1. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection

Accession number: IP53205963
Authors: Kuang, Fangjun (1, 2); Zhang, Siyang (2); Jin, Zhong (1); Xu, Weihong (1, 3)
Author affiliation: (1) School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China; (2) Department of Electrical and Information Engineering, Hunan Vocational Institute of Safety and Technology, Changsha, 410151, China; (3) College of Computer and Communications Engineering, Changsha University of Science and Technology, Changsha, 410077, China
Corresponding author: Xu, W.(xwhxdcs@126.com)
Source title: Soft Computing
Abbreviated source title: Soft Comput.
Issue date: June 22, 2014
Publication year: 2014
Language: English
ISSN: 14327643
E-ISSN: 14337479
Document type: Article in Press
Abstract: A novel support vector machine (SVM) model by combining kernel principal component analysis (KPCA) with improved chaotic particle swarm optimization (ICPSO) is proposed to deal with intrusion detection. The proposed method, in which multi-layer SVM classifier is employed to estimate whether the action is an attack, KPCA is applied as a preprocessor of SVM to reduce the dimension of feature vectors and shorten training time. To shorten the training time and improve the performance of SVM, N-RBF is employed to reduce the noise generated by feature differences, and ICPSO is presented to optimize the punishment factor C, kernel parameters (Formula presented.) and the tube size (Formula presented.) of SVM, which introduces chaos optimization and premature processing mechanism. Experimental results illustrate that the improved SVM model has faster computational time and higher predictive accuracy, and it can also shorten the training time and improve the performance of SVM. © 2014 Springer-Verlag Berlin Heidelberg.
Number of references: 32
Main heading: Support vector machines
Controlled terms: Intrusion detection - Particle swarm optimization (PSO) - Principal component analysis
Uncontrolled terms: Chaos optimization - Chaotic particle swarm optimizations - Computational time - Feature differences - Kernel parameter - Kernel principal component analyses (KPCA) - Multi-layer SVM - Predictive accuracy
Classification code: 723 Computer Software, Data Handling and Applications - 922.2 Mathematical Statistics
DOI: 10.1007/s00500-014-1332-7
Database: Compendex

Data Provider: Engineering Village

Soft Comput
DOI 10.1007/s00500-014-1332-7

METHODOLOGIES AND APPLICATION

A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection

Fangjun Kuang · Siyang Zhang · Zhong Jin · Weihong Xu

Abstract A novel support vector machine (SVM) model by combining kernel principal component analysis (KPCA) with improved chaotic particle swarm optimization (ICPSO) is proposed to deal with intrusion detection. The proposed method, in which multi-layer SVM classifier is employed to estimate whether the action is an attack, KPCA is applied as a preprocessor of SVM to reduce the dimension of feature vectors and shorten training time. To shorten the training time and improve the performance of SVM, N-RBF is employed to reduce the noise generated by feature differences, and ICPSO is presented to optimize the punishment factor C, kernel parameters σ and the tube size ε of SVM, which introduces chaos optimization and premature processing mechanism. Experimental results illustrate that the improved SVM model has faster computational time and higher predictive accuracy, and it can also shorten the training time and improve the performance of SVM.

Communicated by V. Loia.

F. Kuang · Z. Jin · W. Xu
School of Computer Science and Engineering,
Nanjing University of Science and Technology,
Nanjing 210094, China

F. Kuang · S. Zhang
Department of Electrical and Information Engineering,
Hunan Vocational Institute of Safety and Technology,
Changsha 410151, China

W. Xu (✉)
College of Computer and Communications Engineering,
Changsha University of Science and Technology,
Changsha 410077, China
e-mail: xwhxdcs@126.com

Published online: 22 June 2014

Keywords Intrusion detection · Kernel principal component analysis · Support vector machine · Chaotic particle swarm optimization

1 Introduction

As the Internet becomes a part of people's work and daily life, intrusion detection (ID) is an essential requirement to protect the sensitive information stored in the networks. Intrusion detection, as proactive security protection technique in network security, is widely used in detecting, identifying and tracking the intruders (Lee et al. 2008). Researchers always want to find an intrusion detection technology with better detection rate and very low false alarm rate.

Intrusion detection can be seen as a classification problem in essence, to distinguish between the normal activities and the malicious activities. Therefore, some data mining and machine learning techniques are proposed for intrusion detection to automatically learn attack behaviors from historic audit data, such as decision tree (DT) (Lee et al. 2008), genetic algorithm (GA) (Shafi and Abbass 2009), neural network (Wang et al. 2010), principal component analysis (PCA) (Wang and Battiti 2006), fuzzy logic (Chimphlee et al. 2006), K-nearest neighbor (Tsai and Lin 2010), rough set theory (Yang and Zhu 2011) and support vector machine (SVM) (Khan et al. 2007).

Among the methods mentioned above, SVM is an effective one, which is a well-known classifier tool based on small sample learning, it realizes the theory of VC dimension and principle of structural risk minimum, thus it does not have the over fitting problem that artificial neural network cannot overcome. Tsai et al. (2009) thought SVM had manifested its robustness and efficiency in the network action classification, and it was widely used in intrusion detection system (IDS)

as a popular method. Eskin (2000) presented unsupervised anomaly detection model, in which applied three unsupervised learning algorithms, including *K*-neighbor, clustering method and SVM. Hu et al. (2003) proposed an anomaly detection algorithm based on Robust SVM, which can effectively detect intrusions even if noise existed. To improve efficiency of the training, Zhang and Shen (2005) expanded traditional SVM, Robust SVM and One-class SVM to be of online behaviors. Shon et al. (2005) employed SVM for intrusion detection, and used genetic algorithm (GA) for feature selection. Srinoy (2007) proposed an intrusion detection model using SVM and particle swarm optimization (PSO), which used PSO to extract intrusion features and SVM to classify. Peddabachigari et al. (2007) proposed a hierarchical hybrid intelligent system based on decision trees and SVM. Fei et al. (2008) proposed a new anomaly detection algorithm that can update normal profile of system usage pattern dynamically. Horng et al. (2011) used the hierarchical clustering algorithm to provide the SVM with fewer, abstracted, and higher qualified training instances. Wu and Banzhaf (2010) referred to the review of computational intelligence in intrusion detection. Kolias et al. (2011) gave the survey of swarm intelligence in intrusion detection. To overcome the uncertainty problem of an innate feature due to the limited views provided by system monitoring tools, IDS and various types of logs, Kavitha et al. (2012) adopted a new technique known as neutrosophic logic (NL). Kuang et al. (2012) proposed KPCA SVM with GA, which used KPCA to extract intrusion features, and GA to optimize the parameter of SVM.

In addition, when the differences between the sample attributes are very big, using RBF will produce a larger number of support vectors and longer training time. Kuang et al. (2014) presented an improved RBF kernel function (N-RBF) to shorten the training time and improve the performance of SVM.

To solve the above-mentioned problems and get better performance, we propose a new approach for network intrusion detection. In the proposed method, use the KPCA to extract the principal features of the normalized data, and employ multi-layer SVM classifier to estimate whether the action is an attack. To shorten the training time and improve the performance of SVM, use N-RBF to reduce the noise generated by feature differences. A novel improved chaotic particle swarm optimization algorithm (ICPSO) is proposed to optimize the parameters of SVM, which introduces chaos optimization and premature judgment and processing mechanism.

The remainder of this paper is organized as follows. In Sect. 2, the proposed SVM classification model is described, and ICPSO is presented in the Section. Section 3 illustrates how to construct intrusion detection based on the proposed SVM model. The experimental results and discussions are presented in Sect. 4. The conclusions and potential future work are listed in Sect. 5.

2 Related work and contributions

2.1 Kernel principal component analysis

Principal component analysis (PCA) is a common method applied to dimensionality reduction and feature extraction (Jolliffe 1986). PCA method only can extract the linear structure information in the data set but cannot extract this nonlinear structure information. Kernel principal component analysis (KPCA) is an improved PCA, which extracts the principal components by adopting a nonlinear kernel method (Chen et al. 2008; Ding and Tian 2009). A key insight behind KPCA is to transform the input data into a high-dimensional feature space F in which PCA is carried out, and in implementation, the implicit feature vector in F does not need to be computed explicitly, while it is just done by computing the inner product of two vectors in F with a kernel function. Let $x_1, x_2, \ldots, x_n \in R^d$ be the n training samples for KPCA Kuang et al. (2012). The ith KPCA-transformed feature t_i can be obtained by

$$t_i = \frac{1}{\sqrt{\lambda_i}} \gamma_i^{\mathrm{T}} [k(x_1, x_{\mathrm{new}}), k(x_2, x_{\mathrm{new}}), \ldots, k(x_n, x_{\mathrm{new}})]^{\mathrm{T}}, \quad i = 1, 2, \ldots, p \tag{1}$$

Here, Column vector $\gamma_i (i = 1, 2, \ldots, p; 0 < p \leq n)$ is the orthonormal eigenvectors to the p largest positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$, x_{new} is a new column vector sample, $k(x_i, x_j)$ is the calculation of the inner product of two vectors in the hyper-dimensional feature space F with a kernel function.

Using Eq. (1), the KPCA-transformed feature vector of a new sample vector can be obtained.

2.2 SVM classification model

After feature extraction using KPCA, the training data points can be expressed as $(t_1, y_1), (t_2, y_2), \ldots, (t_p, y_p)$, $t_i \in R^k (k < d)$ is the transformed input vector, $y_i \in \{-1, +1\}$ is the target value. In the ε-SVM classification (Srivastava and Bhambhu 2010), the goal is to find a function $f(t)$ that has at most ε deviation from the actually obtained targets y_i for all the training data, and at the same time, is as flat as possible (Kuang et al. 2014). The ε-insensitive loss function denotes as follows

$$e(f(t) - y) = \begin{cases} 0, |f(t) - y| \leq \varepsilon \\ |f(t) - y| - \varepsilon, \text{otherwise} \end{cases} \tag{2}$$

Formally, the optimization problem requires:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{p}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - (w'\Phi(t_i) + b) \le \varepsilon - \xi_i \\ & (w'\Phi(t_i) + b) - y_i \le \varepsilon - \xi_i^* \\ & \xi_i, \xi_i^* \ge 0, i = 1, 2, \ldots, p; C > 0 \end{aligned} \tag{3}$$

where ξ_i and ξ_i^* are slack variables, the punishment factor C is regularization constant, ε denotes the tube size of SVM. C and ε are both determined by users empirically, the constant C determines the trade-off between the flatness of $f(t)$ and the amount up to which deviations large than ε are tolerated.

At the optimal solution, the decision function takes the following form:

$$f(t) = sgn\left(\sum_{i=1}^{p}(\alpha_i - \alpha_i^*)K(t_i, t_j) + b\right) \tag{4}$$

where α_i and α_i^* are the Lagrange multiplier coefficients for the ith training sample, and obtained by solving the dual optimization problem in support vector learning (Srivastava and Bhambhu 2010). The training sample for which $\alpha_i \ne \alpha_i^*$ is corresponded to the support vectors, $K(t_i, t_j)$ is a kernel function, b is found by the Karush–Kuhn–Tucker conditions at optimality.

2.3 N-RBF kernel function for SVM model

In the SVM, there are some common kernels, and any of those can be chosen to achieve the boundary function. Their detailed usages and descriptions, including parameters definitions, can be found in Hsu et al. (2010). In addition, SVM constructed by radial basis kernel function has excellent nonlinear classification ability. In this paper, radial basis kernel function (RBF) used in the SVM classification method is as follows:

$$K(t_i, t_j) = \exp\left(\frac{-\|t_i - t_j\|^2}{\sigma^2}\right), \quad \sigma \in R \tag{5}$$

To shorten the training time and improve its performance, a new kernel function N-RBF is developed to SVM by embedding the mean value and the mean square deviation of attributes, to normalize the attributes' values. The N-RBF is then defined as follows:

$$K(t_i, t_j) = \exp\left(\frac{-\|(t_i - m)/s - (t_j - m)/s\|^2}{\sigma^2}\right) \tag{6}$$

where $m = (m_1, m_2, \ldots, m_j, \ldots, m_k)$ and $s = (s_1, s_2, \ldots, s_j, \ldots, s_k)$ are the mean value and the mean square deviation of attributes, respectively, k is the dimension of sample vectors, m_j and s_j is denoted as $m_j = \frac{1}{n}\sum_{i=1}^{n} L_{ij}$, $s_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(L_{ij} - m_j)}$, respectively. Where $j = 1, 2, \ldots, k$, L_{ij} is the jth attribute of the ith sample, n is the number of training samples.

According to the functional theory, as long as the function K satisfies Mercer's condition, it can be denoted as an inner product of the interchange space, and it should be a positive definite kernel. We know that if K_1 and K_2 are kernel functions in $R^n \times R^n$, and constant $\lambda \ge 0$, then all the functions below are kernel functions:

$$\text{(i) } K(t_i, t_j) = K_1(t_i, t_j) + K_2(t_i, t_j) \tag{7}$$

$$\text{(ii) } K(t_i, t_j) = \lambda K_1(t_i, t_j) \tag{8}$$

$$\text{(iii) } K(t_i, t_j) = K_1(t_i, t_j)K_2(t_i, t_j) \tag{9}$$

$$\text{(iv) } K(t_i, t_j) = \exp(K_1(t_i, t_j)) \tag{10}$$

Theorem 1 *Function N-RBF is a kernel function.*

Proof

$$\begin{aligned} K(t_i, t_j) &= \exp\left(\frac{-\|(t_i - m)/s - (t_j - m)/s\|^2}{\sigma^2}\right) \\ &= \exp\left(\frac{-\|(t_i - m)/s\|^2}{\sigma^2}\right) \\ &\quad \times \exp\left(\frac{-\|(t_j - m)/s\|^2}{\sigma^2}\right) \\ &\quad \times \exp\left(\frac{2((t_i - m)/s)((t_j - m)/s)}{\sigma^2}\right) \end{aligned} \tag{11}$$

Let $f(t_i) = \exp\left(\frac{-\|(t_i - m)/s\|^2}{\sigma^2}\right)$ and $h(t_i, t_j) = \exp\left(\frac{2((t_i - m)/s)((t_j - m)/s)}{\sigma^2}\right)$ are real functions. We have $K(t_i, t_j) = f(t_i)f(t_j)h(t_i, t_j)$. □

We can conclude that $f(t_i)f(t_j)$ is a positive definite kernel, according to the theorem that if ϕ is a real function defined in the space R^n, and then $\phi(t_i)\phi(t_j)$ is a positive definite kernel.

Then, we define $g(t_i) = (t_i - m)/s$, where $g(t_i)$ is a real function.

Similarly, we conclude that $g(t_i)g(t_j) = ((t_i - m)/s)((t_j - m)/s)$ is a positive definite kernel.

Additionally, since $\sigma^2 > 0$, function $\beta(t_i, t_j) = \frac{2g(t_i)g(t_j)}{\sigma^2}$ is a positive definite kernel. Therefore, according to formula (9), the function $h(t_i, t_j)$ is a positive definite kernel.

Since both of $f(t_i)f(t_j)$ and $h(t_i, t_j)$ are positive definite kernels, according to formula (8), function $K(t_i, t_j)$ is also a positive definite kernel.

Therefore, the kernel N-RBF is a positive definite kernel.

Consequently, C, ε and σ are user-determined parameters, the selection of the parameters plays an important role in the performance of SVM model. Several disciplined approaches can be used to obtain the optimal parameters for SVM model, out of which, evolutionary method such as

genetic algorithm, simulated annealing algorithm and PSO algorithm, is one of the most widely used approaches. In this paper, the ICPSO algorithm is proposed to optimize the parameters of SVM.

2.4 ICPSO algorithm

2.4.1 Particle swarm optimization algorithm

Particle swarm optimization (PSO) is a computation intelligence technique, which was motivated by the organisms' behavior such as schooling of fish and flocking of birds (Wang et al. 2009). PSO can solve a variety of difficult optimization problems. The major advantage is that PSO uses the physical movements of the individuals in the swarm and has a flexible and well-balanced mechanism to enhance and adapt to the global and local exploration abilities. Another advantage of PSO is its simplicity in coding and consistency in performance. In the D-dimensional search space, the current position of the ith particle is represented by a vector: $P_i^d(t)(i = 1, 2, \ldots, S; d = 1, 2, \ldots, D)$, S is the number of particles. The best previous position of the ith particle is recorded and represented as $P_{\text{best}}^{d}(t)$. The global best particle in the swarms is represented by $G_{\text{best}}^{d(t)}(t)$. The velocity of the ith particle is represented as $V_i^d(t)$. Let t denote the current generation, the velocity and position of the dth element of the ith particle at $(t + 1)$th search from the knowledge of previous search are updated according to the following equations.

$$\begin{cases} V_i^d(t+1) = W \cdot V_i^d(t) + C_1 \cdot R_1 \cdot (P_{\text{best}}^d(t) - P_i^d(t)) \\ \qquad\qquad +C_2 \cdot R_2 \cdot (G_{\text{best}}^d(t) - P_i^d(t)) \\ P_i^d(t+1) = P_i^d(t) + V_i^d(t+1) \end{cases} \tag{12}$$

where C_1 and C_2 are the positive constant parameters, R_1 and R_2 are the random functions in the range [0, 1], the inertial weight W is used to balance the capabilities of global exploration and local exploration.

2.4.2 ICPSO algorithm

The performance of PSO often suffers the problems of slow convergence speed during the later period and trapped in local optima. A novel ICPSO is proposed to optimize the parameters of SVM, which introducing chaos optimization algorithm and the premature judgment and processing mechanism.

2.4.2.1 Chaos optimization Chaos is characterized as ergodicity, randomicity and regularity. Because chaos queues can experience all the states in a specific area without repeat, chaotic search becomes a novel tool used as an optimizer (Li and Jiang 1997). In general, the parameters C_1, C_2, R_1, R_2 and W are the important factors which influence the convergence of the PSO. However, parameters R_1 and R_2 cannot guarantee the optimization's ergodicity entirely in phase space because they are absolutely random in the traditional PSO. Therefore, chaotic mapping with certainty, ergodicity and the stochastic property is introduced into PSO to improve the global convergence. C_1, C_2, R_1, R_2 and W are chosen as follows:

$$C_i(t) = 4.0C_i(t-1)(1 - C_i(t-1)) \tag{13}$$
$$C_i(t) = C_{\min} + (C_{\max} - C_{\min})C_i(t) \tag{14}$$
$$R_i(t) = 4.0R_i(t-1)(1 - R_i(t-1)) \tag{15}$$
$$W(t) = 4.0W(t-1)(1 - W(t-1)) \tag{16}$$
$$W(t) = W_{\min} + (W_{\max} - W_{\min})W(t) \tag{17}$$

where $C_{\max}$ and $C_{\min}$ denote the max and the min of acceleration constants which are taken as 2.0 and 1.4, respectively; $R_i(t) \in (0, 1), i = 1, 2$; $W_{\max}$ and $W_{\min}$ denote the max and the min weights which are taken as 0.9 and 0.4, respectively; t is the current generation.

2.4.2.2 Premature judgment and processing mechanism The position of particles determines the fitness of the particle, so we can track the status of particle swarms by the overall changing of all particles fitness. Group fitness variance δ^2 is defined as follows:

$$\delta^2 = \sum_{i=1}^{N} \frac{F_i - F_{\text{avg}}}{F} \tag{18}$$

where N is population size; F_i denotes the fitness of the ith particle; F_{avg} is the average fitness of the particle swarms; F denotes normalization factor for limiting the size of δ^2, which is expressed as follows:

$$F = \begin{cases} \max\limits_{1 \le i \le m} \left|F_i - F_{\text{avg}}\right|, & \max\limits_{1 \le i \le m} \left|F_i - F_{\text{avg}}\right| > 1 \\ 1, & \text{else} \end{cases} \tag{19}$$

If $\delta^2 < H$ (H is a given constant), the premature processing is applied. Particles trapped in premature are employed in chaos optimization according to the following equations.

$$V_i^d(t) = 4.0V_i^d(t-1)(1 - V_i^d(t-1)) \tag{20}$$
$$V_i^d(t) = V_{\min} + (V_{\max} - V_{\min})V_i^d(t) \tag{21}$$

where $[V_{\min}, V_{\max}]$ is the velocity range of the particles. The premature particles are updated according to the following

equations.

$$\begin{cases} V_i^d(t+1) = W(t) \cdot V_i^d(t) + C_1(t) \cdot R_1(t) \cdot (P_{\text{best}}^d(t) \\ \quad -P_i^d(t)) + C_2(t) \cdot R_2(t) \cdot (G_{\text{best}}^d(t) - P_i^d(t)) \\ P_i^d(t+1) = P_i^d(t) + V_i^d(t+1) \end{cases} \tag{22}$$

where t is the current generation, T denotes the maximum number of generations.

2.5 Optimizing the parameters of SVM model with ICPSO

By means of the ICPSO algorithm, the three major parameters C, σ and ε of SVM model can be optimized. In solving the parameter selection, each particle represents a potential solution, comprised of a vector (C, σ, ε) $D = 3$. The parameter optimality is measured by means of fitness functions that are defined in relation to the considered optimization problem. In the training and testing process of SVM, the objective is to improve the generalization performance of the regression model, namely, minimize the errors between the true values and forecasting values of the testing samples. Therefore, the fitness function can be defined as follows.

$$\text{Fitness} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{m}\sum_{j=1}^{m}(f(x_{ij}) - y_{ij})^2} \tag{23}$$

where n is the number of folds for cross-validation, m is the number of each subset as validation, y_{ij} and $f(x_{ij})$ represent the actual value and the forecast value of validation samples, respectively.

The objective is to minimize the fitness, so the particle with the minimal fitness value will outperform others and should be reserved during the optimization process. Accordingly, the optimal parameters can be selected. The process of optimizing the SVM parameters with ICPSO is shown in Fig. 1, which is described as follows.

Step 1: Initialize the swarm size S, maximum of generations T, setting $t = 1$, $[\text{W}_{\min}, \text{W}_{\max}]$, $[\text{C}_{\min}, \text{C}_{\max}]$, $[V_{\min}, V_{\max}]$, $D = 3$, $[P_{\min}^d, P_{\max}^d]$ is the value range of SVM parameters, where $d = 1, 2, \ldots, D$. C_1, C_2, R_1, R_2 and W are generated by chaos optimization using Eqs. (13)–(17).

Step 2: Produce the positions and velocity of particles by chaos initialization.

Step 2.1: Initialize a vector $Z_i^d(0)(d = 1, 2, \ldots, D)$, which each component is set the range (0, 1). Generate chaos queues $Z_i^d(t)$ $(i = 1, 2, \ldots, N, N > M)$ by iteration of Logistic equation, which is represented as $Z_i^d(t) = 4.0 \times Z_i^d(t-1) \times (1 - Z_i^d(t-1))$, $i = 0, 1, \ldots, S$.

Step 2.2: Transfer the chaos queues into the range of the parameters of SVM according to $P_i^d(t) = P_{\min}^d + (P_{\max}^d - P_{\min}^d)Z_i^d(t)$.

Step 2.3: Calculate the fitness values of the particles according to Eq. (23), and choose the best M solutions with the minimal fitness in the swarm as the initial solutions of M particles, and randomly initialize the velocity of M particles.

Step 2.4: Obtained the individual best P_{best}^d and the global best G_{best}^d.

Step 3: If the convergence criteria or one of the stopping criteria (Generally, a sufficiently good fitness or maximum iteration is met) is satisfied, go to step 10.

Step 4: Update the velocity V_i and position P_i of each particle according to Eq. (22), respectively. And C_1, C_2, R_1, R_2 and W are Obtained by Step 1.

Step 5: Compare the fitness value of each particle to its individual best P_{best}^d, if current value is better than P_{best}^d, then update P_{best}^d as current position.

Step 6: Compare the fitness value of each particle to the global best G_{best}^d. If current value is better than G_{best}^d, then update G_{best}^d as current position.

Step 7: If the convergence criteria or one of the stopping criteria (Generally, a sufficiently good fitness or maximum iteration is met) is satisfied, go to step 10.

Step 8: Calculate the group fitness variance δ^2 by Eqs. 18, 19. If $\delta^2 < H$ is not satisfied, let $t = t + 1$ and go back to Step 4.

Step 9: Update the velocity and position of the premature particles according to Eqs. (20)–(22), let $t = t + 1$, and go back to Step 3.

Step 10: Obtain the optimal parameters C, σ and ε of SVM model.

3 Proposed SVM model for intrusion detection

3.1 Intrusion detection types and normalized

This paper takes the KDD CUP99 as the datasets of the experiments (Stolfo et al. 2011). The original datasets consist of the training datasets and testing datasets. The datasets can be divided into five categories which are normal, denial of service (DoS), unauthorized access from a remote machine (Remote to Local, R2L), unauthorized access to local supervisor privileges (User to Root, U2R) and probing. Each network record contains 41 attributes, of which 34 are continuous attributes and 7 are discrete ones.

Before the experiments, we need to deal with the discrete attributes by counting the frequency of their values and converting them to numerical attributes, and transformed all attributes into the normalized format. That is, treat the discrete property whose value is 0 or 1 as a continuous property. If the value of one discrete property is noun, we then divide it into several sub-properties according to all possible values of it. Take protocol_type as an example. We divide it into three sub-properties that is protocol_type1 (tcp), protocol_type2 (udp) and protocol_type3 (icmp). If its value is tcp in the records, the sub-property protocol_type1 will be set to

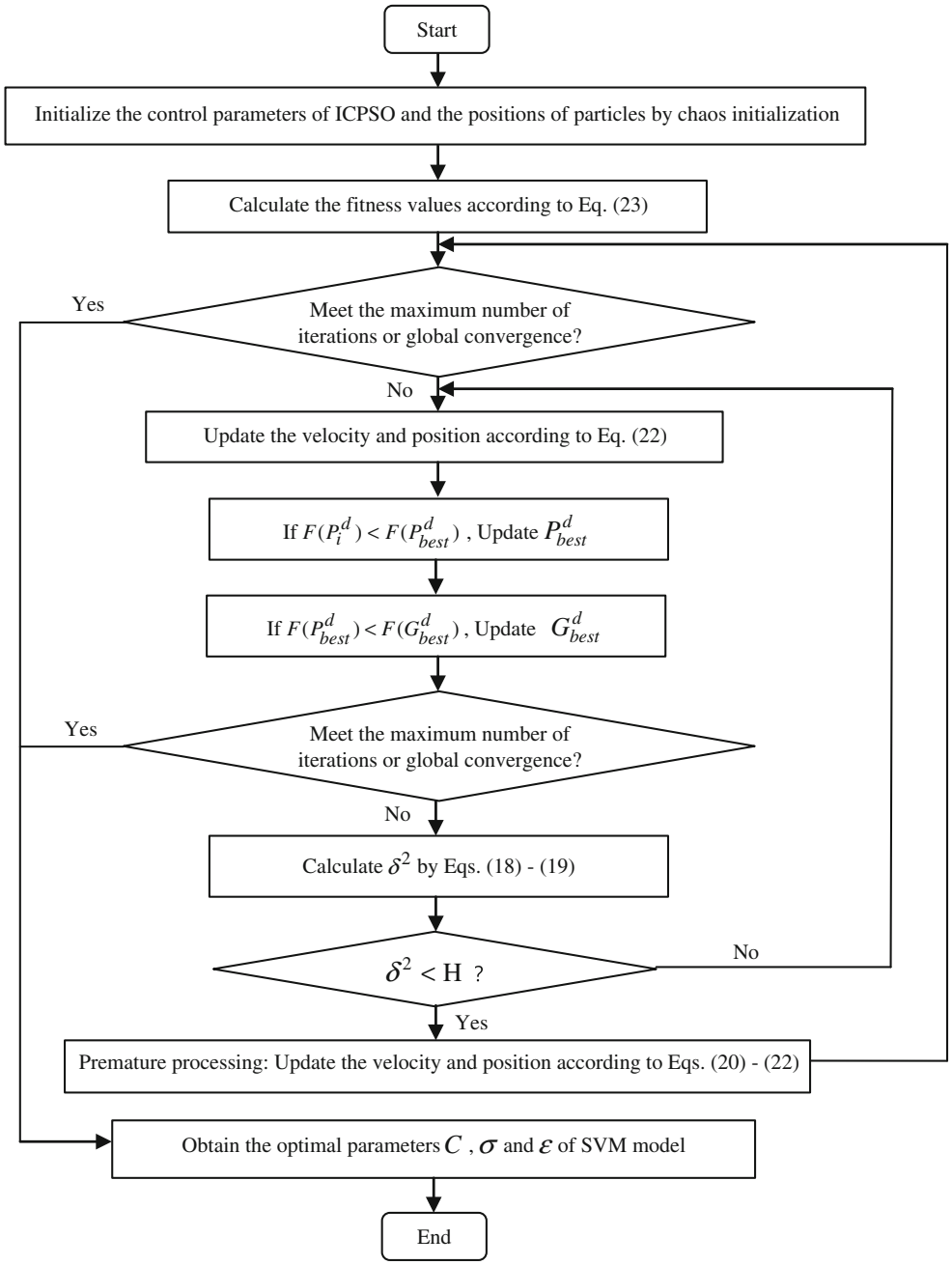

Fig. 1 Optimizing the parameters of improved SVM model with ICPSO

1 while the other two sub-properties set to 0. This is beneficial because it can ensure the differences between identical discrete properties in different records are equal to each other and deviation in calculation will be avoided.

3.2 Intrusion detection based on proposed SVM model

Multi-SVM classifiers are applied to intrusion detection because of multi-types existing in network. 'One-against-one', 'One-against-all' and 'Binary tree' are the popular methods in SVM multi-class classification (Srivastava and Bhambhu 2010). As shown in Fig. 2, 'Binary tree' SVM classification algorithm needs only $k-1$ two-class SVM classifiers for a case of k classes, while 'One-against-all' SVM classification algorithm needs k two-class SVM classifiers where each one is trained with all the samples and 'One-against-one' SVM classification algorithm needs $k(k-1)/2$ two-class SVM classifiers where each one is trained on data from two classes (Srivastava and Bhambhu 2010; Hsu et al. 2010). Obviously less two-class classifiers help expedite the rate of

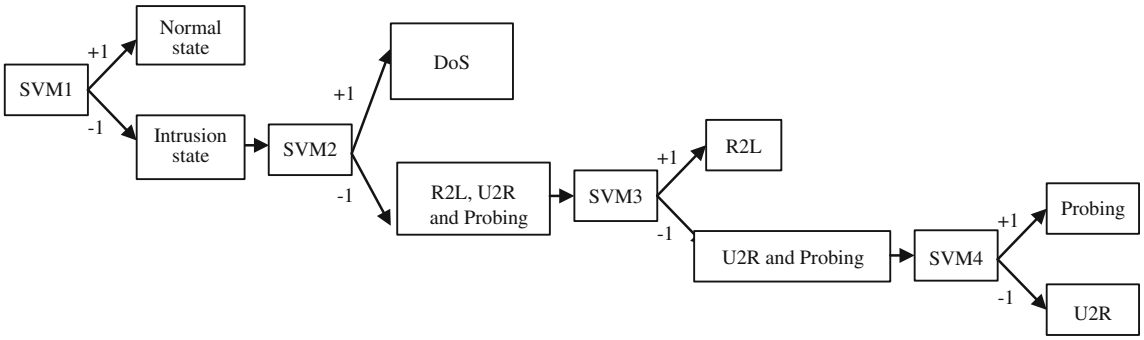

Fig. 2 The scheme of intrusion detection based on improved SVM model

training and recognition. Thus, 'Binary tree' SVM classification algorithm is adapted to construct detection model in this paper.

Based on the characteristics of different intrusion detection types, four SVM classifiers are developed to identify the five states: normal state (Nc) and the four intrusion state (DoS, R2L, U2R, and Probing) (Kuang et al. 2014). The scheme of intrusion detection model based on improved SVM classifiers by combining KPCA with ICPSO is shown in Fig. 2.

All the four SVMs adopt the N-RBF function as their kernel function, the parameters C, σ and ε are optimized with ICPSO. The adjusted parameters with maximal classification accuracy are selected as the most appropriate parameters. Then, the optimal parameters are utilized to train the SVM classifiers.

3.3 Proposed intrusion detection model implementation

Intrusion detection belongs to classification problems in essence, it discriminates abnormal data from anomaly data, and intrusion data is of a high dimension and contains many noise attributes. Therefore, KPCA is used to extract the principal components, SVM classifiers are applied to intrusion detection. The proposed hybrid approach is composed of three stages. In the first stage, the principal components are achieved based on KPCA theory, which find an optimal subset of all attributes and delete irrelevant and redundant attributes that have no any classification ability. In this paper, we chose p eigenvectors by trial and error, which corresponded to the first p biggest eigenvalues, to form the sub-eigenspace, satisfying $\sum_{i=1}^{p} \lambda_i / \sum_{i=1}^{n} \lambda_i \geq 90\,\%$.

The second stage is to use this attribute subset as the training dataset and testing dataset of SVM to perform the classification, and N-RBF kernels are adopted for SVM, ICPSO is used to select the optimal parameter of SVM. The third stage is to use negative mean absolute percentage error (MAPE) as criteria evaluation. MAPE $= \frac{1}{N} \sum_{i=1}^{N} \left| \frac{a_i - d_i}{a_i} \right| \times 100\,\%$, where a_i and d_i represent the actual and forecast values, respectively; N is the number of classification. Figure 3 shows the flowchart of KPCA-ICPSO-SVM classification model for intrusion detection.

4 Experimental results and discussions

4.1 Experimental description

There are some performance indicators for the intrusion detection system as follows: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). TP represents the abnormal behavior is correctly detected, FP represents the normal behavior is judged as abnormal, TN represents the normal behavior is correctly forecasted, and FN represents the abnormal behavior is wrongly thought as normal (Tsai and Lin 2010).

(1) Detection rate: $\mathrm{DR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ (24)

(2) False alarm rate: $\mathrm{FAR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$ (25)

(3) Correlation coefficient:

$$\mathrm{CC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FN})(\mathrm{TP} + \mathrm{FP})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{26}$$

where DR denotes the detection rate and FAR denotes the false alarm rate. They are important to evaluate the performance of the intrusion detection system. In addition, CC denotes the correlation between the forecast result and the actual situation. It ranges from -1 to 1, where 1 represents the forecast result is fully consistent with the actual situation and 0 is on behalf of a random prediction.

In this paper, the detection rate, false alarm rate and correlation coefficient are used as the evaluation indicators for KPCA-ICPSO-SVM. The purpose of KPCA-ICPSO-SVM is not only to enhance the intrusion detection rate and reduce false alarm rate, but also to reduce the training and testing time as much as possible. So the training and testing time are adopted as well. The experiments are processed within a MATLAB R2013b environment, which is running

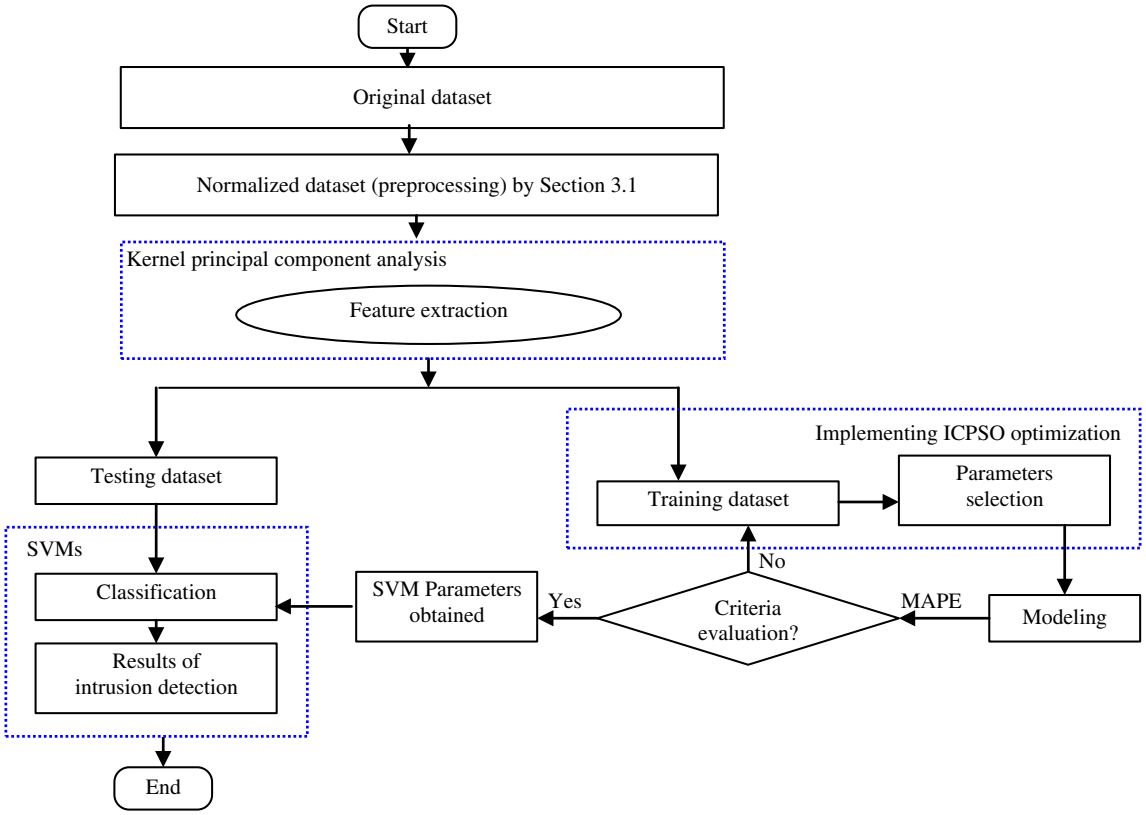

Fig. 3 The flowchart of the proposed KPCA-ICPSO-SVM model for intrusion detection

on a PC powered by Pentium IV 3.0 GHz CPU and 3.0 GB RAM.

4.2 Experiments of KPCA-ICPSO-SVM

In this section, we selected samples from the subset of KDD to form the training and testing set. There are five data sets in Table 1.

To verify the performance and effectiveness of KPCA-ICPSO-SVM, the subset we obtained in Table 1 was randomly divided into two subsets, each subset contains both the data of normal and abnormal class, one was as the training set, and the other was as the testing set. Second, randomly select 10 datasets from the training subset, named from F1 to F10,

Table 1 Five training and testing sets

No.	Training set			Test set		
	Normal (%)	Abnormal (%)	Total	Normal (%)	Abnormal (%)	Total
D1	83.5	16.5	12,560	72.5	17.5	11,040
D2	90.5	9.5	11,050	35.0	65.0	11,428
D3	55.3	44.7	9,040	57.9	42.1	13,818
D4	93.9	6.1	10,640	85.8	14.2	11,650
D5	76.5	23.5	6,540	64.9	35.1	12,318

Table 2 Different optimization algorithm for the SVM parameters combination

Models	Parameters		
	C	σ	ε
KPCA-ICPSO-SVM	127.784	12.481	0.00017
KPCA-CPSO-SVM	86.784	2.496	0.00032
N-KPCA-GA-SVM	83.5191	0.0907	0.0008
KPCA-GA-SVM	218.835	0.7319	0.00739
PCA-GA-SVM	79.437	9.8423	0.2856
PCA-PSO-SVM	198.839	21.8332	0.4872
CPSO-SVM	97.9347	15.0348	0.00347

as the training set, and any two training sample sets did not intersect. Third, from the testing subset, select the normal and attack records with the same number to form the testing set.

Now, we evaluated KPCA-ICPSO-SVM by comparing it with KPCA-CPSO-SVM, N-KPCA-GA-SVM (Kuang et al. 2014), KPCA-GA-SVM (Kuang et al. 2012), PCA-GA-SVM, PCA-PSO-SVM, CPSO-SVM (Zhang and Li 2012), Single-SVM and radical basis function neural networks (RBFNN) on the detection rate (DR), false alarm rate (FAR), correlation coefficient (CC), and training time (TrD) and testing time (TeD). We employed four SVMs for the five-class classification problem including Sect. 3.2, and partitioned

Table 3 Experiment results among different algorithms

Models	Datasets									
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
KPCA-ICPSO-SVM										
DR (%)	95.036	95.987	95.732	95.021	96.698	96.523	96.357	96.174	95.368	96.372
FAR (%)	1.012	1.006	0.925	1.015	1.001	0.948	0.992	1.005	0.964	0.978
CC	0.962	0.969	0.964	0.952	0.963	0.978	0.958	0.967	0.958	0.976
TrD (s)	0.738	1.629	1.292	1.106	0.454	0.563	0.993	0.623	0.682	0.697
TeD (s)	1.961	5.329	3.008	2.163	1.012	1.139	1.078	1.034	1.236	1.224
KPCA-CPSO-SVM										
DR (%)	94.537	95.478	95.226	94.548	96.379	96.289	95.144	95.815	194.846	95.992
FAR (%)	1.108	1.017	1.056	1.337	1.204	1.028	1.004	1.032	0.975	1.134
CC	0.958	0.961	0.952	0.948	0.951	0.967	0.949	0.956	0.952	0.958
TrD (s)	0.872	1.823	1.471	1.213	0.468	0.628	1.099	0.753	0.642	0.752
TeD (s)	2.065	5.986	3.261	2.647	1.235	1.241	1.198	1.421	1.536	1.321
N-KPCA-GA-SVM										
DR (%)	94.226	95.302	95.188	94.264	96.302	96.377	95.302	95.302	94.280	96.032
FAR (%)	1.025	1.025	1.0	1.35	1.0	0.956	1.025	1.0	0.975	0.984
CC	0.955	0.966	0.935	0.941	0.946	0.968	0.956	0.956	0.949	0.963
TrD (s)	0.718	1.719	1.328	1.015	0.438	0.553	0.984	0.453	0.438	0.672
TeD (s)	1.985	5.546	3.391	2.719	1.105	1.11	1	1.11	1.469	1.286
KPCA-GA-SVM										
DR (%)	92.065	93.033	92.617	93.936	94.017	95.175	93.828	92.093	90.615	93.092
FAR (%)	4.25	4.2	4.3	4.475	4.2	4.9	4.15	4.15	4.425	4.452
CC	0.814	0.831	0.826	0.818	0.839	0.848	0.838	0.84	0.767	0.869
TrD (s)	2.078	6.781	5.797	3.156	8.609	13.812	8.156	10.485	1.094	6.678
TeD (s)	6.218	13.641	11.719	9.266	16.938	21.328	15.532	18.969	4.656	18.254
PCA-GA-SVM										
DR (%)	87.403	86.567	87.529	82.475	85.995	86.45	88.166	83.346	88.615	86.658
FAR (%)	3.675	4.4	5.3	5.125	4.075	4.175	4.375	4.05	4.425	4.478
CC	0.867	0.891	0.879	0.789	0.832	0.835	0.880	0.810	0.867	0.852
TrD (s)	7.547	13.3	14.44	6.85	9.164	15.27	18.69	23.2	1.105	14.264
TeD (s)	16.203	14.297	19.656	13.984	14.563	36.047	30.547	30.016	5.688	24.689
PCA-PSO-SVM										
DR (%)	88.826	87.353	89.287	83.769	86.422	87.559	90.642	85.042	89.907	88.356
FAR (%)	3.398	4.226	4.642	4.917	3.879	4.006	4.101	3.983	4.285	4.129
CC	0.878	0.897	0.885	0.842	0.859	0.864	0.892	0.835	0.872	0.868
TrD (s)	7.225	11.984	14.902	6.732	9.028	14.252	15.671	26.012	1.219	13.893
TeD (s)	14.865	13.381	15.334	14.082	13.872	32.372	29.637	29.034	6.336	22.345
CPSO-SVM										
DR (%)	87.059	85.269	85.987	81.093	83.458	85.075	87.231	82.181	86.016	84.935
FAR (%)	3.734	4.454	5.423	5.634	4.448	4.365	4.627	5.602	4.971	4.701
CC	0.862	0.868	0.876	0.791	0.829	0.842	0.875	0.811	0.863	0.849
TrD (s)	8.347	15.201	16.034	8.012	14.342	23.354	19.069	26.243	1.573	16.125
TeD (s)	17.732	17.452	21.345	16.342	15.563	36.047	33.563	32.556	6.763	26.436
Single-SVM										
DR (%)	86.752	77.139	76.571	81.302	75.095	79.637	76.95	75.007	78.615	80.765
FAR (%)	10.95	6.275	5.875	5.8	6.3	6.475	5.625	3.125	4.425	6.8
CC	0.754	0.729	0.73	0.771	0.712	0.748	0.737	0.724	0.767	0.762
TrD (s)	3.844	18.86	17.093	15.625	22.672	28.14	18.047	33.094	1.016	16.251
TeD (s)	14.813	26.656	23.922	20.562	42.094	43.813	35.047	47.969	5.64	32.682
RBFNN										
DR (%)	87.063	79.236	77.139	82.265	73.265	80.983	77.654	78.278	80.142	82.247
FAR (%)	8.68	5.62	5.854	6.26	6.85	9.475	6.487	6.128	5.825	5.41
CC	0.812	0.789	0.768	0.798	0.708	0.804	0.826	0.804	0.828	0.8141
TrD (s)	18.345	20.662	15.216	13.245	24.132	26.254	19.452	31.421	2.345	15.564
TeD (s)	16.952	28.346	30.983	24.652	45.584	44.987	26.253	46.874	8.986	30.248

the data into the two classes of "Normal" and "Rest" (DoS, R2L, U2R, and Probing) patterns, where the rest was the collection of four classes of attack instances in the dataset. The objective was to separate normal and attack patterns. Repeat this process for all classes.

In KPCA-ICPSO-SVM, KPCA-CPSO-SVM and N-KPCA-GA-SVM model, KPCA was applied to extract feature, which held the principal features and abandoned the subordinate and noise data. N-RBF kernels were adopted for SVM. ICPSO, CPSO and GA were used to optimize the parameters (C, σ, ε) of SVM, respectively. In the other SVM models, RBF kernels were used as the kernel functions of SVM. The parameters of all optimization algorithms were chosen as follows: population size: 50, maximal iteration: 200. In KPCA-ICPSO-SVM model: $C_{\max} = 300$, $W_{\max} = 0.9W_{\min} = 0.4$, $C_{\max} = 2.0$, $C_{\min} = 1.4$, $H = 1$. In KPCA-CPSO-SVM model: $C_{\max} = 300$, $W_{\max} = 0.9W_{\min} = 0.4$, $C_1 = 1.5$, $C_2 = 1.7$. In N-KPCA-GA-SVM, the probabilities of crossover and mutation were set to 0.8 and 0.05, respectively. The parameters σ, ε and C of Single-SVM were randomly selected, while the parameters of the other SVM models were obtained by the corresponding optimization algorithm. Through 30 simulation experiments, the parameters (C, σ, ε) of SVMs are shown in Table 2. In RBFNN model, RBFNN had four-layer ANN, with 5 input neurons, with two hidden layers with 20 and 30 neurons each, and 5 output neurons. The experiment results among different algorithms are listed in Table 3.

The performance comparisons of all models in detection rate (DR), false alarm rate (FAR), the correlation coefficient (CC), the training time (TrD) and testing time (TeD) are shown in Figs. 4, 5, 6, 7, 8.

As shown in Table 3 and Figs. 4, 5, 6, 7, 8, we can see that the performance of KPCA-ICPSO-SVM, KPCA-CPSO-SVM, N-KPCA-GA-SVM, KPCA-GA-SVM, PCA-GA-SVM, PCA-PSO-SVM and CPSO-SVM was better than Single-SVM. The reason is that the parameters of Single-SVM are randomly selected, while the parameters of the other SVM are obtained by the corresponding optimization

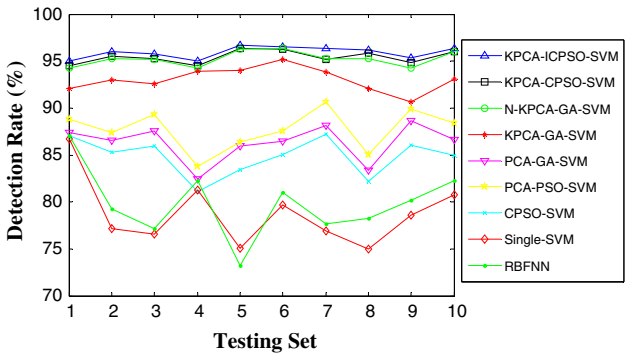

Fig. 4 Comparison of detection rate

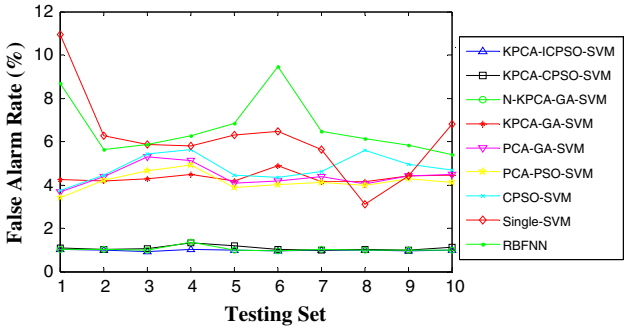

Fig. 5 Comparison of false alarm rate

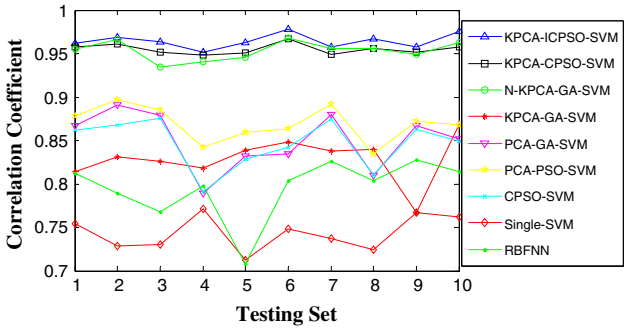

Fig. 6 Comparison of correlation coefficient

algorithm. SVM classifier for intrusion detection using PCA, KPCA to extract feature has a good performance in DR, FAR, CC and runtime than that without feature extraction. Furthermore, results also show that KPCA is better than PCA. The reason lies in the fact that KPCA can provide more additional discriminatory information for improving classification performance than PCA, and dimension reduction can improve the generalization performance and running time of SVM classifier. We can also see that Single-SVM needs longer training time, because it has to do cross-judging and more training. RBFNN also obtains good classification accuracy, but RBFNN requires large amounts of training data, and needs to adjust the parameters of the hidden activation function, the parameters are determined by experience or using the optimum method to tune the network parameters and connecting weights. In addition, Table 3 and Figs. 4, 5, 6, 7, 8 can also see that the overall performance of KPCA-ICPSO-SVM model is better than the other models for intrusion detection. The above results show that ICPSO plays some role in DR, FAR and CC, and N-RBF also plays some role in saving the training and testing time.

The above experiments have not considered the attacks of different kinds. To further analyze the detection performance of KPCA-ICPSO-SVM on unknown attacks, we gave the following experiment. In the experiment, we select samples randomly from the KDD CUP1999 datasets, 6,092 of which were chosen as training datasets and 5,890 of which as the testing datasets. Comparing KPCA-ICPSO-SVM

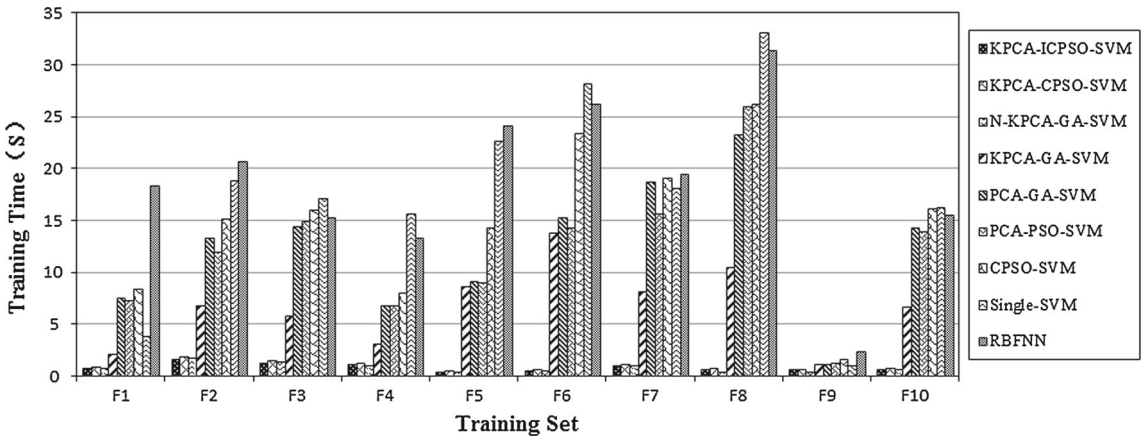

Fig. 7 Comparison of training time

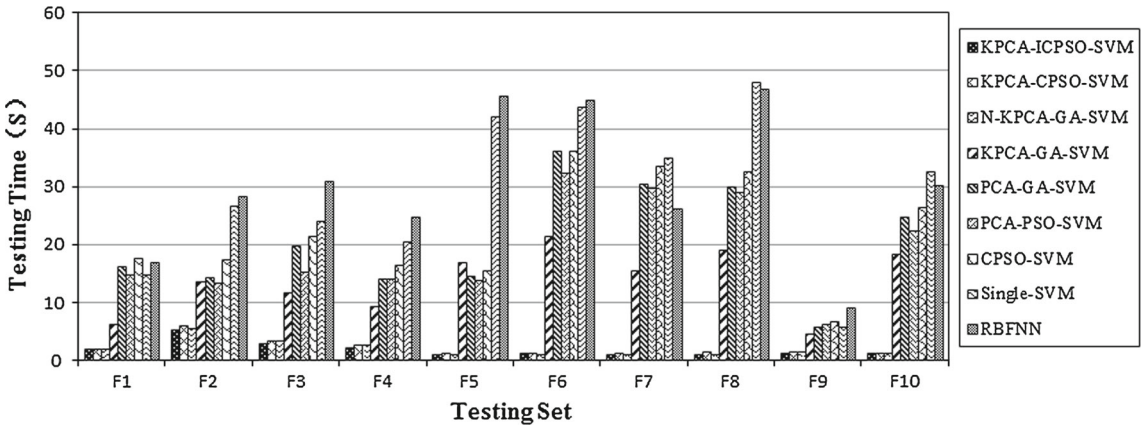

Fig. 8 Comparison of testing time

with KPCA-CPSO-SVM, N-KPCA-GA-SVM, KPCA-GA-SVM, CPSO-SVM and RBFNN in the experiment, and counted the accuracy rates (Acc) and testing time (TeD) of these models on the attacks of all categories, where $\text{Acc} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$. The comparisons of experimental results in 30 simulation experiments are given in Table 4.

As shown in Table 4, we can see that all models show high detection rates on forecasting normal behaviors, and the first three kinds of models have the higher the accuracy rates and the less testing time on predicting the attacks of Probing and DoS. However, the results for detecting attacks of U2R and R2L are all unsatisfactory. In general, the accuracy rates and testing time of KPCA-ICPSO-SVM on attacks of all categories is better than the other five models.

5 Conclusions

In this paper, a Novel hybrid KPCA SVM with ICPSO model is proposed for intrusion detection. In the KPCA-ICPSO-SVM model, KPCA is adopted to extract the principal features of the intrusion detection data, and multi-layer SVM classifier is employed to estimate whether the action is an attack. N-RBF kernel function is employed to shorten the training time and improve the performance of SVM classification model, ICPSO is proposed to select suitable parameters for SVM classifier, which introduces chaos optimization and premature judgment and processing mechanism. The experimental results show that the classification accuracies of the proposed KPCA-ICPSO-SVM model are superior to those of SVM classifiers whose parameters are randomly selected, and SVM classifier by feature extraction using KPCA can achieve better generalization performance than that without feature extraction. The reason lies in the fact that KPCA can explore higher order information of the original inputs. The test results indicate that the proposed method shows more excellent detection performance for intrusion detection, and also saves a lot of training and testing time.

For future work, we will focus on how to improve the detection rate on predicting attacks, especially the attacks of U2R and R2L. And research some other optimization algorithm for SVM parameters optimization.

Table 4 Comparisons of the detection performance of various categories

Models	Categories				
	Normal	Probing	Dos	U2R	R2L
KPCA-ICPSO-SVM					
TeD (s)	2.462	1.634	6.093	1.826	1.385
Acc (%)	98.134	96.652	94.291	74.453	73.944
KPCA-CPSO-SVM					
TeD (s)	4.133	3.046	8.029	4.115	3.728
Acc (%)	96.242	94.794	93.181	72.675	72.553
N-KPCA-GA-SVM					
TeD (s)	5.037	3.621	7.737	5.314	4.115
Acc (%)	96.046	94.492	93.079	72.436	72.257
KPCA-GA-SVM					
TeD (s)	5.485	4.267	8.712	5.679	4.926
Acc (%)	94.863	93.857	91.622	71.849	71.638
CPSO-SVM					
TeD (s)	8.372	6.986	11.213	8.268	7.131
Acc (%)	94.263	93.292	90.324	69.438	70.871
RBFNN					
TeD (s)	13.952	11.326	15.687	12.469	11.913
Acc (%)	93.384	88.497	85.962	67.235	66.687

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grant 61373063 and 61233011, Science and Technology Department of Hunan Province of China under Grant 2012SK4046 and 2013FJ4217, and Research Foundation of Education Bureau of Hunan Province of China under Grant 13C086. And the authors are grateful to the referees for their suggestions and comments.

References

Chen ZG, Ren HD, Du XJ (2008) Minimax probability machine classifier with feature extraction by kernel PCA for intrusion detection. In: Proceedings of WiCOM08, pp 1–4

Chimphlee W, Addullah AH, Sap MNM et al (2006) Anomaly-based intrusion detection using fuzzy rough clustering. In: Proceedings of ICHIT06, pp 329–334

Ding M, Tian Z, Xu H (2009) Adaptive kernel principal analysis for online feature extraction. Proc World Acad Sci Eng Technol 59:288–293

Eskin E (2000) Anomaly detection over noisy data using learned probability distributions. In: Proceedings of ICML2000, pp 255–262

Fei R, Hu L, Liang H (2008) Using density-based incremental clustering for anomaly detection. In: Proceedings of CSSE08, pp 986–989

Horng SJ, Su MY, Chen YH et al (2011) A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Syst Appl 38:306–313

Hsu CW, Chang CC, Lin C J (2010) A practical guide to support vector classification. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf. Accessed 2 December 2011

Hu W, Liao Y, Vemuri V (2003) Robust support vector machines for anomaly detection in computer security. In: Proceedings of ICMLA03, pp 23–24

Jolliffe IT (1986) Principle component Analysis. Springer, New York

Kavitha B, Karthikeyan S, Maybell PS (2012) An ensemble design of intrusion detection system for handling uncertainty using neutrosophic logic classifier. Knowl Based Syst 28:88–96

Khan L, Awad M, Thuraisingham B (2007) A new intrusion detection system using support vector machines and hierarchical clustering. Int J Very Large Data Bases 16:507–521

Kolias C, Kambourakis G, Maragoudakis M (2011) Swarm intelligence in intrusion detection: a survey. Comput Secur 30:625–642

Kuang FJ, Xu WH, Zhang SY et al (2012) A novel approach of KPCA and SVM for intrusion detection. J Comput Inform Syst 8(8):3237–3244

Kuang FJ, Xu WH, Zhang SY (2014) A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl Soft Comput 18:178–184

Lee JH, Lee JH, Sohn SG, et al (2008) Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system. In: Proceedings of ICACT08, pp 1170–1175

Li B, Jiang WS (1997) Chaos optimization method and its application. Control Theory Appl 14(4):613–615

Peddabachigari S, Abraham A, Grosan C (2007) Modeling intrusion detection system using hybrid intelligent systems. J Netw Comput Appl 30(1):114–132

Schölkopf B, Smola A, Müller KR (1998) Nonlinear component analysis as a Kernel eigenvalue problem. Neural Comput 10(5):1299–1319

Shafi K, Abbass HA (2009) An adaptive genetic based signature learning system for intrusion detection. Expert Syst Appl 36(10):12036–12043

Shon T, Kim Y, Lee C, Moon J (2005) A machine learning framework for network anomaly detection using SVM and GA. In: Proceedings of IWIAS05, pp 176–183

Srinoy S (2007) Intrusion detection model based on particle swarm optimization and support vector machine. In: Proceedings of CISDA07, pp 186–192

Srivastava D, Bhambhu L (2010) Data classification using support vector machine. J Theor Appl Inf Technol 12(1):1–7

Stolfo S J, Fan W, Prodromidis A, et al (1999) KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed 22 December 2011

Tsai CF, Hsu YF, Lin CY, Lin WY (2009) Intrusion detection by machine learning: a review. Expert Syst Appl 36:11994–12000

Tsai CF, Lin CY (2010) A triangle area based nearest neighbors approach to intrusion detection. Pattern Recognit 43(1):222–229

Wang J, Hong X, Ren R, Li T (2009) A real-time intrusion detection system based on PSO-SVM. In: Proceedings of IWISA09, pp 319–321

Wang G, Hao JX, Ma J, Huang LH (2010) A new approach to intrusion detection using artificial neural networks and fuzzy clustering. Expert Syst Appl 37:6225–6232

Wang W, Battiti R (2006) Identifying intrusions in computer networks with principal component analysis. In: Proceedings of ARES06, pp 270–279

Wu SX, Banzhaf W (2010) Use of computational intelligence in intrusion detection systems: a review. Appl Soft Comput 10(1):1–35

Yang P, Zhu QS (2011) Finding key attribute subset in dataset for outlier detection. Knowl Based Syst 24(2):269–274

Zhang MH, Li G (2012) Network intrusion detection based on least squares support vector machine and chaos particle swarm optimization algorithm. J Converg Inf Technol 7(4):169–173

Zhang Z, Shen H (2005) Application of online-training SVMs for real-time intrusion detection with different considerations. Comput Commun 28(12):1428–1442