# An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines *

Xin Xu[1,2] and Xuening Wang[2]

[1] School of Computer, National University of Defense Technology,
410073, Changsha, P.R.China
xuxin_mail@263.net
[2] Institute of Automation, National University of Defense Technology,
410073, Changsha, P.R.China

**Abstract.** Network intrusion detection is an important technique in computer security. However, the performance of existing intrusion detection systems (IDSs) is unsatisfactory since new attacks are constantly developed and the speed of network traffic volumes increases fast. To improve the performance of IDSs both in accuracy and speed, this paper proposes a novel adaptive intrusion detection method based on principal component analysis (PCA) and support vector machines (SVMs). By making use of PCA, the dimension of network data patterns is reduced significantly. The multi-class SVMs are employed to construct classification models based on training data processed by PCA. Due to the generalization ability of SVMs, the proposed method has good classification performance without tedious parameter tuning. Dimension reduction using PCA may improve accuracy further. The method is also superior to SVMs without PCA in fast training and detection speed. Experimental results on KDD-Cup99 intrusion detection data illustrate the effectiveness of the proposed method.

## 1 Introduction

With the wide spread of computer networks, security problems in computer systems have become more and more important since various intrusions or viruses may cause significant losses to information assets. To defend attacks of information systems, lots of security techniques and products, such as firewalls, intrusion detection systems (IDSs), etc., have been developed. Among these security techniques, intrusion detection plays a key role because it is a dynamic defense technique, which is different from earlier static defense techniques including firewalls and access control.

In intrusion detection systems, misuse detection and anomaly detection are the two main classes of detection policies. Misuse detection can detect known attacks by constructing a set of features or signatures of attacks while anomaly detection detects novel attacks by modeling normal behaviors. Both misuse and anomaly detections rely on analysis of large amounts of audit data or events. It is a time-consuming and tedious

* Supported by the National Natural Science Foundation of China Under Grants 60303012, 90104001, Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20049998027, and Chinese Post-Doctor Science Foundation under Grant 200403500202.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 696 – 703, 2005.

work. Thus, intrusion detection techniques based on data mining or machine learning [1][2] have attracted much attention in recent years, which are usually called adaptive intrusion detection techniques.

In data-mining-based IDSs, the process of data analysis and behavior modeling can be automatically carried out and there are also two different processing policies, i.e., misuse detection and anomaly detection. In misuse detection, various standard data mining algorithms [2,3], fuzzy logic models [4], and neural networks [5] have been used to classify network intrusions. In anomaly detection, data mining methods based on statistics [6], or clustering techniques [7] are employed to identify attacks as deviation from normal usage.

Despite many advances that have been achieved, existing IDSs still have some difficulties in improving their performance to meet the requirements of detecting increasing attacks in high-speed networks. One difficulty is the problem of detection accuracy. Since misuse IDSs employ signatures of known attacks, it is hard for them to detect deformed attacks, notwithstanding completely new attacks. Although anomaly detection can detect new types of attacks by modeling a model of normal behaviors, the false alarm rates in anomaly-based IDSs are usually high. Another difficulty of IDSs is to detect intrusions in real-time with large amounts of data in high-speed networks.

To overcome the above problems, this paper proposes a novel adaptive network intrusion detection method based on principal component analysis (PCA) [8] and support vector machines (SVMs) [9]. In the proposed method, PCA is used to reduce the feature dimension of network connection records and SVMs are employed to construct intrusion detection model based on the processed training data. Compared to previous IDS methods, the adaptive intrusion detection method not only has good accuracy but also has advantages both in fast training and testing speed, which will be illustrated in the experiment on KDD-Cup99 dataset.

## 2 Intrusion Detection as a Multi-class Pattern Recognition Problem

Although the intrusion detection method studied in this paper can be applied to general-purpose IDSs, to facilitate discussion, we will only consider network connection data, especially the KDD-Cup99 dataset in the following.

The KDD-Cup99 dataset is based on the 1998 DARPA intrusion detection evaluation program, where an environment was setup to simulate a typical US Air Force LAN and raw tcpdump data were collected. For each TCP/IP connection, 41 quantitative and qualitative features were extracted as a data record. The data records are all labeled with one of the five types, which are

- Normal: Normal connections are generated by simulated daily user behavior such as visiting web pages, downloading files, etc.
- DoS: DoS denotes the denial of service attacks. A denial of service attack causes the computing power or memory of a victim machine too busy or too full to response to legitimate access. Examples of DoS attacks are Apache2, Back, Land, Mail bomb, SYN Flood, Ping of death, Process table, Smurf, Syslogd, Teardrop, Udpstorm.ï
- U2R: U2R means user to root, which is a class of attacks that a hacker begins with the access of a normal user account and then become a super-user by

exploiting vulnerabilities of the system. Examples are Eject, Ffbconfig, Fdformat, and Loadmodule.

- R2L: The R2L attack or remote to local attack is a class of attacks that a remote user gains access of a local account by network communication, which include Sendmail, Xlock, and Xsnoop.
- Probe: A Probe attack scans the network to gather information of computers so that vulnerabilities can be found for further attacks.

To construct an intrusion detection model based on the KDD-Cup99 dataset, each data record is denoted by a 41-dimensional vector with class labels from 1 to 5, where qualitative elements are transformed to discrete values and class labels correspond to the above five types of connections. Then, the construction of intrusion detection model becomes a multi-class pattern recognition problem so that data mining methods can be employed.

# 3 Adaptive IDS Using PCA and SVM

## 3.1 Framework of the Adaptive IDS

The framework of the adaptive IDS includes a training process and a testing process, as shown in Fig.1.

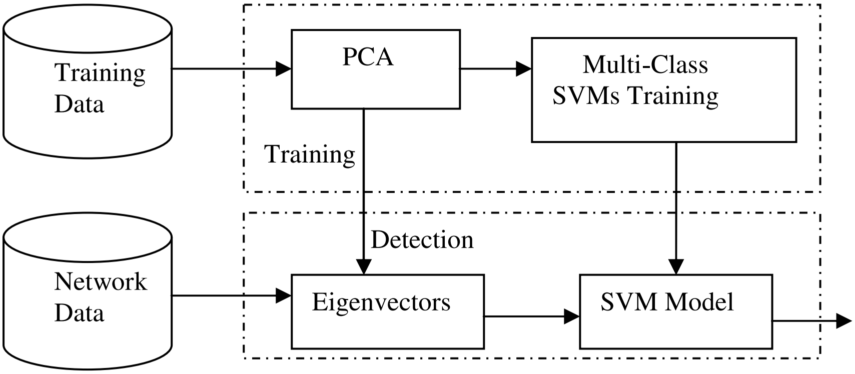

**Fig. 1.** Framework of the adaptive IDS

The training process consists of two main steps, i.e., the dimension reduction step and the classifier training step. In the dimension reduction step, the PCA algorithm is used to compute the principal components of the data.

## 3.2 Dimensionality Reduction Using PCA

In both neural network and statistics studies, PCA is one of the most fundamental tools of dimensionality reduction for extracting effective features from high-dimensional vectors of input data. In the following, we will discuss the application of PCA to dimension reduction of network connection data and its combination with SVMs.

As discussed in Section 2, the network data records can be denoted as

$$x_t = [x_{t1}, x_{t2}, ..., x_{tn}]^T \qquad (t\text{=}1,2,\ldots, N),\ n\text{=}41 \tag{1}$$

Let

$$\mu = \frac{1}{N}\sum_{t=1}^{N} x_t \tag{2}$$

Then, the covariance matrix of data vectors is

$$C = \frac{1}{N}\sum_{t=1}^{N}(x_t - \mu)(x_t - \mu)^T \tag{3}$$

The principal components are computed by solving the eigenvalue problem of covariance matrix $C$:

$$Cv_i = \lambda_i v_i \tag{4}$$

where $\lambda_i$ $(i = 1,2,...,n)$ are the eigenvalues and $v_i (i = 1,2,...,n)$ are the corresponding eigenvectors.

To represent network data records with low dimensional vectors, we only need to compute the first $m$ eigenvectors which correspond to the $m$ largest eigenvalues.

Let

$$\Phi = [v_1, v_2, ..., v_m], \quad \Lambda = \mathrm{diag}[\lambda_1, \lambda_2, ..., \lambda_m] \tag{5}$$

Then we have

$$C\Phi = \Phi\Lambda \tag{6}$$

In PCA, a parameter $\nu$ can be introduced to denote the approximation precision of the $m$ largest eigenvectors so that the following relation holds.

$$\sum_{i=1}^{m}\lambda_i / \sum_{i=1}^{n}\lambda_i \geq \nu \tag{7}$$

Given a precision parameter $\nu$, we can select the number of eigenvectors based on (6) and (7), and the low-dimensional feature vector of a new input data $x$ is determined as follows

$$x_f = \Phi^T x \tag{8}$$

### 3.3 Hybrid Intrusion Detection Using Multi-class SVMs

Support vector machines (SVMs) are relatively new statistical learning algorithms that provide powerful tools for learning classification or regression models with good generalization ability in sparse, high-dimensional settings. The success of SVMs is due to the statistical learning theory studied by Vapnik, which gives key insights into the structural risk minimization (SRM) principle for improving generalization ability of learning machines [13]. SVM learning can be viewed as an efficient realization of Vapnik's SRM principle and lots of work has been done on revised SVM algorithms with applications in many supervised learning or pattern recognition problems [9].

As discussed previously, adaptive intrusion detection can be viewed as a multi-class pattern classification problem. In the following, we will present the multi-class SVM algorithm for adaptive network intrusion detection.

Suppose the training samples of network audit data are given as

$$\{(\vec{x}_i, y_i)\}, \quad i = 1,2,\ldots,N \qquad y_i \in \{1,2,\ldots,m\} \tag{9}$$

where $N$ is the total number of training samples and $m$=5 is the number of class labels, which are normal, U2R, DOS, R2L and Probe.

Since SVMs were originally proposed for two-class problems and research work on SVMs mainly focused on binary classification problems, the multi-class pattern recognition problem of intrusion detection can be tackled by decomposing the 5-class problem to several binary problems. In the decomposition, we use the one-against-one strategy commonly applied in the literature. Based on the one-against-one strategy, the construction of multi-class SVM classifiers can be implemented by training 10 two-class SVM classifiers with different training samples.

In two-class SVM learning, a hyperplane is considered to separate two classes of samples. Following is the linear form of a separating hyperplane.

$$(\vec{w} \cdot \vec{x}) + b = 0 \qquad \vec{w} \in R^n, \; b \in R \tag{10}$$

Based on the SRM principle in statistical learning theory, the optimal separating hyperplane can be constructed by solving the following optimization problem

$$\min_{\vec{w},b} \frac{1}{2}\|\vec{w}\|^2 \tag{11}$$

subject to

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad i = 1,2,\ldots,N \tag{12}$$

In support vector learning, the optimization problem for constructing optimal hyperplane is solved by its Lagrangian dual using Karush-Kuhn-Tucker (KKT) conditions. Furthermore, to reduce the effects of noise and outliers in real data, the following soft margin techniques are usually used, which is to solve the primal optimization problem as

$$\min_{\vec{w},b} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N} \xi_i \tag{13}$$

subject to

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \; i = 1,2,\ldots,N \tag{14}$$

An important element for the success of SVMs is the 'kernel trick', which is to transform the above linear form of support vector learning algorithms to nonlinear ones without explicitly computing the inner products in high-dimensional feature spaces. In the kernel trick, a Mercer kernel function is employed to express the dot products in high-dimensional feature space

$$k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j) \tag{15}$$

Then the dual optimization problem of SVMs for two-class soft margin classifiers is formulated as follows

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(\vec{x}_i \cdot \vec{x}_j) \tag{16}$$

subject to

$$0 \le \alpha_i \le C,\ i = 1,2,...,N \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{17}$$

To solve the above quadratic optimization problem, various decomposition-based fast algorithms have been proposed in the literature, such as SMO [10], etc. For details on the algorithmic implementation of SVMs, please refer to [11].

In our multi-class SVM classifier, the decision function of each binary SVM is

$$f_k(\vec{x}) = \mathrm{sgn}(\sum_{i=1}^{N} \alpha_{ki} y_{ki} k(\vec{x}_{ki}, \vec{x}) + b_k) \quad k = 1,2,...,m \tag{18}$$

where $f_k(\vec{x})$ is the decision function of classifier $k$ and ($\vec{x}_{ki}, y_{ki}$) ($k$=1,2,…,$m$) are the corresponding training samples.

In the multi-class SVM classification for adaptive intrusion detection, a voting strategy is used: each binary classification is considered to be a voting where votes can be cast for all data points and an input is designated to be in a class with maximum number of votes.

## 4 Experiments on KDD-Cup99 Data

The proposed method was applied in the KDD-Cup99 intrusion detection dataset to demonstrate its effectiveness in automatic model construction and processing speed enhancement. In the experiments, a subset of KDD-Cup99 data was selected and partitioned to a training data set with 9321 records and a test set with 15705 records. We tested the proposed multi-class SVMs with PCA for dimension reduction, as well as multi-class SVMs using original data dimension. The network data records are normalized to interval [0, 1]. The performance of the classifiers includes training and testing accuracy and the processing speed during training and testing.

The experimental results are shown in Table 1 and 2. In all the experiments, RBF kernel functions are used and the width parameter is chosen as $\sigma$ =0.1, which was optimized manually. In the implementation of binary SVMs, the LibSVM [12] package was used. Table 1 shows the training and testing accuracy of multi-class SVM using PCA for dimension reduction as well as SVMs without PCA. Note that the accuracy of the proposed SVM+PCA method is fairly good except that the results of class 'R2L' are not very satisfactory. The reason may be the amount of U2R data is very small in KDD-Cup99 dataset so that it will cause some information loss when dimension reduction is performed using PCA. However, this problem may be solved by collecting more training data of U2R attacks.

**Table 1.** Comparisons of training and testing accuracy

| Class | SVMs with PCA | | SVMs without PCA | |
|---|---|---|---|---|
| | Training accuracy | Test accuracy | Training accuracy | Test accuracy |
| Normal | 99% | 83.9% | 100% | 74.3% |
| Dos | 99.3% | 99.9% | 100% | 100% |
| Probe | 94.7% | 94.1% | 99.1% | 98.9% |
| U2R | 97.8% | 97.8% | 100% | 100% |
| R2L | 60% | 58.3% | 100% | 100% |

For SVMs without PCA, although the training accuracies are slightly better than SVMs using PCA, the test accuracy of class Normal is worse than that of multi-class SVMs using PCA. This demonstrates that dimension reduction using PCA may improve the generalization ability of classifiers. For training data of R2L, the training and testing accuracies of multi-class SVM using full dimension data are very good. As discussed above, the amount of R2L data is very small (only about 20 records) so that classifiers using higher dimension data usually have better performance in accuracy. However, when data amount increases, PCA can be used to reduce data dimension without sacrificing much performance in accuracy and the generalization ability of classifiers using PCA may be improved. Furthermore, as illustrated in the following Table 2, SVMs with PCA will benefit from improved training and testing speed, which is important for high-speed network applications.

Table 2 shows the comparisons of training and testing speed of SVMs with and without PCA. It is clear that the proposed PCA+SVMs classifier is 5 times faster in training and 2 times faster in testing than conventional SVMs without PCA.

**Table 2.** Speed comparison of PCA+SVMs and conventional SVMs

| Classifiers | Training time (s) | Testing time (s) |
|---|---|---|
| SVM (Original 41-dimension feature) | 151.9 | 30.4 |
| SVM (Using PCA for feature extraction) | 33.3 | 14.4 |

## 5 Conclusion and Future Work

This paper proposes an adaptive network intrusion detection method using SVMs combined with PCA. The aim of the approach is to not only realize automatic construction of intrusion detection models with high accuracy but also make network IDS models to be processed as fast as possible so that the applications in high-speed networks can be feasible. Experimental results on KDD-Cup99 intrusion detection dataset show that the proposed method has comparable accuracy as that of conventional SVMs without PCA and it is much faster in processing speed than conventional SVMs. Future work may include applying and testing the proposed method in real-time intrusion detection for real network data.

## References

1. Lippmann R., Cunningham R.: Improving Intrusion Detection Performance Using Keyword Selection and Neural Networks. Computer Networks, 34(4), (2000) 597--603
2. Lee W., Stolfo S. J.: Data Mining Approaches for Intrusion Detection. Proceedings of the 1998 USENIX Security Symposium, (1998)
3. Lee, W., Stolfo, S., and Mok, K.: Adaptive Intrusion Detection: A Data Mining Approach. Artificial Intelligence Review, 14(6), (2000) 533 – 567
4. Luo J., Bridges S. M.: Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection. International Journal of Intelligent Systems, (2000) 687-703
5. Cannady, J.: Applying Neural Networks to Misuse Detection. In: Proceedings of the 21st National Information Systems Security Conference. (1998)
6. Mahoney M., Chan P.: Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks. In: Proceedings of 8th International Conference on Knowledge Discovery and Data Mining, (2002) 376-385
7. Shah H., Undercoffer J. and Joshi A.: Fuzzy Clustering for Intrusion Detection. In: Proceedings of the 12th IEEE International Conference on Fuzzy Systems. (2003) 1274-1278
8. Jolliffe I. T.: Principal Component Analysis. Springer. 2nd edition. (2002)
9. Hastie, T. J., Tibshirani, R. J. and Friedman, J. H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2001
10. Platt J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: B.Scholkopf, C.J.C. Burges, and A.J.Smola, editors, Advances in Kernel Methods—Support Vector Learning, Cambridge, MIT Press. (1999) 185-208
11. Lin C.-J.: Formulations of Support Vector Machines: a Note from an Optimization Point of View . Neural Computation, 13(2), (2001) 307-317
12. Fan R.-E., Chen P.-H., and Lin C.-J.: Working Set Selection using the Second Order Information for Training SVM. Technical report, Department of Computer Science, National Taiwan University, (2005)
13. Vapnik, V. N. Statistical Learning Theory. Wiley. (1998)