development environment of Eclipse+PyDev.

6.3 Experimental Evaluation Criteria

In the performance check of the IKPCA-ELM method proposed in this paper, the detection rate (DR), accuracy rate (AR), false alarm rate (FAR) and detection time (TC) are used as evaluation indicators and quantified as follows:

$$Detection\ rate = \frac{Number\ of\ detected\ intrusion\ samples}{Total\ number\ of\ intrusion\ samples} \times 100\% \tag{21}$$

$$Accuracy\ rate = \frac{Number\ of\ normally\ classified\ samples}{Total\ number\ of\ test\ samples} \times 100\% \tag{22}$$

$$False\ alarm\ rate = \frac{Number\ of\ normal\ samples\ mistaken\ as\ intrusion\ samples}{Total\ number\ of\ normal\ samples} \times 100\% \tag{23}$$

6.4 Analysis of Experimental Results

Two groups of experiments are designed in this paper:

1.The effects on ELM detection performance after different kernel functions are used in KPCA to reduce dimensions in order to verify the performance of the hybrid kernel function constructed in this paper;

2. The comparison of the IKPCA-ELM method in this paper with other intrusion detection methods in terms of detection performance.

6.4.1 Performance Comparison of Different Kernel Functions

To ensure the objectivity of the experiments, a total of 4 datasets are constructed respectively with the data extracted randomly from the training set and the test set, as shown in **Table 4**:

Table 4. Selection of Data from Training Set and Test Set

Category	Data description	DS1	DS2	DS3	DS4
Training set	Normal	9500	7300	6500	11005
	Abnormal	3203	4576	1200	2589
	Total	12703	11579	7700	13594
Test set	Normal	8909	6784	8765	9760
	Abnormal	1700	4534	1235	1890
	Total	10609	11318	10000	11650

The parameters of various kernel functions (polynomial kernel function k_{ploy}, traditional Gaussian radial basis kernel function k_{RBF}, hybrid kernel function of the traditional Gaussian radial basis kernel function and polynomial kernel function $k_{ploy} + k_{RBF}$, and the hybrid kernel function constructed in this paper $k_{I-ploy} + k_{I-RBF}$) are addressed respectively by the IPSO algorithm through 30 simulation experiments, as shown in **Table 5**. For different kernel functions, KPCA is conducted for feature extraction, and the classification results of the ELM on the 4 datasets are then utilized to check the performance of each kernel function, in which the number of neutrons in the ELM hidden layer is 200. The results are shown in **Table 6**.

Table 5. Kernel Function Parameters

Kernel Function name	c	a	b	d	σ
k_{ploy}	---	3.4527	1	1	---
k_{RBF}	---	---	---	---	8.7326
$k_{ploy} + k_{RBF}$	0.7764	2.7813	3	1	8.1549
$k_{I-ploy} + k_{I-RBF}$	0.4764	2.4506	2	1	9.0586

Table 6. Performance Comparison of Kernel Functions

S/N	KPCA(Kernels)-ELM	DR(%)	AR(%)	FAR(%)	TC(s)
DS1	k_{ploy}	91.5	95.6	3.45	1.03
	k_{RBF}	95.13	95.6	3.32	2.05
	$k_{ploy} + k_{RBF}$	96.12	95.9	2.56	1.67
	$k_{I-ploy} + k_{I-RBF}$	98.93	98.13	1.32	0.96
DS2	k_{ploy}	88.4	90.25	3.35	1.29
	k_{RBF}	95.25	95.7	3.27	2.41
	$k_{ploy} + k_{RBF}$	95.45	95.1	2.89	1.97
	$k_{I-ploy} + k_{I-RBF}$	98.78	98.16	1.27	1.02
DS3	k_{ploy}	89.3	90.13	3.01	1.41
	k_{RBF}	95.34	95.9	2.97	2.99

	$k_{ploy}+k_{RBF}$	95.92	96.21	2.46	2.39
	$k_{I-ploy}+k_{I-RBF}$	97.78	99.01	0.97	1.25
DS4	k_{ploy}	92.1	91.42	2.68	0.97
	k_{RBF}	96.97	97.8	2.15	1.93
	$k_{ploy}+k_{RBF}$	97.12	98.19	1.97	1.67
	$k_{I-ploy}+k_{I-RBF}$	97.93	98.89	1.15	0.95

From **Table 6**, after the hybrid kernel function constructed in this paper is used to reduce the dimensions in KPCA, on average, the DR of the ELM increases by 3.26%, AR increases by 2.07%, FAR decreases by 1.59% and TC decreases by 0.76s. The reasons include: 1) The attribute homogenization of the kernel function reduces the effects of attribute differences on feature extraction; and 2) The combination of local kernel functions and global kernel functions diminishes the effects of uneven sample distribution on feature extraction. As reasons 1) and 2) improve the feature extraction results, the detection performance of the ELM is enhanced accordingly.

According to the above analysis, the ELM shows acceptable detection performance when the hybrid kernel function constructed in this paper is used to reduce the dimensions in KPCA, which further illustrates the effectiveness of the hybrid kernel function constructed in this paper.

Different attack types are not detected separately in the above experiments. In order to further analyze the detection performance of the hybrid kernel function constructed in this paper for different attack types, 10,000 entries of data are extracted randomly from the training dataset and test dataset respectively. The data of each attack type is given in **Table 7**. Taking DR as an example, the experimental results are shown in **Table 8**.

Table 7. Data Distribution in Training Set and Test Set by Attack Type

Attack type	Training dataset	Test dataset
Normal	4548	3469
Probe	3000	3300
Dos	1700	2000
U2R	52	228
R2L	700	1003

Table 8. Comparison of DR of Kernel Functions

KPCA(Kernels)-ELM	Normal	Dos	Probe	R2L	U2R
k_{ploy}	93.31	90.07	91.69	89.77	25.23
k_{RBF}	94.29	91.06	89.58	90.34	36.54
$k_{ploy}+k_{RBF}$	96.93	93.12	91.85	91. 86	40.67
$k_{I-ploy}+k_{I-RBF}$	98.31	98.06	98.69	95.77	44.23

From **Table 8**, after the hybrid kernel function constructed in this paper is used to reduce the dimensions in KPCA, the ELM DR of Normal, Dos, Probe, R2L and U2R are 98.31%, 98.06%, 98.69%, 95.77% and 44.23% respectively. Compared with other kernel functions, the DR increases by 3.74% on average, and thus the effectiveness of the hybrid kernel function constructed in this paper is demonstrated again.

6.4.2 Comparison Between IKPCA-ELM and Other Methods

In order to check the detection performance of the IKPCA-ELM method proposed in this paper against different attack types, the method is compared with KPCA-SVM [8], KPCA-BP [7], PCA-ELM [10], KPCA-ELM [11], and ELM respectively in terms of AR, FAR and TC. The experimental results are given in **Table 9** and **Fig. 4** and **Fig. 5** respectively.

Table 9. Comparison of AR (%) of Different Classifiers

Classifier	Normal	Dos	Probe	R2L	U2R
IKPCA-ELM	98.31	98.07	98.79	95.77	44.23
KPCA-ELM	97.29	98.06	97.58	94.34	36.54
PCA-ELM	95.93	92.12	91.85	91.68	23.67
ELM	92.24	90.76	89.98	88.76	20.46
KPCA-SVM	95.26	93.45	94.08	93.18	26.78
KPCA-BP	98.81	98.89	97.95	91.35	22.15

From **Table 9**, (1) the AR of the ELM classifier using PCA, KPCA and IKPCA for feature extraction increases compared with ELM classifiers without feature extraction during intrusion detection. The results also show that the AR of IKPCA-ELM increases by 5.03% on average compared with PCA-ELM and KPCA-ELM. This is caused by the application of the hybrid kernel function used in IKPCA, which provides the powerful generalization capability of global kernel functions and the strong learning capability of local kernel functions that can provide more distinguishing information in data processing, thereby improving the generalization performance of the ELM classifier. (2) Compared with KPCA-SVM and KPCA-BP, the accuracy rate increases significantly when IKPCA-ELM is used for intrusion detection against the attacks of Probe, R2L and U2R.

From **Fig. 4** and **Fig. 5**, the FAR and TC of IKPCA-ELM decline compared with other intrusion detection methods; in particular, FAR is especially low against R2L and U2R attacks.

The experimental results of the above evaluation indicators reveal that the IKPCA-ELM method guarantees a high AR, while FAR and TC are decreased compared with other methods.

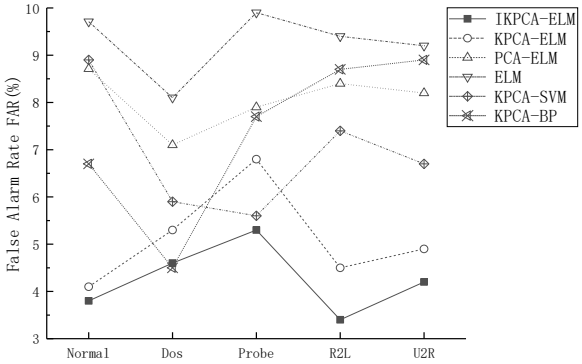

Fig. 4. Comparison of FAR of Different Classifiers.

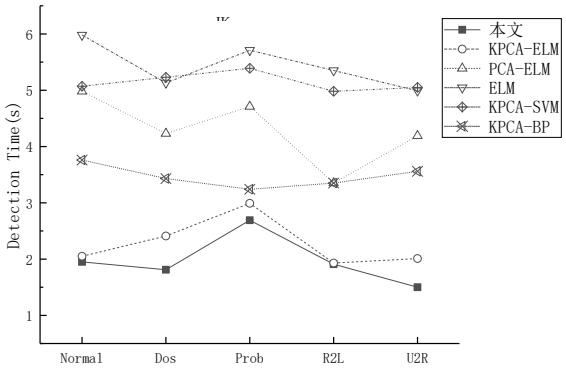

Fig. 5. Comparison of TC of Different Classifiers.

7. Conclusion

How to improve the performance of intrusion detection techniques has long been a hot topic at home and abroad. To this end, an intrusion detection method based on IKPCA-ELM is proposed in this paper. First, the sample attribute mean and mean square error are introduced into the Gaussian radial basis kernel function and polynomial kernel function, and the two improved kernel functions are combined to construct a hybrid kernel function that can make up for the shortcomings of ignoring sample attribute differences and uneven sample distribution during the feature extraction of KPCA. Second, an improved particle swarm algorithm is used to further optimize KPCA. Finally, the improved KPCA is conducted to complete the feature extraction of the datasets, and an ELM is used for intrusion detection. The experimental results demonstrate the effectiveness of the hybrid kernel function proposed in this paper. Compared with such intrusion detection methods as ELM, PCA-ELM, KPCA-ELM, KPCA-SVM and KPCA-BP, IKPCA-ELM shows a higher AR against all attack types, with reduced FAR and TC. The ELM will be further optimized in the future to strengthen the robustness of intrusion detection algorithms.

References

[1] W. Elmasry, A. Akbulut, A. H. Zaim, "Empirical study on multiclass classification-based network intrusion detection," *Computational Intelligence*, vol.35, no.4, pp.919-954, November, 2019. Article (CrossRef Link)

[2] J. Jiang, D. Wu, Y. Chen, et al, "Fast artificial bee colony algorithm with complex network and naive bayes classifier for supply chain network management," *Soft Computing*, vol.23, no.24, pp. 13321-13337, December, 2019. Article (CrossRef Link)

[3] L. Duan, D. Han, Q. Tian, "Design of intrusion detection system based on improved ABC_elite and BP neural networks," *Computer Science and Information Systems*, vol. 16, no. 3, pp. 773-795, October, 2019. Article (CrossRef Link)

[4] S. Moslemnejad, J. Hamidzadeh, "A hybrid method for increasing the speed of SVM training using belief function theory and boundary region," *International Journal of Machine Learning and Cybernetics*, vol.10, no.12, pp.3557-3574, December, 2019. Article (CrossRef Link)

[5] C. R. Wang, R. F. Xu, S. J. Lee, et al., "Network intrusion detection using equalit constrained-optimization-based extreme learning machines," *Knowledge-Based Systems*, vol. 147, no. 1, pp. 68-80, 2018. Article (CrossRef Link)

[6] S. A. Adedigba, F. Khan, M. Yang, "Dynamic failure analysis of process systems using principal component analysis and Bayesian network," *Industrial & Engineering Chemistry Research*, vol. 56, no. 8, pp. 2094-2106, March, 2017. Article (CrossRef Link)

[7] A. Yang, Y. Zhuansun, C. Liu, et al., "Design of Intrusion Detection System for Internet of Things Based on Improved BP Neural Network," *IEEE Access*, vol. 7, no. 8, pp. 106043-106052, 2019. Article (CrossRef Link)

[8] W. Wang, M. Zhang, D. Wang, et al., "Kernel PCA feature extraction and the SVM classification algorithm for multiple-status, through-wall, human being detection," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, pp. 151, September 6, 2017. Article (CrossRef Link)

[9] Z. Chen, K. Gryllias, W. Li, "Mechanical fault diagnosis using Convolutional Neural Networks and Extreme Learning Machine," *Mechanical Systems and Signal Processing*, vol. 133, pp. 106272, November 1, 2019. Article (CrossRef Link)

[10] A. P. Kale, S. Sonavane, "PF-FELM: A Robust PCA Feature Selection for Fuzzy Extreme Learning Machine," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1303-1312, December, 2018. Article (CrossRef Link)

[11] W. Deng, B. Ye, J. Bao, et al., "Classification and Quantitative Evaluation of Eddy Current Based on Kernel-PCA and ELM for Defects in Metal Component," *Metals*, vol.9, no.2, pp.155, February, 2019. Article (CrossRef Link)

[12] S. Velliangiri, "A hybrid BGWO with KPCA for intrusion detection," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 32, no. 1, pp.165-180, July 31, 2019. Article (CrossRef Link)

[13] A. Al-Amro, M. Hussain, I. Ahmad, et al., "Performance Comparison of Kernel Principal Component Analysis's Kernels in Intrusion Detection," *Journal of Medical Imaging and Health Informatics*, vol. 7, no.3, pp. 733-738, June, 2017. Article (CrossRef Link)

[14] A. Tharwat, "Parameter investigation of support vector machine classifier with kernel functions," *Knowledge and Information Systems*, vol.61, no.3, pp.1269-1302, December, 2019. Article (CrossRef Link)

[15] B. Tran, B. Xue, M. Zhang, "A new representation in PSO for discretization-based feature selection," *IEEE Transactions on Cybernetics*, vol.48, no.6, pp.1733-1746, June, 2018. Article (CrossRef Link)

[16] C. Scheepers, A. P. Engelbrecht, C. W. Cleghorn, "Multi-guide particle swarm optimization for multi-objective optimization: empirical and stability analysis," *Swarm Intelligence*, vol.13, no.3-4, pp. 245-276, December, 2019. Article (CrossRef Link)

[17] M. Li, H. Chen, X. Shi, et al., "A multi-information fusion "triple variables with iteration" inertia weight PSO algorithm and its application," *Applied Soft Computing*, vol.84, pp. 105677, November, 2019. Article (CrossRef Link)
[18] L. Zhao, J. Zhu, "Learning from correlation with extreme learning machine," *International Journal of Machine Learning and Cybernetics*, vol.10, no.12, pp. 3635-3645, December, 2019. Article (CrossRef Link)

Hui Wang is a professor at the School of Computer Science and Technology in Henan Polytechnic University, Jiaozuo, Henan, China. He received his PhD degree from Jilin University, Changchun, Jilin, China in 2009. His current research interests include network security, artificial intelligence, etc.

Chengjie Wang is currently pursuing a master degree at the School of Computer Science and Technology in Henan Polytechnic University, Jiaozuo, Henan, China. His current research interests include network security, artificial intelligence, etc.

Zihao Shen is a teacher at the School of Computer Science and Technology in Henan Polytechnic University, Jiaozuo, Henan, China. He is currently pursuing a doctoral degree at the School of Computer Science and Technology in Jilin University, Changchun, Jilin, China. His current research interests include network security, artificial intelligence, etc.

Dengwei Lin received his M.S. degree from East China University of Science and Technology, Shanghai, China in 2008. He is currently a professor in Jiaozuo University, Jiaozuo, China. His current research interests include computer control, software development, artificial intelligence, etc.