Knowledge-Based Systems 40 (2013) 1–6

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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

Anomaly intrusion detection based on PLS feature extraction and core vector machine

Gan Xu-sheng [a,*], Duanmu Jing-shun [b], Wang Jia-fu [c], Cong Wei [c]

[a] XiJing College, Shaanxi, Xi'an 710123, China
[b] Equipment Management and Safety Engineering College, Air Force Engineering University, China
[c] Engineering College, Air Force Engineering University, China

ARTICLE INFO

Article history:
Received 25 April 2012
Received in revised form 13 September 2012
Accepted 13 September 2012
Available online 23 October 2012

Keywords:
Core vector machine
Partial least square
Feature extraction
Anomaly intrusion detection
Support Vector Machine

ABSTRACT

To improve the ability of detecting anomaly intrusions, a combined algorithm is proposed based on Partial Least Square (PLS) feature extraction and Core Vector Machine (CVM) algorithms. Principal elements are firstly extracted from the data set using the feature extraction of PLS algorithm to construct the feature set, and then the anomaly intrusion detection model for the feature set is established by virtue of the speediness superiority of CVM algorithm in processing large-scale sample data. Finally, anomaly intrusion actions are checked and judged using this model. Experiments based on KDD99 data set verify the feasibility and validity of the combined algorithm.

1. Introduction

With the development of computer technology and the popularity of internet, network security is attracting more and more attention. Intrusion detection, a new network security mechanism for monitoring, preventing and resisting intrusions, plays a very important role in ensuring network security. Due to its superiority in detecting new and unknown attacks, anomaly intrusion detection has become a hot research topic in the field of intrusion detection technology [1–3].

Anomaly intrusion detection is essentially a classification problem, namely, to classify given data into normal data and abnormal data. In recent years, in order to improve the detection efficiency, some artificial intelligence algorithms, such as Immune Algorithm [4], Rough Set [5] and Neural Network [6,7], have been adopted to solve the intrusion detection problem. The learning speed of these methods is too slow for intrusion detection, and the detection accuracy also need to be improved. Support Vector Machine (SVM) is a novel pattern recognition technique based on Statistic Learning Theory. SVM can be classified as L1-SVM and L2-SVM according to the loss function. SVM obeys the principle of Structure Risk Minimization to maximize the generalization ability of the learning machine. In addition, because SVM solves an essentially convex optimization problem, the obtained locally optimal solution must be a globally optimal one [8]. Compared with conventional intrusion detection methods, SVM has obvious advantages in detection accuracy, operation speed and excessive matching. This guarantees that the intrusion detection system with SVM classifier has a high classification accuracy even with the lack of a priori knowledge. In 2002, Mukkamala et al. used the SVM technique to realize intrusion detection and compared it with Neural Network. The result showed than the SVM technique achieved good experiment effects [9]. In the same year, Eskin et al. also proposed an intrusion detection method based on the first-class SVM [10]. Although the intrusion detection efficiency can be improved by SVM, the contradiction between the training speed and samples size was not resolved completely, especially in the training problem with large-scale data set. To solve this problem, Schölkopf et al. proposed nu-SVM algorithm [11,12]. Bottou used Gradient Descent method to solve the L1-SVM [13]. Lin et al. gave a L2-SVM algorithm based on Trust Region Newton method [14]. Tsang et al. proposed Core Vector Machine (CVM) [15]. In these methods, CVM is the most outstanding one, which can complete the training of classification model for 494,021 intrusion detection samples within a few seconds. However, the detection capability of CVM is not very satisfactory and needs to be further improved.

Based on the above analysis, a hybrid anomaly intrusion detection algorithm is proposed based on Partial Least Square (PLS) and CVM algorithms, namely, using PLS algorithm to extract the features

* Corresponding author. Address: Equipment Management and Safety Engineering College, Air Force Engineering University, Shaanxi, Xi'an 710038, China. Tel.: +86 15934896556.
E-mail address: Ganxsheng123@163.com (X.-s. Gan).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2012.09.004

from the sample data and then adopting CVM algorithm to build the anomaly intrusion model for the extracted features.

The paper is organized as follows. The PLS feature extraction technology is reviewed briefly in Section 2. Section 3 describes CVM algorithm. The anomaly intrusion detection model based on PLS and CVM algorithms is given in Section 4. The experiments are presented in Section 5. The last section gives some conclusive remarks.

2. PLS feature extraction technology

Partial Least Square (PLS), a second-generation multivariable statistical analysis technique, is a progressive integration of multiple linear regression, canonical correlation analysis and principal component analysis. At present, it has been being used widely in a variety of fields, such as chemometrics and industry design. In feature extraction, the principal elements obtained by PLS cannot only summarize the information given by independent variables excellently but also can explain dependent variables well [16,17].

Denote independent variables and dependent variables by $x_1, x_2, \cdots, x_p$ and $y_1, y_2, \cdots, y_q$, respectively, where p is the number of independent variables and q the number of dependent variables. To study the relation between independent variables and dependent variables, we construct the data block $X = [x_1, x_2, \cdots, x_p]_{n\times p}$ and $Y = [y_1, y_2, \cdots, y_q]_{n\times q}$. When PLS extracts the principal elements t_1 and u_1 from X and Y, respectively, (t_1 is a linear combination of $x_1, x_2, \cdots, x_p$ and u_1 is a linear combination of $y_1, y_2, \cdots, y_q$), it must satisfy.

(1) t_1 and u_1 should, respectively, carry the mutation information of X and Y as much as possible, namely, var $(t_1) \to \max$, var $(u_1) \to \max$.
(2) Correlation between t_1 and u_1 reaches to the maximum, namely, $r(t_1, u_1) \to \max$.

The two conditions above can also be summarized as

$$\max \quad \mathrm{cov}(t_1, u_1) = \sqrt{\mathrm{var}(t_1)\mathrm{var}(u_1)}\, r(t_1, u_1) \tag{1}$$

where $\mathrm{cov}(\cdot,\cdot)$ is the covariance operator, $var(\cdot)$ the variance operator and $r(\cdot)$ the correlation coefficient operator. After a proper transformation, Eq. (1) can be written as an optimization problem

$$\max_{w_1, c_1} \quad w_1^T E_0^T F_0 c_1$$

$$s.t. \begin{cases} w_1^T w_1 = 1 \\ c_1^T c_1 = 1 \end{cases} \tag{2}$$

where E_0 and F_0 are the standardized matrix of X and Y, respectively. w_1 and c_1 are the first main shaft of X and Y, respectively. In Eq. (2), t_1 can express X as well as possible and has a best explanation of Y.

After the first principal element is extracted, the regression model of E_0, F_0 against t_1 is established. Once the accuracy is achieved, the computation terminates. Otherwise, after E_0, F_0 are explained by t_1, the residual information is used to extract the second principal element t_2. The process is repeated until the accuracy is satisfied. In this paper, PLS is regarded as a feature extraction method. As long as the number of principal elements m ($m < A, A = rank(X)$, where $rank(\cdot)$ is the matrix rank operator) is reached, the computation stops. The pseudo codes of PLS algorithm are as follows. for $i = 1$ to m

$$w_i = E_0^T F_0 / \|E_0^T F_0\|$$
$$t_i = E_0 w_i$$
$$r_i = F_0^T t_i / \|t_i\|^2$$
$$p_i = E_0^T t_i / \|t_i\|^2$$
$$E_0 = E_0 - t_i s_i^T$$
$$F_0 = F_0 - t_i r_i^T$$

end

Through the above steps, we can extract m principal elements $T = [t_1, t_2, \cdots, t_m]$ with $W = [w_1, w_2, \cdots, w_m]$ and $S = [s_1, s_2, \cdots, s_m]$. Thus, for the standardized matrix E_{0t} of the test sample X_t, the projection matrix of principal elements is given as

$$T_{test} = E_{0t} W (S^T W)^{-1} T \tag{3}$$

3. Core vector machine algorithm

CVM, which was proposed in 2005, is a fast classification algorithm oriented to large-scale sample data. It first converts QP problem of SVM into Minimum Enclosing Ball (MEB) problem, and then uses an iterative $(1+\varepsilon)$-approximation algorithm to solve the MEB problem [15,18]. It has obvious advantages in pattern recognition problems with large-scale sample data and complex nonlinearity.

Suppose a set of points $x_1, x_2, \cdots, x_n$ is given, where $x_i \in R^d$. All these points can be mapped into a high-dimension feature space by a function φ and thus a set of mapped points $S_\varphi = \{\varphi(x_1), \cdots, \varphi(x_n)\}$ are produced. In the feature space, its MEB can be denoted by $B(c^*, R^*)$ with the center c^* and radius R^*. Then, the problem of searching the smallest ball that encloses all the mapped points is equivalent to

$$(c^*, R^*) = \arg\min_{R,c} R^2 : \|c - \varphi(x_i)\| \leqslant R^2, \quad \forall i = 1, \cdots, n \tag{4}$$

By Lagrange method, the corresponding dual matrix form can be given as

$$\max_{\alpha} \quad \alpha^T diag(K) - \alpha^T K \alpha : \alpha \geqslant \mathbf{0}, \quad \alpha^T e = 1 \tag{5}$$

where $\alpha = [\alpha_1, \cdots, \alpha_n]^T$ are Lagrange multiplier vector. $K_{m\times m} = [K(x_i, x_j)] = [\langle \varphi(x_i), \varphi(x_j)\rangle]$ is kernel matrix and $\mathbf{0} = [0, \cdots, 0]^T$, $e = [1, \cdots, 1]^T$. If Eq. (5) satisfies the condition

$$K(x, x) = \eta \tag{6}$$

where η is a constant, $\alpha^T diag(K)$ is equal to η. Then, Eq. (5) can be simplified as

$$\max_{\alpha} \quad -\alpha^T K \alpha : \ \alpha \geqslant \mathbf{0}, \quad \alpha^T e = 1 \tag{7}$$

Whenever the kernel K satisfies Eq. (6), any QP of the form like Eq. (5) can be regarded as a MEB problem.

The primal variables of MEB problem can be recovered from optimal α as

$$c = \sum_{i=1}^{n} \alpha_i \varphi(x_i), R = \sqrt{\alpha^T diag(K) - \alpha^T K \alpha} \tag{8}$$

Suppose a training set is $\{z_i \sim (x_i, y_i)\}_{i=1}^n$ with $y_i \in \{-1, 1\}$. The primal of the second-class L2-SVM is

$$\min_{\mathrm{w}, b, \rho, \xi_i} \quad \|\mathrm{w}\|^2 + b^2 - 2\rho + C\sum_{i=1}^{n} \xi_i^2$$
$$s.t. \quad y_i(\mathrm{w}^T \varphi(x_i) + b) \geqslant \rho - \xi_i, \forall i = 1, \cdots, n \tag{9}$$

According to the form of Eq. (7), the corresponding dual is given as

$$\max_{\alpha} \quad -\alpha^T \widetilde{K} \alpha : \alpha \geqslant \mathbf{0}, \alpha^T e = 1 \tag{10}$$

where $\widetilde{K} = [\widetilde{K}(z_i, z_j)]$ with

$$\widetilde{K}(z_i, z_j) = y_i y_j K(x_i, x_j) + y_i y_j + \frac{\delta_{ij}}{C} \tag{11}$$

Since $\widetilde{K}(z, z)$ is equal to a constant $\eta + 1 + \frac{1}{C}$ that satisfies Eq. (6), Eq. (10) can be viewed as a MEB problem

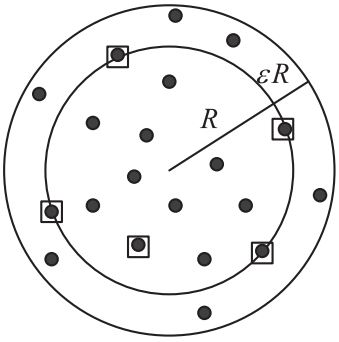

Fig. 1. Minimum Enclosing Ball (MEB).

$$\tilde{\varphi}(z_i) = \begin{bmatrix} y_i\varphi(x_i) \\ y_i \\ \tau_i/\sqrt{C} \end{bmatrix} \tag{12}$$

where τ_i is an m-dimensional vector, all of whose elements are 0 except that the ith element is 1.

After the QP problem is transformed into an MEB problem, CVM adopts an iterative $(1+\varepsilon)$-approximation algorithm to obtain a nearly-optimal solution. We denote the core set, the ball's center and radius at the tth iteration by S_t, c_t and R_t, respectively. The center and radius of the ball B are denoted by c_B and r_B, respectively. For a given $\varepsilon > 0$, the steps of CVM algorithm is as follows.

1. Initialize S_0, c_0 and R_0.
2. For all training point z, if $\tilde{\varphi}(z)$ does not fall outside the $(1+\varepsilon)$ ball B $(c_t, (1+\varepsilon)R_t)$, the computation terminates. Otherwise, find z farthest away from c_t, and set $S_{t+1} = S_t \bigcup \{z\}$
3. Find new MEB (S_t) and use Eq. (8) to set $c_{t+1} = c_{\mathrm{MEB}(S_{t+1})}$ and $R_{t+1} = r_{\mathrm{MEB}(S_{t+1})}$.
4. $t = t + 1$, and go back to Step 2.

Through the above steps, all the points added to the core set will be called core vectors. The MEB obtained is shown in Fig. 1. The set of squares is a Core Set (CS). The inner circle is the MEB of the square set. The outer circle, which covers all the points, is $(1+\varepsilon)$ expansion of the inner circle. Despite its simplicity, CVM greatly reduces the time and space complexities compared with the SVM algorithm.

4. PLS–CVM model for anomaly intrusion detection

PLS method requires to bi-linearly decompose the input and output data synchronously. The principal component vector obtained from input data is most relevant to that from the output data. Therefore, the features of the sample data extracted by PLS method contains information of both the input and output data. This can better reflect the inherent relationship between the input and output data compared with PCA method.

The principal element matrix T obtained by feature extraction, instead of the original input data, is used to establish the CVM model. This can significantly reduce the dimensions of input variables and can eliminate the noise introduced by the collinearity between high-dimension variables. The establishment of PLS–CVM anomaly intrusion detection model is just based on by this very idea. In addition, because the value ranges for different sample attributes are different, the standardization method need to be adopted in order to normalize the input and output sample data before modeling. Let $a_{\min}$, $a_{\max}$ be the minimum value and maximum value of the sample attribute A, respectively. The value a of the attribute A can be mapped into the standardization interval (L, H), then we get

$$a' = L + (H - L) \cdot \frac{a - a_{\min}}{a_{\max} - a_{\min}} \tag{13}$$

The standardization interval $(-0.5,0.5)$ is set to be uniform in the paper.

The principle of PLS–CVM modeling for anomaly intrusion detection is shown in Fig. 2. It can be divided into three stages as follows.

1. Extracting the principal element features
   (1) Standardize the training data block X, Y as E_0, F_0, respectively.
   (2) Extract the features of E_0, F_0 by PLS algorithm to obtain T, W and S.
   (3) Use Eq. (3) to calculate the principal element projection matrix T_{test} of the standardized test sample E_{t0}.
2. Training the anomaly intrusion detection CVM model Select parameters to establish the anomaly intrusion detection CVM model, where T is the input and Y is the single output.
3. Testing the anomaly intrusion detection CVM model
   As to the predicted output $\widehat{F}_{0t}$ of CVM model with the input T_{test}, execute the reverse process of the standardization in order to recover actual test output $\widehat{Y}_t$, and then make a decision with response to the detected intrusions.

5. Experimental simulation

All experiments are performed on a Pentium IV 2.4 GHz CPU PC with a 2 GB RAM. We used Gaussian kernel $\exp(-\|x-z\|^2/\beta)$, where β is the average squared distance between training patterns.

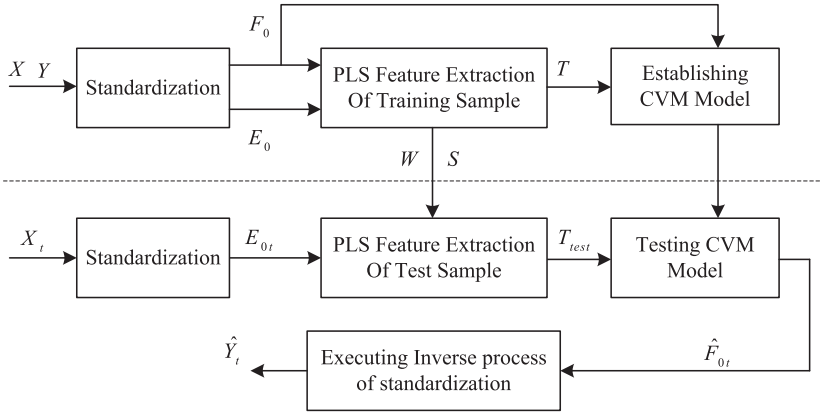

Fig. 2. Principle of anomaly intrusion detection PLS–CVM modeling.

Table 1
Comparison of simulation result under different training set size.

Training set size	Algorithm	Number of SV	CPU time (s)	DA (%)	DR (%)
5000	nu-SVC	1003	4.25	97.90	97.11
	PLS-nu-SVC	1001	2.22	98.16	97.69
	Ll-SVM	66	6.38	99.69	99.49
	PLS-L1-SVM	53	6.59	99.69	99.62
	L2-SVM	235	8.94	99.68	99.48
	PLS-L2-SVM	216	10.55	99.70	99.61
	CVM	22	0.42	97.49	94.76
	PLS–CVM	17	0.83	99.67	99.61
10000	nu-SVC	2002	27.61	97.69	96.62
	PLS-nu-SVC	2001	11.28	98.06	97.08
	Ll-SVM	51	14.05	99.69	99.49
	PLS-L1-SVM	58	3.41	99.83	99.78
	L2-SVM	80	18.28	99.75	99.63
	PLS-L2-SVM	122	3.70	99.82	99.75
	CVM	20	1.22	98.64	98.45
	PLS–CVM	16	1.72	99.52	99.67
20000	nu-SVC	4002	86.41	98.12	97.16
	PLS-nu-SVC	4001	51.33	98.54	98.09
	Ll-SVM	137	34.58	99.85	99.81
	PLS-L1-SVM	140	28.06	99.76	99.57
	L2-SVM	1085	627.69	99.86	99.58
	PLS-L2-SVM	503	163.11	99.77	99.79
	CVM	20	0.42	98.75	98.04
	PLS–CVM	16	0.56	99.83	99.79
40000	nu-SVC	8001	343.47	98.07	97.23
	PLS-nu-SVC	8001	268.75	98.20	97.45
	Ll-SVM	185	296.09	99.91	99.77
	PLS-L1-SVM	246	267.94	99.83	99.73
	L2-SVM	1661	5452.23	99.92	99.82
	PLS-L2-SVM	1246	5027.20	99.85	99.75
	CVM	27	1.19	80.04	50.32
	PLS–CVM	21	2.72	99.87	99.74

In nu-SVC, L1-SVM and L2-SVM algorithms, the LIBSVM software (SMO algorithm) well-known in the field of machine learning [19] was adopted. The CVM algorithm used LIBCVM software [15]. All the simulations were implemented in C++ language. In addition, no heuristic algorithm was used in the experiments.

5.1. Experimental data description

To verify the superiority of PLS–CVM algorithm in anomaly intrusion detection, the KDD99 data set, the most widely used data set in the evaluation of anomaly detection, was selected. This data set was prepared by Lee and Stolfo et al. It was built based on the data produced from the 1998 DARPA Intrusion Detection Evaluation program. The KDD data set consists of approximately 4,900,000 single connection vectors, each of which contains 41 features (34 continuous features and 7 discrete features). Since the data amount of KDD99 data set is too large, we chose the sample data randomly from Kddcup.data_10_percent.gz as the experiment data. Kddcup.data_10_percent.gz is about 10% of the KDD99 data set and contains 494,021 records, of which 97,278 are normal (accounting for 19.6%) and 396,743 are intrusion (accounting for 80.4%).

5.2. Experiment scheme design

Before the experiment, we firstly marked the normal record as 1 and the intrusion record as 0 in the sample data set, and then the attribute values of discrete and symbolic types were processed digitally. For example, as for the protocol-type attributes, Tcp is denoted as 1, Icmp as 2 and Udp as 3. In addition, considering the impact of attributes with value range larger than other attributes as well as the computation speed, we normalized the sample data within [0,1] in advance.

Four training sets, which contain 5,000, 10,000, 20,000 and 40,000 random samples selected from Kddcup.data_10_percent.gz, were constructed as well as their corresponding four test sets with 10000 random samples. For each training sets, L1-SVM, L2-SVM and CVM algorithms were used to establish the anomaly intrusion detection model based on all the 41 attributes firstly, and then 15 principal elements were exacted from the training sets by PLS algorithm to set up the anomaly intrusion detection CVM model. Finally, the models were verified by the test sets. It should be pointed out that the anomaly intrusion detection model of PLS–CVM was compared with that of L1-SVM, L2-SVM and CVM in terms of CPU time, Support Vector (SV), Detection Accuracy (DA) and Detection Rate (DR). DA is the ratio between the number of correctly-classified samples and the total number of samples. DR is the ratio between the number of detected anomaly samples and the total number of anomaly samples. In addition, for the four training sets with 10,000, 20,000, 50,000, and 60,000 random samples, we established the anomaly intrusion detection CVM model based on different principal elements extracted by PLS algorithm, in order to analyze the impacts of the number of principal elements on the performance of PLS–CVM model.

5.3. Experimental result analysis

The capability comparison of anomaly intrusion detection models based on nu-SVC, PLS-nu-SVC, L1-SVM, PLS-L1-SVM, L2-SVM, PLS-L2-SVM, CVM and PLS–CVM is given in Table 1 under different training set size. Fig. 3 shows ROC curves of different detection models built by 20,000 random training samples. The impacts of

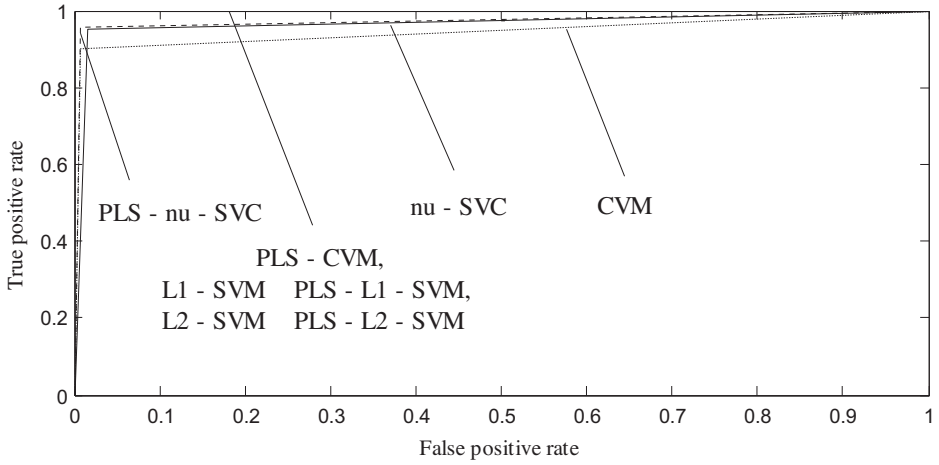

Fig. 3. ROC curves obtained by different intrusion detection models.

the principal elements t on PLS–CVM model performances are shown in Fig. 4 including: (a) t-SV, (b) t-CS, (c) t-DA and (d) t-DR.

The results in Table 1 indicate that, for the modeling problem of anomaly intrusion detection, CVM algorithm has no advantage and even gets an unsatisfactory result in DA and DR compared with nu-SVC, L1-SVM, L2-SVM and their PLS combined algorithms. This is further verified in Fig. 3, where the area under ROC curve of CVM is the smallest, namely, the detection efficiency is the smallest. However, CVM algorithm takes CPU time several orders of magnitude less than nu-SVC, L1-SVM, L2-SVM and their PLS combined algorithms in training, and needs a much less number of support vectors. Under the same conditions, the anomaly intrusion detection model built by PLS–CVM algorithm not only inherits the advantages of CVM algorithm in training speed and in support vector for large-scale data set, but also approximately equals L1-SVM, PLS-L1-SVM, L2-SVM and PLS-L2-SVM algorithms in DA and DR. This can also be validated by ROC curve in Fig. 3. Fig. 3 shows that the anomaly intrusion detection CVM model based on PLS feature extraction has a better detection capability than CVM model without PLS feature extraction. It can be seen from Fig. 4 that, for

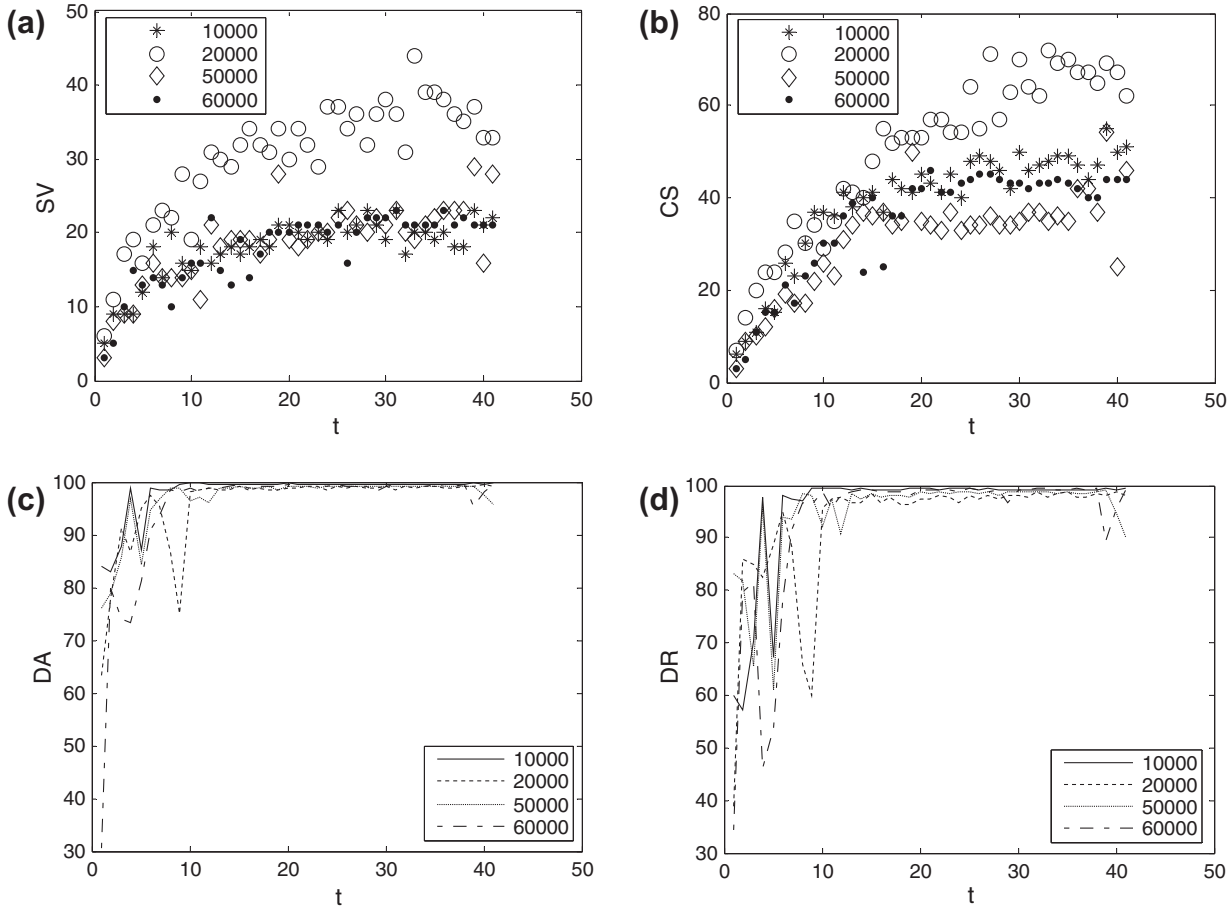

Fig. 4. Impact of different number of principal components on anomaly intrusion detection PLS–CVM model.

differently-scaled training sets, as the number of principal elements is in the range of 15–38, the anomaly intrusion detection PLS–CVM model can be on a relatively high level in DA and DR. Meanwhile, owing to information insufficiency or noise elements, the detection results of PLS–CVM model (in which the number of principal elements is chosen in the range of $1 \sim 14$ or $38 \sim 41$) are somewhat affected. In addition, PLS–CVM models with different numbers of principal elements all need less numbers of support vectors and core vectors. This is a great advantage in processing large-scale anomaly intrusion detection data. It should be mentioned that, owing to the introduction of PLS features extraction, the modeling speeds of nu-SVC, L1-SVM and L2-SVM are increased to some extent with slight improvement on the detection efficiency. However, in terms of speed and support vector, there is still a great deal of differences compared with PLS–CVM.

6. Conclusion

To solve the problem of anomaly intrusion detection, a combined intrusion detection algorithm was proposed based on PLS algorithm and CVM algorithm. PLS algorithm has functions such as dimension reduction, de-noising and multi-correlation elimination between independent variables. CVM algorithm has the advantage of fast-processing large-scale sample data. By integrating PLS algorithm with CVM algorithm in a systematic way so as to complement each other, the problem of feature extraction and fast modeling for large-scale sample data in anomaly intrusion detection can be effectively solved. Simulation results indicate that, as the number of principal elements is selected in the range of $15 \sim 38$, PLS–CVM algorithm basically retains the advantages of CVM modeling such as speediness and simplicity, and has a significant improvement on the detection effect compared with CVM algorithm. Thus, PLS–CVM algorithm provides an effective way of solving the problem of anomaly intrusion detection.

References

[1] R. Bace, Intrusion Detection, Macmillan Technical Publishing, New York, 2000.
[2] T. Verwoerd, R. Hunt, Intrusion detection techniques and approaches, Computer Communications 25 (15) (2002) 1356–1365.
[3] C. Endorf, E. Schultz, J. Mellander, Intrusion Detection & Prevention, McGraw-Hill, New York, 2004.
[4] D. Dasgupta, F. Gonzalez, An immunity-based technique to characterize intrusions in computer networks, IEEE Transactions on Evolutionary Computation 6 (3) (2002) 281–291.
[5] H. Peng, Research of intrusion detection method based on rough set, Journal of UEST of China 35 (1) (2006) 108–110.
[6] J.M. Bonifacio Jr., A.M. Cansian, A.C.P.L.F. De Carvalho et al., Neural networks applied in intrusion detection systems, in: Proceedings of IEEE International Joint Conference on Neural Networks, vol. 1, 1998, pp. 205–210.
[7] H. Debar, M. Becker, D. Siboni, A neural network component for an intrusion detection system, in: Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, 1992, pp. 240–250.
[8] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[9] S. Mukkamala, G. Ljanoski, A.H. Sung, Intrusion detection using neural networks and support vector machines, in: Proceedings of IEEE International Joint Conference on Neural Networks, 2002, pp. 87–90.
[10] E. Eskin, A. Amokd, M. Prerau, et al., A geometric framework for unsupervised anomaly detection: detecting intrusion in unlabeled data, in: Applications of Data Mining in Computer Security, Kluwer, 2002, pp. 537–547.
[11] B. Schölkopf, A. Smola, R. Williamson, P.L. Bartlett, New support vector algorithms, Neural Computation 12 (2000) 1207–1245.
[12] R.E. Fan, P.H. Chen, C.J. Lin, Working set selection using second order information for training SVM, Journal of Machine Learning Research 6 (2005) 1889–1918.
[13] L. Bottou, Stochastic gradient descent examples, 2007. <http://leon.bottou.org/projects/sgd>.
[14] C.J. Lin, R.C. Weng, S.S. Keerthi, Trust region newton method for large-scale logistic regression, Journal of Machine Learning Research 9 (2008) 627–650.
[15] I.W. Tsang, J.T. Kwok, P.M. Cheung, Core vector machines: fast SVM training on very large data sets, Journal of Machine Learning Research 6 (2005) 363–392. Software available at <http://c2inet.sce.ntu.edu.sg/ivor/cvm.html>.
[16] M. Barker, W. Rayens, Partial least squares for discrimination, Journal of Chemometrics 17 (2003) 166–173.
[17] H. Wold, Partial least squares, in: Encyclopedia of Statistical Sciences, John Wiley & Sons, New York, 1985.
[18] I.W. Tsang, A. Kocsor, J.T. Kwok, Simpler core vector machines with enclosing balls, in: Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML), Corvallis, Oregon, USA, June 2007.
[19] C.J. Lin, C.C. Chang, LIBSVM: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.