Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **2**, Number **2**, May **2017**

A Novel Network Intrusion Detection Based on Support Vector Machine and Tent Chaos Artificial Bee Colony Algorithm

Fang-jun Kuang

School of information engineering
Wenzhou Business College
Wenzhou City, Zhejiang Province, 325035, China
kfjzyf@126.com

Si-yang Zhang

School of information engineering
Wenzhou Business College
Wenzhou City, Zhejiang Province, 325035, China
kfjzyf@126.com

Received February 2017; revised April 2017

Abstract. *A novel network intrusion detection model by combining support vector machine (SVM) and Tent chaos artificial bee colony algorithm (TCABC) is proposed in the paper. The proposed method, in which a multi-layer SVM classifier is adopted to estimate whether the action is an attack, KPCA is applied as a preprocessor of SVM to reduce the dimension of feature vectors, and TCABC is employed to optimize kernel parameters σ, tube size ε and punishment factor C of SVM. The experimental results demonstrate that the improved intrusion detection model has higher detection accuracy and faster computational time, and it can also shorten the training time.*
Keywords: Network intrusion detection, Support vector machine, Kernel principal component analysis, Tent chaos search, Artificial bee colony algorithm

1. **Introduction.** With the development of computer and communication technologies, network security has been a challenge for both the researchers and enterprises. Intrusion detection system (IDS) is one of the key methods to protect network security [1]. Researches always want to find an intrusion detection technology with less computational time and better detection accuracy.

Network intrusion detection can be seen as a classification problem, to distinguish between the normal activities and the malicious activities. Therefore, various machine learning methods are developed to build the intrusion detection model had got better performance than the traditional intrusion detection technologies, such as neural network [2], K-nearest neighbor [3], rough set theory [4] and support vector machine (SVM) [5, 6]. Among the methods mentioned above, SVM is an effective one, which is a well-known classifier tool based on small sample learning. SVM has manifested its robustness and efficiency in the network action classification, it therefore becomes a popular method widely used in IDS [7].

To get better performance of intrusion detection, we propose a novel approach for network intrusion detection. In the proposed method, use kernel principal component analysis (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 employ multi-layer SVM classifier to estimate whether the action is an attack. Tent chaos artificial bee colony algorithm (TCABC) is proposed to optimize the parameters of SVM.

The remainder of this paper is organized as follows. The SVM classification model is described in Section 2. In Section 3, how to use the proposed SVM model for intrusion detection is illustrated in detail. The experimental results and discussions are presented in Section 4. Section 5 lists the conclusions and potential future work.

2. Related Work & Contributions.

2.1. **Kernel principal component analysis.** Principal component analysis (PCA) is a common method applied to dimensionality reduction and feature extraction [8]. PCA method only can extract the linear structure information in the data set but can not 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 [9]. 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 $x_1, x_2, \cdots, x_n \in R^m$ be the training samples for KPCA [6]. The ith KPCA-transformed feature can be obtained by (1).

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

Here, Column vector $\gamma_i (i = 1, 2, \cdots, p; 0 < p \leq n)$ 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. 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 $(t_1, y_1), (t_2, y_2), \cdots, (t_p, y_p)$, $t_i \in R^n$ $(n < m)$ is the transformed input vector, $y_i \in \{-1, +1\}$ is the target value. In the $\varepsilon-$SVM classification [10], the goal is to find a function $f(t)$ that has at most deviation from the actually obtained targets y_i for all the training data, and at the same time, is as flat as possible [6]. The decision function takes the following form:

$$f(t) = \sum_{i=1}^{p} (\alpha_i - \alpha_i^*) K(t_i, t_j) + b \tag{2}$$

where $K(t_i, t_j)$ is a kernel function, b is found by the Karush-Kuhn-Tucker conditions at optimality, α_i and α_i^* are the Lagrange multiplier coefficients for the ith training example of regression, and obtained by solving the dual optimization problem in support vector learning [10], the non-negative coefficients α_i and α_i^* are bounded by a user-specified constant C. In the SVM, there are some common kernels, and any of those can be chosen to achieve the boundary function. 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:

$$K(t_i, t_j) = \exp(\frac{-\|t_i - t_j\|^2}{\sigma^2}), \sigma \in R \tag{3}$$

Consequently, C 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. In this paper, employed is the TCABC algorithm.

2.3. **Tent Chaos Artificial Bee Colony Algorithm.**

2.3.1. *Artificial Bee Colony Algorithm.* ABC algorithm was applied to multidimensional and multimodal function optimization in [11]. The swarm is divided into employed bees, scouts and onlookers. In the initialization phase, the algorithm generates a group of food sources corresponding to the solutions in the search space. The food sources are produced randomly within the range of the boundaries of the variables.

$$x_{i,j} = x_j^{\min} + R(x_j^{\max} - x_j^{\min}) \tag{4}$$

where $i = 1, 2, \cdots, SN$, $j = 1, 2, \cdots, D$. SN is the number of food sources and equals to half of the colony size. D is the dimension of the problem, representing the number of parameters to be optimized. $x_j^{\min}$ and $x_j^{\max}$ are lower and upper bounds of the jth parameter, respectively. The fitness of food sources will be evaluated. Additionally, counters which store the numbers of trials of each bee are set to 0 in this phase. In the employed bees phase, a number of employed bees, set as the number of the food sources and half the colony size, are used to find new food sources using (5)

$$v_{i,j} = x_{i,j} + \Phi_{i,j}(x_{i,j} - x_{k,j}) \tag{5}$$

where $i = 1, 2, \cdots, SN$, j and is a randomly selected number in $[1, D]$, D is the number of dimensions. $\Phi_{i,j}$ is a random number uniformly distributed in the range$[-1, 1]$. k is the index of a randomly chosen solution, where $k \neq i$. Both V_i and X_i are then compared against each other and the employed bee exploits the better food source.

Onlooker bees choose a random food source according to probability $P_i(t) = \frac{fit_i(t)}{\sum_{n=1}^{SN} fit_n(t)}$, $fit_i(t)$ is the fitness of the ith food source. Then, each onlooker bee tries to find a better food source around the selected one using (4). If a food source cannot be improved for a predetermined number of cycles, referred to as $Limit$, this food source is abandoned. The employed bee that was exploiting this food source becomes a scout that looks for a new food source by randomly searching the problem domain.

2.3.2. *Tent Chaos Map.* Similar to other evolutionary algorithms, artificial bee colony still has the premature convergence problem. Therefore, chaotic search strategy has been applied in the ABC algorithm to improve the ability to search global optimal solution [12]. Howeverthe Tent-map shows outstanding advantages and higher iterative speed than the Logistic map [13, 14]. So the Tent-map is used in chaos optimization to generate the chaotic series in this study. Consider the equation of Tent-map:

$$cx_{t+1} = \begin{cases} 2cx_t, & 0 \leq cx_t \leq 1/2, \\ 2(1 - cx_t), & 1/2 \leq cx_t \leq 1. \end{cases} \tag{6}$$

where $cx_t \notin \{0.2, 0.4, 0.6, 0.8\}$.

2.3.3. *Chaotic Opposition-based Learning Initialization.* Population initialization is a crucial task in evolutionary algorithms because it can affect the convergence speed and the quality of the final solution. At the same time, according to [14], replacing the random initialization with the opposition-based population initialization can get better initial solutions and accelerate convergence speed. So this paper proposes a novel initialization approach which employs the Tent chaotic map and the opposition-based learning method

to generate initial population. The chaotic opposition-based learning population initialization is described as follows:

Step 1: Set the maximum number of chaotic iteration $C_{\max}$, the population scale .

Step 2: Randomly generate initialize variables $cx_{0,j} \in (0,1)$ except 0.2, 0.4, 0.6, and 0.8, and calculate chaotic variables $cx_{k,j}(k = 1, 2, \cdots, C_{\max}, j = 1, 2, \cdots, D)$ for next iteration using (6).

Step 3: Generate the initialization solution $x_{\mathrm{i},j}$ using $x_{i,j} = x_j^{\min} + cx_{k,j} \times (x_j^{\max} - x_j^{\min})$, $i = 1, 2, \cdots, SN$, $j = 1, 2, \cdots, D$,$k = 1, 2, \cdots, C_{\max}$.

Step 4: Calculate the opposition solution $ox_{\mathrm{i},j}$ using $ox_{i,j} = x_j^{\min} + x_j^{\max} - x_{i,j}$.

Step 5: Selecting SN fittest individuals from set the $\{x\}_{i=1}^{SN} \cup \{ox\}_{i=1}^{SN}$ as initial population.

2.3.4. *Tent Chaotic Search.* The effect of chaotic local search is aimed to utilize the Tent map to explore a better solution near the . Tent chaotic local search of the TCABC algorithm increases the ability to avoid local optima, and reduces the computation time. The detail procedure of chaotic local search is described as follows:

Step 1: Find the best solution named $X_{best} = (x_{k,1}, \cdots, x_{k,D})$, and calculate the fitness of X_{best}.

Step 2: Set the iteration count = 0 and generate the initial chaotic vector distribute in (0, 1) using (7).

$$z_{k,j}^0 = (x_{k,j} - x_j^{\min})/(x_j^{\max} - x_j^{\min});\ k = 1, 2, \cdots, SN;\ j = 1, 2, \cdots, D \tag{7}$$

Step 3: Calculate chaotic variables $z_{k,j}^m (m = 1, 2, \cdots, C_{\max})$ for next iteration using (6).

Step 4: Convert the chaotic variables $z_{k,j}^m$ to the decision variables and generate new solution using (8).

$$v_{k,j} = x_{k,j} + \frac{x_j^{\max} - x_j^{\min}}{2} \times (2z_{k,j}^m - 1) \tag{8}$$

Step 5: Calculate the fitness of V_k and compare it to the X_{best}, if the fitness of V_k is better than the fitness of X_{best}, the solution should be selected as the new X_{best}.

Step 6: $count = count + 1$, if the maximum iteration cycle is not reached yet, then go to step 2. Otherwise, chaotic search is completed.

2.3.5. *Tournament Selection.* The proportional selection in ABC algorithm requires the fitness function greater than zero. However, tournament selection [15] is different, its a selection process based on local competition which only refers to the relative value of individuals. In this paper, we select two individuals from the population and compare their fitness values, then assign one score to a better individual of the two, repeat such process and then the individual with the highest values wins the heaviest weight. Tournament selection probability is as follow:

$$P_i(t) = \frac{c_i(t)}{\sum_{i=1}^{N} c_i(t)} \tag{9}$$

where c_i is the score of an individual.

2.4. **Optimizing the parameters of SVM model with TCABC.** By means of the TCABC algorithm, the three major parameters C, σ and ε of SVM model, can be optimized, which a potential solution is 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 prediction model, namely, minimize the errors between the true values and forecasting values of the testing samples. Therefore,

put the three parameters values into the SVM model and calculate the forecasting error. Therefore, the fitness function (MSE) can be defined as follows.

$$Fitness = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{m}\sum_{j=1}^{m}(f(x_{ij}) - y_{ij})^2} \tag{10}$$

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 bee with the minimal fitness value will outperform others and should be reserved during the optimization process. Accordingly, the optimal parameters can be selected. The detail procedure of the TCABC algorithm for the parameters selection of SVM model (KPCA-TCABC-SVM) can be described as follows:

Step 1: Initial the food sources and computation conditions include population of bee colony N, number of employed bees $SN = (N/2)$, upper and lower boundaries of every decision variable, the maximum iteration G_{max}, $Limit$ and chaotic local search iteration number C_{max}, the number of parameters D is set as 3 in this study.

Step 2: Set iteration $iter = 0$, generate the SN vectors X_i with D dimensions as food sources according to chaotic opposition-based learning initialization method.

Step3Sent SN employed bees to food sources. Initialize the flag vector $trial(i) = 0$, which is recorded the cycle number of a food source.

Step 4: Produce new solutions V_i using employed bees by (5), and calculate the fitness value using (10).

Step 5: If the fitness value of V_i is better than that of X_i, then $X_i = V_i$,$trial(i) = 0$; Else X_i is maintained, $trial(i) = trial(i) + 1$.

Step 6: Calculate the probability values P_i of food sources by (9) applying tournament selection.

Step 7: Onlooker bees choose the food sources by probabilities P_i until all of them have a corresponding food source, and produce new solutions V_i . Calculate the fitness value using (10).

Step 8: If the fitness value of V_i is better than that of X_i, then $X_i = V_i$,$trial(i) = 0$; Else X_i is maintained, $trial(i) = trial(i) + 1$.

Step 9: If $trial(i) > Limit$, there is an abandoned solution for the scout then replace it with a new food source V_i, which will be reinitialized by carrying out Tent chaotic search.

Step 10: Memorize the best solution found so far.

Step 11: Update $iter = iter + 1$. If the maximum iteration cycle is not reached yet, then go to step 4.

Step 12 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 [16]. 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 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.

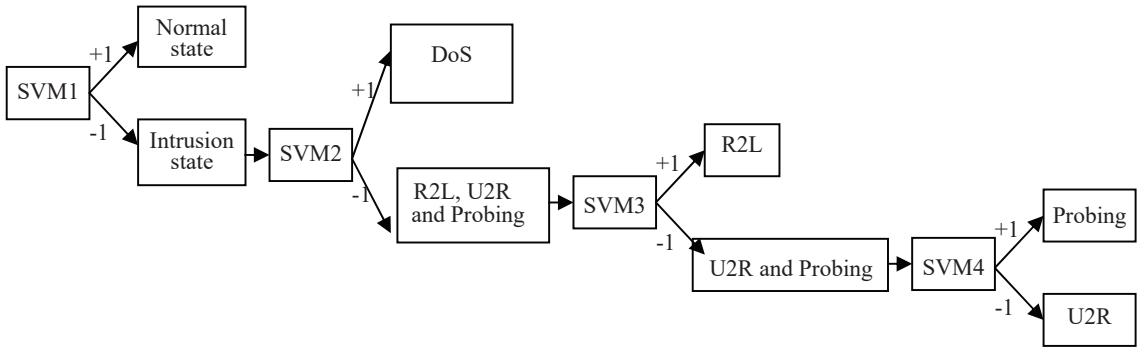

FIGURE 1. The scheme of intrusion detection based on improved SVM and TCABC

3.2. **Intrusion detection base 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 [6]. 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 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 and TCABC is shown in figure 2.

All the four SVMs adopt the RBF function as their kernel function, the parameters C, σ and ε are optimized with TCABC. 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, TCABC 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 TCABC 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 $\sum_{i=1}^{p} \lambda_i \Big/ \sum_{i=1}^{n} \lambda_i \geq 90\%$.The second stage is to use this attribute subset as the training dataset and testing dataset of SVM to perform the classification, and RBF kernels are also adopted for KPCA and SVM, TCABC method is used to select the optimal parameter of SVM. In the third stage, the optimal parameters are fed to SVM for classification. The flowchart of KPCA-TCABC-SVM classification model for intrusion detection is shown in figure 2.

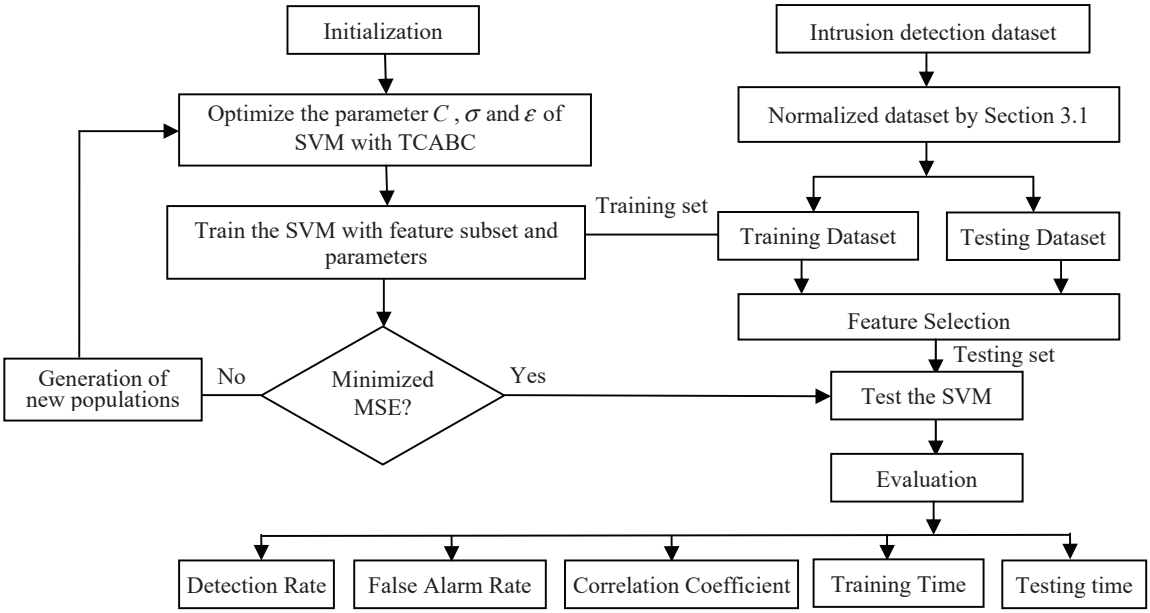

FIGURE 2. The flowchart of the proposed model for intrusion detection

4. **Experimental Results and Discussions.**

4.1. **Experimental description.** There are some performance indicators for the intrusion detection system as follows: TP, FP, TN and FN, where TP represents the abnormal behavior is correctly forecasted, FP represents the normal behavior is wrongly judged as abnormal, FN represents the abnormal behavior is wrongly thought as normal, and TN represents the normal behavior is correctly detected [6]. Detection Rate is counted by $DR = TP/(TP + FP)$, False Alarm Rate is counted by $FAR = FP/(FP + TN)$, Correlation Coefficient is counted by $cc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$.

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 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-TCABC-SVM. The purpose of KPCA-TCABC-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 on a PC powered by Interl(R) Core(TM) i7-6700k 4.0 GHz CPU and 8.0 GB RAM.

4.2. **Experiments of KPCA-TCABC-SVM.** 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.

The following experiments were done to verify the effectiveness of KPCA-TCABC-SVM. In this section, firstly, 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 test set. Secondly, randomly select 10 datasets from the training subset, named from $F1$ to F10, as the training set, and

TABLE 1. Five training and testing sets

No.	Training set			Training 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

any two training sample sets did not intersect. Thirdly, from the test subset, select the normal and attack records with the same number to form the test set. Now, we evaluated KPCA-TCABC-SVM by comparing it with KPCA-ICPSO-SVM [6], KPCA-GA-SVM, PCA-PSO-SVM and Single-SVM, 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 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 the proposed KPCA-TCABC-SVM model, RBF kernels were used for KPCA and RBF kernels were also adopted for SVM, TCABC 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 the proposed KPCA-TCABC-SVM model, by many experiments, the parameters of the models were chosen as follows: swarm size: 50 , maximal iteration: 200,$C_{max} = 300$, $Limit = 100$. Through 50 simulation experiments, the parameters $(\mathrm{C},\sigma,\varepsilon) = (2245.2517, 1.0246, 0.00013)$ of SVM were obtained. The experiment results among different algorithms were listed in Table 2.

As shown in Table 2, we could see that the learning stabilities of KPCA-TCABC-SVM were better than the other four algorithms. Compared to PCA-PSO-SVM, KPCA-ICPSO-SVM was more effective in detecting, because DR and cc of KPCA-ICPSO-SVM were higher than PSO-SVM. We could also see that Single-SVM needs longer training time, because it had to do cross-judging and more training, and the training time of KPCA-GA-SVM and PSO-SVM was in the acceptable range. Table 2 shows that the classification accuracies of the proposed KPCA-TCABC-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. It was apparent that KPCA-TCABC-SVM needed less testing time than the other four algorithms. The above results showed that RBF and TCABC algorithms played some role in saving the training and testing time. Compared other four algorithms, KPCA-TCABC-SVM had more excellent detection performance, and also saved a lot of training and testing time.

5. **Conclusions.** In this paper, a novel network intrusion detection model by combining support vector machine (SVM) and Tent chaos artificial bee colony algorithm (TCABC) is proposed. In the KPCA-TCABC-SVM model, KPCA is adopted to extract the principal

TABLE 2. Experiment results among different algorithms

Methods \ Dataset		$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$	$F8$	$F9$	$F10$
KPCA-TCABC-SVM	DR(%)	95.818	96.714	96.286	96.003	97.138	96.969	97.243	96.884	96.201	97.125
	FAR(%)	0.974	0.883	0.969	0.973	1.002	0.846	0.986	0.854	1.025	0.851
	cc	0.965	0.970	0.968	0.962	0.973	0.975	0.968	0.976	0.967	0.981
	TrD(s)	0.627	1.421	1.150	1.001	0.384	0.421	0.921	0.657	0.453	0.623
	TeD(s)	1.781	5.513	3.012	2.447	1.016	1.013	0.921	1.394	1.246	1.015
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-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-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
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

features of the intrusion detection data, and multi-layer SVM classifier is employed to estimate whether the action is an attack. TCABC is used to select suitable parameters for SVM classifier. The experimental results show that the proposed method has 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.

6. **Acknowledgement.** This work was supported in part by the National Natural Science Foundation of China under Grant 61373063 and 61402227.

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,*International Conference on Advanced Communication Technology*, pp. 1170-1175, 2008.

[2] G. Wang, J. X. Hao, J. Ma, L. H. Huang, A new approach to intrusion detection using artificial neural networks and fuzzy clustering,*Expert Systems With Applications*, vol. 37, pp. 6225-6232, 2010.

[3] C. F. Tsai, C. Y. Lin, A triangle area based nearest neighbors approach to intrusion detection, *Pattern Recognition*, vol. 43, no. 1, pp. 222-229, 2010.

[4] Yang P, Zhu Q S, Finding key attribute subset in dataset for outlier detection, *Knowledge-Based Systems*, vol. 24, no. 2, pp. 269-274, 2011.

[5] 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 Systems With Applications*, vol. 38, pp. 306-313, 2011.

[6] F. J. Kuang, S.Y. Zhang, Z. JIN, A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection, *Soft Computing*, vol. 19, no. 5, pp. 1187-1199, 2015.

[7] C. F. Tsai, Y. F. Hsu, C. Y. Lin, W. Y. Lin, Intrusion detection by machine learning: A review, *Expert Systems With Applications*, vol. 36, pp. 11994-12000, 2009.

[8] I. T. Jolliffe, Principle Component Analysis, *Springer-Verlag*, New York, 1986.

[9] M. Ding, Z. Tian, and H. Xu, Adaptive Kernel Principal Analysis for Online Feature Extraction, *Proceedings of World Academy of Science, Engineering and Technology*, vol. 59, pp. 288-293, 2009.

[10] D. Srivastava, L. Bhambhu, Data classification using support vector machine, *Journal of Theoretical & Applied Information Technology*, vol. 12, no. 1, pp. 1-7, 2010.

[11] D. Karaboga, B. Akay. A modified artificial bee colony (ABC) algorithm for constrained optimization problems, *Applied Soft Computing*, vol. 11, no. 3, pp. 3021-3031, 2011.

[12] W. F. Gao, S.Y. Liu, A modified artificial bee colony algorithm, *Computers & Operations Research*, vol. 39, pp. 687C697, 2012.

[13] L. Shan, H. Qiang, J. Li, et al., Chaotic optimization algorithm based on Tent map, *Control and Decision*, vol. 20, no. 2, pp. 179-182, 2005.

[14] F. J. KUANG, Z. JIN, W. H. XU et al., Hybridization algorithm of Tent chaos artificial bee colony and particle swarm optimization, *Control and Decision*, vol. 30, no. 5, pp. 839-847, 2015.

[15] . L. Bao, J. C. Zeng, Comparison and analysis of the selection mechanism in the artificial bee colony algorithm, *9th International Conference on Hybrid Intelligent Systems*, Los Alamitos, CA: IEEE Computer Society, 2009: 411- 416.

[16] S. J. Stolfo, W. Fan, A. Prodromidis, et al., KDD cup 1999 dataset, 2010. *http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html*