Applied Soft Computing 18 (2014) 178–184

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A novel hybrid KPCA and SVM with GA model for intrusion detection

Fangjun Kuang[a,b], Weihong Xu[a,c,*], Siyang Zhang[b]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210018, China
[b] Department of Electronic and Information Engineering, Hunan Vocational Institute of Safety & Technology, Changsha 410151, China
[c] College of Computer and Communications Engineering, Changsha University of Science and Technology, Changsha 410077, China

ARTICLE INFO

Article history:
Received 11 October 2012
Received in revised form
17 November 2013
Accepted 15 January 2014
Available online 30 January 2014

Keywords:
Intrusion detection
Kernel principal component analysis
Kernel function
Support vector machines
Genetic algorithm

ABSTRACT

A novel support vector machine (SVM) model combining kernel principal component analysis (KPCA) with genetic algorithm (GA) is proposed for intrusion detection. In the proposed model, a multi-layer SVM classifier is adopted to estimate whether the action is an attack, KPCA is used as a preprocessor of SVM to reduce the dimension of feature vectors and shorten training time. In order to reduce the noise caused by feature differences and improve the performance of SVM, an improved kernel function (N-RBF) is proposed by embedding the mean value and the mean square difference values of feature attributes in RBF kernel function. GA is employed to optimize the punishment factor C, kernel parameters σ and the tube size ε of SVM. By comparison with other detection algorithms, the experimental results show that the proposed model performs higher predictive accuracy, faster convergence speed and better generalization.

1. Introduction

Intrusion detection is one of the most essential things for security infrastructures in network environments, and it is widely used in detecting, identifying and tracking the intruders [1]. Capabilities of intrusion detection technologies have great importance with the performance of intrusion detection system (IDS). Researches always want to find an intrusion detection technology with better detection accuracy and less training time.

However, there are many problems in the traditional IDS, such as the low detection capability against the unknown network attack, high false alarm rate, and insufficient analysis capability and so on. In nature, intrusion detection can be seen as classification problem, to distinguish between the normal activities and the malicious activities. The concerned problems of machine learning are how the systems automatically improve the performance with the increase of experience, which is consistent with that of the IDS. Therefore, various machine learning methods are developed for intrusion detection, such as decision tree [1], genetic algorithm (GA) [2], neural network [3], principal component analysis (PCA) [4], fuzzy logic [5], K-nearest neighbor [6], rough set theory [7] and support vector machine (SVM) [8].

Among the methods mentioned above, SVM is an effective one, the main reason is that the distribution of different types of attacks is imbalanced, where the learning sample size of the low-frequent attacks is too small compared to the high-frequent attack. SVM is a margin-based classifier based on small sample learning with good generalization capabilities, which is frequently used in real world applications of classification [9]. 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. SVM has manifested its robustness and efficiency in the network action classification, and it is widely used in IDS as a popular method [10]. Eskin [11] addressed an unsupervised anomaly detection framework, and applied it in three unsupervised learning algorithms, including clustering method, K-nearest neighbor and SVM. Shon et al. [12] employed genetic algorithm (GA) for feature selection, and used SVM for intrusion detection. Srinoy [13] proposed an intrusion detection model using SVM and particle swarm optimization (PSO) which used PSO to extract intrusion features and SVM to classify. Fei et al. [14] proposed an incremental clustering method based on the density. Horng et al. [15] used the hierarchical clustering algorithm to provide the SVM with fewer, abstracted, and higher qualified training instances. To overcome the problem of uncertainty in IDS, Kavitha et al. [16] adopted a new technique known as neutrosophic logic (NL). Wu and Banzhaf [17] referred to the review of computational intelligence in intrusion

* Corresponding author at: School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210018, China.
Tel.: +86 18073101198.
E-mail addresses: kfjztb@126.com (F. Kuang), xwhxdcs@126.com (W. Xu), ztb731021@126.com (S. Zhang).

1568-4946/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2014.01.028

detection and applied numerical evaluation measures to quantify the performance of IDS. Kolias and Kambourakis [18] gave the survey of swarm intelligence in intrusion detection. Kuang et al. [19] proposed a SVM model based on kernel principal component analysis (KPCA) and GA, which used KPCA to extract intrusion features, and GA to optimize the parameter of SVM. Li et al. [20] put forward pipeline of data preprocess and data mining in IDS, and used gradually feature removal method to feature reduction and SVM to classify.

However, standard SVM still has some limitations, the performance depends on its parameters selection, and when the differences between the attributes of the sample are very big, using RBF in the training process will produce a large number of support vectors and the training time will be longer too. And two main parts should be conducted which are detection model set-up and intrusion feature extraction to get better performance.

To solve the above mentioned problems, we present a novel intrusion detection approach combining SVM and KPCA to enhance the detection precision for low-frequent attacks and detection stability. In the proposed method, KPCA maps the high dimension features in the input space to a new lower dimension eigenspace and extracts the principal features of the normalized data, and multi-layer SVM classifier is employed to estimate whether the action is an attack. In order to shorten the training time and improve the performance of SVM classification model, an improved radial basis kernel function (N-RBF) based on Gaussian kernel function is developed, and GA is used to optimize the parameters of SVM.

The rest of this paper is organized as follows. In Section 2, the proposed SVM classification model is described in detail. The classification procedure is presented to illustrate how to use the proposed SVM model for intrusion detection in Section 3.The experimental results are discussed in Section 4. Section 5 presents conclusion and future work.

2. Novel KPCA SVM classification model

2.1. Kernel principal component analysis

Principal component analysis (PCA) [21] is a common method applied to dimensionality reduction and feature extraction. PCA method can only extract the linear structure information in the data set, however, it cannot extract this nonlinear structure information. KPCA is an improved PCA, which extracts the principal components by adopting a nonlinear kernel method [22,23]. 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 learning [19]. The ith KPCA-transformed feature t_i can be obtained by

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

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

By 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 is the target value [19]. In the ε-SVM classification [24], 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. The ε-insensitive loss function denotes as follows:

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

Formally the optimization problem by requiring the follows:

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

where ξ_i and $\xi_i{}^*$ 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 $\alpha_i{}^*$ are the Lagrange multiplier coefficients for the ith training sample, and obtained by solving the dual optimization problem in support vector learning [24]. The training sample for which $\alpha_i \ne \alpha_i{}^*$ is corresponded to the support vectors, $K(k_i, k_j)$ 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, shown as follows, and any of those can be chosen to achieve the boundary function. Their detailed usages and descriptions, including parameters definitions, can be found in [25,26].

(1) Gaussian RBF kernel: $K(t_i, t_j) = \exp\left(\frac{-\left\|t_i - t_j\right\|^2}{\sigma^2}\right), \sigma \in R$

(2) Polynomial kernel: $K(t_i, t_j) = (a(t_i \cdot t_j) + b)^d, a \in R, b \in R, d \in N$

(3) Sigmoid kernel: $K(t_i, t_j) = \tanh(a(t_i \cdot t_j) + b), a \in R, b \in R$

(4) Inverse multi-quadric kernel: $K(t_i, t_j) = \frac{1}{\sqrt{\left\|t_i - t_j\right\|^2 + \sigma^2}}, \sigma \in R$

SVM always has good performance in classification when using RBF, which is an effective kernel function for fewer parameters set and an excellent overall performance. A network record contains dozens of attributes, and there may be significant differences between them. Therefore, when the differences between the attributes are very big, using RBF in the training process will produce a larger number of support vectors and the training time will be longer too.

In order to shorten the training time and improve the performance of SVM, an improved kernel function N-RBF is developed by embedding the mean value and the mean square difference values

of feature attributes in Gaussian RBF kernel function to normalize the attribute values. The N-RBF is defined as follows:

$$K(t_i, t_j) = \exp\left(\frac{-\left\|(t_i - m)/s - (t_j - m)/s\right\|^2}{\sigma^2}\right) \quad (5)$$

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 follows:

$$m_j = \frac{1}{n}\sum_{i=1}^{n} L_{ij}, j = 1, 2, \ldots, k \quad (6)$$

$$s_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(L_{ij} - m_j)^2}, j = 1, 2, \ldots, k \quad (7)$$

where L_{ij} is the jth attribute of the ith sample and 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, and it should be a positive definite kernel. Obviously, N-RBF is a positive definite kernel, and is a kernel function.

Consequently, the three positive parameters σ, ε and C are user-determined parameters in SVM classification model, the selection of the parameters plays an important role in the performance of SVM model. A better approach is to apply cross validation to select the best choices among some candidate parameters. Based on the idea, several disciplined approaches can be used to obtain the optimal parameters for SVM classification model, out of which, evolutionary method such as genetic algorithm (GA), simulated annealing algorithm and PSO algorithm, is one of the most widely used approaches. In this paper, GA is employed.

2.4. Optimizing the SVM model parameters with GA

GA is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms, as global search heuristics, are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. GAs have been considered with increasing interest in a wide variety of applications [27]. These algorithms are used to search the solution space through simulated evolution of "survival of the fittest". These are also used to solve linear and nonlinear problems by exploring all regions of state space and exploiting potential areas through mutation, crossover and selection operations applied to individuals in the population.

Therefore, in this paper, genetic algorithms are used to optimize the parameters σ, ε and C of SVM. And a negative mean absolute percentage error (MAPE) is used as the fitness function for evaluating fitness [24–26]:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{a_i - d_i}{a_i}\right| \times 100\% \quad (8)$$

where a_i and d_i represent the actual and forecast values, respectively. N is the number of classification periods. GA is used to yield a smaller MAPE by searching for better combinations of three parameters in SVM. The process of optimizing the SVM parameters with GA is shown in Fig. 1, which is described below.

Step 1: Encode SVM parameters and initialization of population. The three free parameters σ, ε and C are encoded in binary numbers and represented by a chromosome. Each bit of the chromosome represents whether the corresponding feature is selected or not. '1' in each bit means the corresponding feature is selected, whereas '0' means it is not selected. Randomly generate an initial population of chromosomes which represent the values of parameters in SVM model.

Step 2: Calculate the fitness function of each chromosome according to Eq. (8). It is evaluated by the cross-validated predictive accuracy of the SVM model.

Step 3: GA operators, which are selection, crossover and mutation. Selection is performed to select excellent chromosomes to reproduce. Based on fitness functions, chromosomes with higher fitness values are more likely to yield offspring in the next generation by means of the roulette wheel. The single-point crossover principle is employed. Segments of paired chromosomes between two determined break-points are swapped. Mutations are performed randomly by converting a '1' bit into a '0' bit or a '0' bit in to a '1' bit. The rates of crossover and mutation are probabilistically determined. In this study, the probabilities of crossover and mutation are set to 0.8 and 0.05, respectively.

Step 4: Generate a new population for the next generation. Offspring replaces the old population and forms a new population in the next generation by the three operations.

Step 5: Obtain the parameters of SVM model. If one of the stopping criteria (Generally, a sufficiently good fitness or a given number of generations) is satisfied, then the best chromosomes are presented as a solution, else go to step 2.

After these steps, the optimal parameters σ, ε and C of the KPCA SVM model are obtained.

3. Proposed SVM model for intrusion detection

3.1. Intrusion detection types and normalized

This paper takes the KDD CUP99 [28] as the datasets of the experiments. 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 Probe. Each network record contains 41 attributes, of which 34 attributes are continuous and 7 attributes are discrete. 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 transform all attributes into the normalized format.

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 [24]. '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 [24,25]. Obviously less two-class classifiers help to expedite the rate of training and recognition. Thus, 'Binary tree' SVM classification algorithm is adopted to construct detection model for intrusion detection.

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). With all the training samples of the five states, SVM1 is trained to separate the normal state from the intrusion state. When input of SVM1 is a sample representing the normal state, output of SVM1 is set to +1; otherwise −1. SVM2 is trained to separate the DoS from the other intrusion states. When the input of SVM2 is a sample

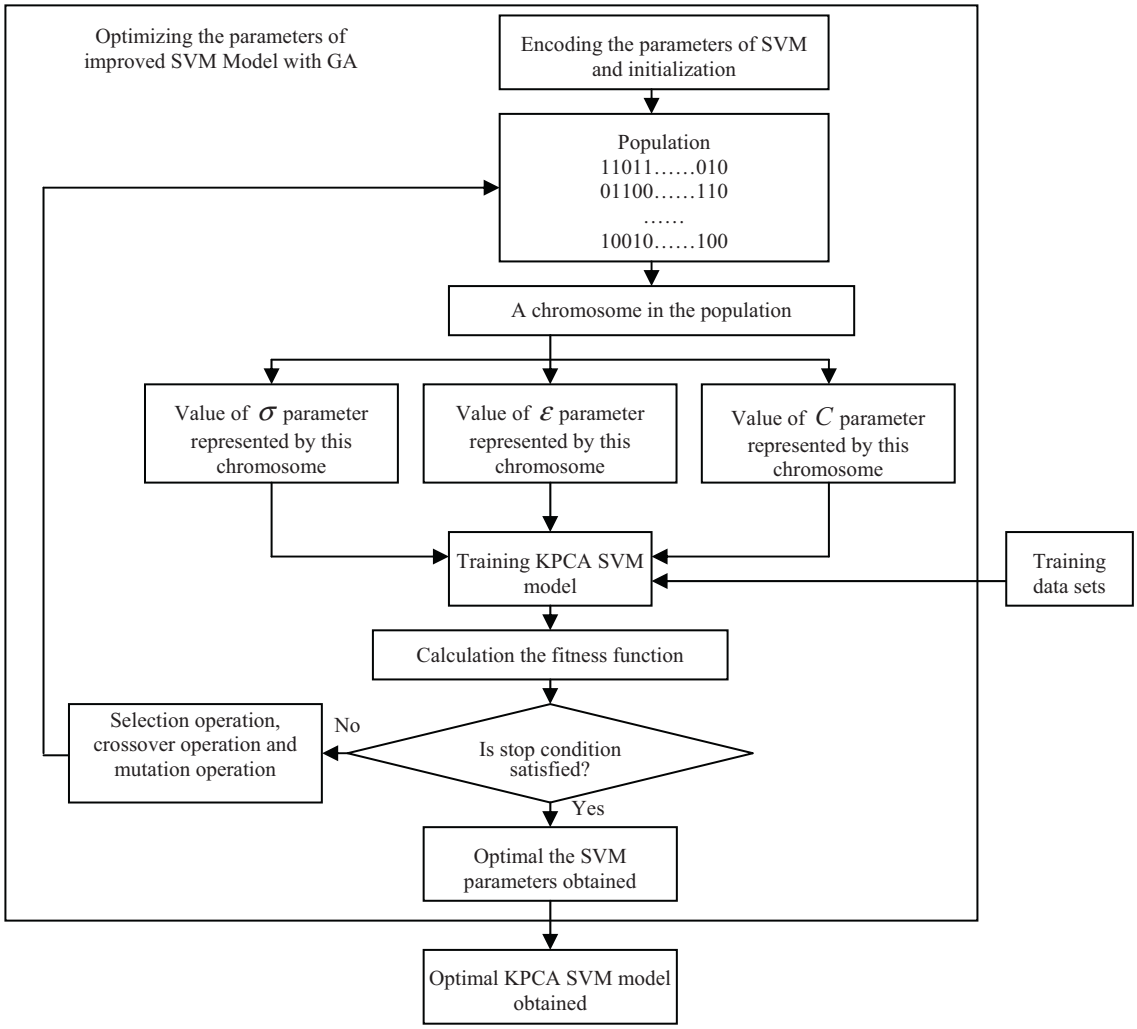

Fig. 1. Optimizing the parameters of improved KPCA SVM model with GA.

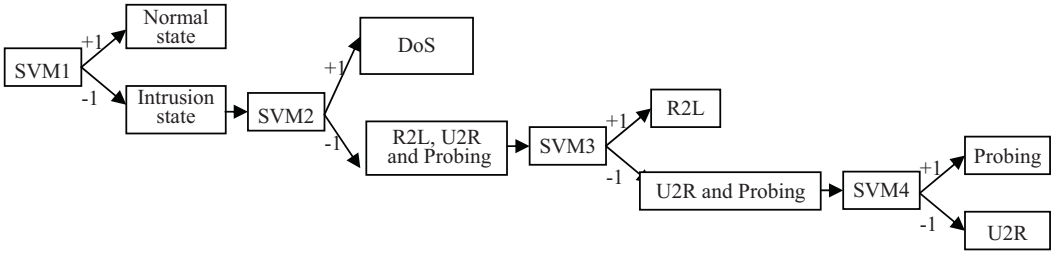

Fig. 2. The scheme of intrusion detection based on improved SVM model.

representing DoS, the output of SVM2 is set to +1; otherwise −1. SVM3 is trained to separate R2L from U2R and Probing. When the input of SVM3 is a sample representing the R2L, the output of SVM3 is set to +1; otherwise −1. SVM4 is trained to separate Probing from U2R. When the input of SVM4 is a sample representing Probing, the output of SVM4 is set to +1; otherwise −1. Thus, the multilayer SVM classifier is obtained. The basic principle of intrusion detection model based on improved SVM classifiers by combining KPCA and GA is shown in Fig. 2.

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

3.3. Proposed intrusion detection model implementation

Intrusion detection belongs to classification problems in essence, it distinguishes between the abnormal data and the normal data, and the 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 two 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. 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 used for KPCA and N-RBF kernels are also adopted for SVM, GA is used to select

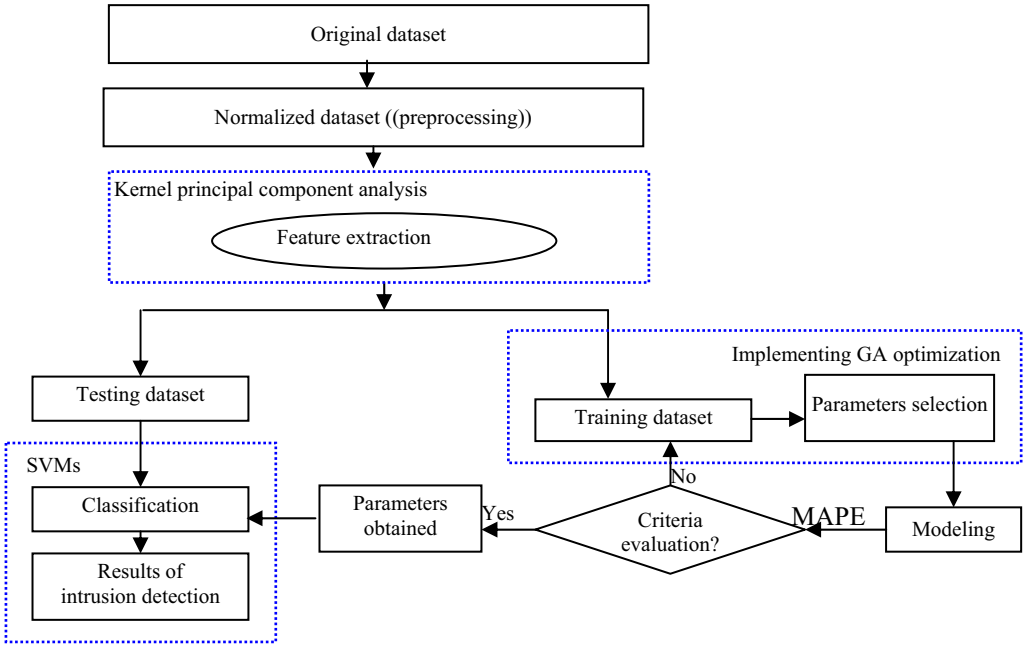

Fig. 3. The procedures of the proposed SVM model for intrusion detection.

the optimal parameter of SVM. Fig. 3 shows the procedures of the proposed SVM classification model for intrusion detection.

4. Experimental results and discussions

4.1. Experimental description

In this section, we selected samples from the subset of KDD to form the training and testing set. There are some performance indicators for the intrusion detection system as follows: *TP*, *FP*, *TN* and *FN*, where *TP* represents that the normal behavior is correctly forecasted, *FP* indicates that the abnormal behavior is judged as normal, *FN* denotes that the normal behavior is wrongly thought as abnormal, and *TN* represents the abnormal behavior is correctly detected [17].

(1) Detection rate: $DR = \frac{TN}{TN+FP}$

(2) False alarm rate: $FAR = \frac{FN}{FN+TP}$

(3) Correlation coefficient: $cc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$

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, we consider another indicator *cc*, which denotes the correlation between the forecast result and the actual situation. It ranges from −1 to 1, where 1 implies the forecast result is fully consistent with the actual situation and −1 is on behalf of a random prediction.

The experiment was processed within a MATLAB R2009b environment, which was running on a PC powered by Pentium IV 3.0 GHz CPU and 3.0 GB RAM.

4.2. Effectiveness of N-RBF

In this section, we selected samples from the subset of KDD to form the training and testing set. There were five data sets in Table 1. In order to verify the effectiveness of N-RBF, the percentage of the normal samples in each set was different. Adopt the SVM as the classifier. Because the choice of parameters would impact algorithms' performance directly, we used the python in Libsvm as a supplementary tool. The parameter σ of RBF and N-RBF is set to be 0.00028.The parameters *a*, *d* and *b* of POLY are set to be 0.00028, 3 and 1, respectively. The penalty parameter C of SVM is set to be 1024. The experiment results are shown in Table 2.

As shown in Table 2, the detection rates of N-RBF and POLY are higher, especially N-RBF. Due to the introduction of normalization, it reduces the noise among attributes, so the detection rate of N-RBF is higher than RBF. Secondly, the training time of RBF is dozens or even hundreds of times of N-RBF, while the training time of POLY is several or dozens of times of N-RBF, which indicate N-RBF has good performance in reducing the training time and only costs a few seconds; and the test time POLY and N-RBF cost is significantly less than RBF. In general, although the detection effect of N-RBF is not

Table 1
Five training and test sets.

No.	Training set			Test set		
	Normal (%)	Abnormal (%)	Total	Normal (%)	Abnormal (%)	Total
D1	83.5	16.5	12560	72.5	17.5	11040
D2	90.5	9.5	11050	35.0	65.0	11428
D3	55.3	44.7	9040	57.9	42.1	13818
D4	93.9	6.1	10640	85.8	14.2	11650
D5	76.5	23.5	6540	64.9	35.1	12318

Table 2
Comparison of the performance of the three kernel functions.

No.	SVM (Kernels)	DR (%)	FAR (%)	cc	TrD (s)	TeD (s)
D1	POLY	92.892	0.12	0.926	28.693	1.528
	RBF	89.256	0.06	0.912	456.062	158.67
	N-RBF	93.867	0.0825	0.968	3.896	1.821
D2	POLY	92.259	0.10	0.897	21.266	1.484
	RBF	93.551	0.025	0.914	489.625	158.63
	N-RBF	95.907	0.10	0.943	3.359	1.828
D3	POLY	97.731	0.188	0.978	33.75	1.797
	RBF	94.259	0.0375	0.951	174.375	116.625
	N-RBF	95.669	0.0625	0.962	4.734	3.11
D4	POLY	93.333	0.08	0.958	25.219	1.516
	RBF	84.909	0.04	0.908	431.062	155.67
	N-RBF	95.879	0.29	0.966	3.547	1.844
D5	POLY	93.122	0.625	0.938	16.016	1.531
	RBF	91.894	0.0375	0.937	61.172	103.937
	N-RBF	93.330	0.187	0.946	2.188	1.844

Table 3
Comparison of the detection rates of various categories.

Method/category	Normal	Probe	Dos	U2R	R2L
SVM (N-RBF)	0.9973	0.9862	0.8869	0.68	0.2484
SVM (RBF)	0.9980	0.1598	0.3628	0.04	0.0069
SVM (POLY)	0.9965	0.9532	0.5563	0.07	0.2391

greatly improved in comparison to the other two kernel functions, it has saved lots of time in the training process, which lays the foundation for the following experiments of the SVM.

The above experiments have not considered the attacks of different kinds, respectively. In order to further analyze the detection performance of N-RBF on unknown attacks, we gave the following experiments. First, regenerated a test set, containing more than 90% attacks of new categories. Then we compared SVM with N-RBF, SVM with RBF, and SVM with POLY in the experiments, and counted the detection rates of these methods on the attacks of all categories, the comparisons of experimental results in 30 simulation experiments are given in Table 3.

As shown in Table 3, we can see that the three methods show high detection rates on forecasting normal behaviors, and SVM with N-RBF has the highest detection rates on predicting the attacks of Probe and DoS. However, the results for detecting attacks of U2R and R2L are all unsatisfactory. In general, the detection rate of SVM with N-RBF on attacks of all categories is better than the other two methods.

4.3. Experiments of novel KPCA-GA-SVM

The following experiments were done to verify the effectiveness of the novel KPCA-GA-SVM (N-KPCA-GA-SVM). In this section, firstly, the subset we obtained in Section 4.2 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 test set. Secondly, randomly select 10 datasets from the training subset, named from F1 to F10, as the training set, and any two training sample sets did not intersect. Thirdly, from the test subset, selected the normal and attack records with the same number to form the test set.

Now, we evaluated N-KPCA-GA-SVM by comparing it with KPCA-GA-SVM [19], PCA-GA-SVM, 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 test time (TeD). We employed four SVMs for the 5-class classification problem including Section 3.2, and partitioned the data into the two classes of "Normal" and "Rest" (DoS, R2L, U2R, Probe) 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 this paper, we chose p eigenvectors by trial and error, which corresponded to the first p biggest eigenvalues, to form the sub-eigenspace, satisfying:

$$\frac{\sum_{i=1}^{p}\lambda_i}{\sum_{i=1}^{n}\lambda_i} \geq 90\% \tag{9}$$

In N-KPCA-GA-SVM model, N-RBF kernels were used for KPCA and N-RBF kernels were also adopted for SVM, GA method was used to select the optimal parameter of SVM and KPCA. KPCA was applied to feature extraction, this method aimed to map the high dimensional original input data to a lower dimensional eigensapce, which held the principal features and abandoned the subordinate and noise data. In N-KPCA-GA-SVM model, by many experiments, the parameters of the models were chosen as follows: population size: 50, maximal iteration: 200, the probabilities of crossover and mutation were set to 0.8 and 0.05, respectively. Through 20 simulation experiments, parameters $(C, \sigma, \varepsilon) = (83.5191, 0.0907, 0.0008)$ of SVM were obtained. In the other SVM model, all SVMs adopted RBF as their kernel function, and the parameters σ, ε and C were randomly selected. 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 4.

As shown in Table 4, the classification accuracies of the proposed SVM model are superior to those of SVM classifiers whose parameters are randomly selected, SVM classifier for intrusion detection by using PCA, KPCA to extract feature has a good performance in accuracy and runtime than that without feature extraction. The experimental results demonstrate that the features extracted by KPCA can provide more additional discriminatory information for improving classification performance than PCA. And dimension

Table 4
Experimental results among different algorithms.

Methods	Dataset	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
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	0.931	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	18.44	6.85	9.164	15.27	48.69	83.2	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
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)	13.844	18.864	19.093	5.625	22.672	28.146	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

reduction can improve the generalization performance and running time of SVM classifier. Furthermore, results also show that KPCA is better than PCA. The reason lies in the fact that KPCA can explore higher order information of the original inputs than PCA. By using the kernel method to generalize PCA into nonlinear, KPCA implicitly takes into account higher order information of the original inputs. More number of principal components can also be extracted in KPCA, eventually resulting in better generalization performance.

We can also see from Table 4 that the stabilities of the learning of N-KPCA-GA-SVM and KPCA-GA-SVM are better than Single-SVM. Compared to KPCA-GA-SVM, N-KPCA-GA-SVM is more effective in detecting, because DR and cc of N-KPCA-GA-SVM are higher than that of KPCA-GA-SVM. We can also see that Single-SVM needed longer training time, because it has to do cross-judging and more training, and the training time of KPCA-GA-SVM and PCA-GA-SVM is in the acceptable range. And N-KPCA-GA-SVM has obvious advantages in training time over KPCA-GA-SVM. It is apparent that N-KPCA-GA-SVM needed less test time than another three algorithms. 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 by using the optimum method to tune the network parameters and connecting weights such as Genetic algorithm. In addition, this table can also see that the overall performance of N-KPCA-GA-SVM model is better than other four methods for intrusion detection.

The above results show that N-RBF and KPCA play some role in saving the training and test time. N-KPCA-GA-SVM is more reliable, compared to KPCA-GA-SVM, PCA-GA-SVM, Single-SVM and RBFNN. And it does not cause large fluctuations in detection performance. Moreover, it can improve the detection performance.

5. Conclusion

In this paper, a Novel hybrid KPCA SVM with GAs model is proposed for intrusion detection. In N-KPCA-GA-SVM model, KPCA is adopted to extract the principal features of intrusion detection data, and multi-layer SVM classifier is employed to estimate whether the action is an attack. N-RBF kernel function based on Gaussian kernel function is developed to shorten the training time and improve the performance of SVM classification model, GA is used to select suitable parameters for SVM classifier, which avoids over-fitting or under-fitting of the SVM model occurring because of the improper determination of these parameters. The experimental results show that the classification accuracies of the proposed KPCA SVM model are superior to those of SVM classifiers whose parameters are randomly selected, and SVM classifier by feature extraction using PCA, KPCA can achieve better generalization performance than that without feature extraction. Furthermore, the experimental results also show that on intrusion detection data, KPCA perform is better than PCA. The reason lies in the fact that KPCA can explore higher order information of the original inputs than PCA. By using the kernel method to generalize PCA into nonlinear, KPCA also implicitly takes into account higher order information of the original inputs. More number of principal components can also be extracted in KPCA, eventually resulting in better generalization performance.

For future work, we want to develop more algorithms of combining kernel methods with some other classification methods for pattern analysis and online intrusion detection, and research some other optimization algorithm for SVM parameters optimization.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61373063), the Science and Technology Department of Hunan Province of China (No. 2012SK4046, 2012FJ3005, and 2013FJ4217), the Research Foundation of Education Bureau of Hunan Province of China (No. 13C086, 12C0983). And the authors are grateful to the anonymous reviewers for valuable suggestions and comments, which are very helpful for improvement of this paper.

References

[1] J.H. Lee, J.H. Lee, S.G. Sohn, et al., Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system, in: 10th International Conference on Advanced Communication Technology (ICACT'08), 2008, pp. 1170–1175.
[2] K. Shafi, H.A. Abbass, An adaptive genetic based signature learning system for intrusion detection, Expert Syst. Appl. 36 (10) (2009) 12036–12043.
[3] G. Wang, J.X. Hao, J. Ma, L.H. Huang, A new approach to intrusion detection using artificial neural networks and fuzzy clustering, Expert Syst. Appl. 37 (2010) 6225–6232.
[4] W. Wang, R. Battiti, Identifying intrusions in computer networks with principal component analysis, in: Proceedings of the First International Conference on Availability Reliability and Security (ARES'06), 2006, p. 270-279.
[5] W. Chimphlee, A.H. Addullah, M.N.M. Sap, S. Srinoy, S. Chimphlee, Anomaly-based intrusion detection using fuzzy rough clustering, in: Paper Presented at the International Conference on Hybrid Information Technology (ICHIT'06), 2006, p. 329-334.
[6] C.F. Tsai, C.Y. Lin, A triangle area based nearest neighbors approach to intrusion detection, Pattern Recogn. 43 (1) (2010) 222–229.
[7] P. Yang, Q.S. Zhu, Finding key attribute subset in dataset for outlier detection, Knowl. Syst. 24 (2) (2011) 269–274.
[8] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering, Int. J. Very Data Bases 16 (2007) 507–521.
[9] A. Majid, A. Khan, A.M. Mirza, Combining support vector machines using genetic programming, Int. J. Hybrid Intell. Syst. 3 (2) (2006) 109–125.
[10] C.F. Tsai, Y.F. Hsu, C.Y. Lin, W.Y. Lin, Intrusion detection by machine learning: a review, Expert Syst. Appl. 36 (2009) 11994–12000.
[11] E. Eskin, Anomaly detection over noisy data using learned probability distributions, in: Proceedings of the International Conference on Machine Learning, 2000, p. 255-262.
[12] T. Shon, Y. Kim, C. Lee, J. Moon, A machine learning framework for network anomaly detection using SVM and GA, in: Proceedings of 3rd IEEE International Workshop on Information Assurance and Security, New York, USA, 2005, pp. 176–183.
[13] S. Srinoy, Intrusion detection model based on particle swarm optimization and support vector machine, in: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA'07), 2007, p. 186-192.
[14] R. Fei, L. Hu, H. Liang, Using density-based incremental clustering for anomaly detection, in: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, 2008, p. 986-989.
[15] S.J. Horng, M.Y. Su, Y.H. Chen, T.W. Kao, et al., A novel intrusion detection system based on hierarchical clustering and support vector machines, Expert Syst. Appl. 38 (2011) 306–313.
[16] B. Kavitha, S. Karthikeyan, P.S. Maybell, An ensemble design of intrusion detection system for handling uncertainty using neutrosophic logic classifier, Knowl. Syst. 28 (2012) 88–96.
[17] S.X. Wu, W. Banzhaf, Use of computational intelligence in intrusion detection systems: a review, Appl. Soft Comput. 10 (1) (2010) 1–35.
[18] C. Kolias, G. Kambourakis, M. Maragoudakis, Swarm intelligence in intrusion detection: a survey, Comput. Security 30 (2011) 625–642.
[19] F.J. Kuang, W.H. Xu, S.Y. Zhang, et al., A novel approach of KPCA and SVM for intrusion detection, J. Comput. Inf. Syst. 8 (8) (2012) 3237–3244.
[20] Y.H. Li, J.B. Xia, S.L. Zhang, et al., An efficient intrusion detection system based on support vector machines and gradually feature removal method, Expert Syst. Appl. 39 (2012) 424–430.
[21] I.T. Jolliffe, Principle Component Analysis, Springer-Verlag, New York, 1986.
[22] Z.G. Chen, H.D. Ren, X.J. Du, Minimax probability machine classifier with feature extraction by kernel PCA for intrusion detection, Wireless Communications, Netw. Mobile Comput. (2008) 1–4.
[23] M. Ding, Z. Tian, H. Xu, Adaptive kernel principal analysis for online feature extraction, Proc. World Acad. Sci. Eng. Technol. 59 (2009) 288–293.
[24] D. Srivastava, L. Bhambhu, Data classification using support vector machine, J. Theoret. Appl. Inf. Technol. 12 (1) (2010) 1–7.
[25] C.W. Hsu, C.C. Chang, C.J. Lin, A Practical Guide to Support Vector Classification [EB/OL], 2010 http://www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf
[26] R. Chen, D.Y. Sun, D.T. Qin, F.B. Hu, A novel engine identification model based on support vector machine and analysis of precision-influencing factors, J. Cent. South Univ. Technol. 41 (4) (2010) 1391–1397.
[27] G.E. Goldberg, Genetic Algorithms in Search Optimization Machine Learning, Addison-Wesley, New York, USA, 2005.
[28] S.J. Stolfo, W. Fan, A. Prodromidis, et al., KDD Cup 1999 Data [EB/OL] (2011) http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html