Application of Support Vector Machine and Genetic Algorithm to Network Intrusion Detection

Hua Zhou, Xiangru Meng, Li Zhang
The Telecommunication Engineering Institute
AFEU
Xi'an, China
zhoumiaomiao_2005@126.com

***Abstract*—Intrusion detection is actually a classification problem. It is very important to increase the classification accuracy. Support Vector Machine (SVM) is a powerful tool to solve classification problems. Many works have been done in intrusion detection based on SVM, and the detection accuracy is relatively high. But how to get a higher accuracy is a new question. In this paper, we apply SVM and Genetic Algorithm (GA) to intrusion detection to solve this problem. We first use GA for feature selection and optimization, and then use SVM model to detect intrusions. In order to verify our approach, we tested our proposal with KDD Cup99 dataset, and analyzed its performance. The experimental results show that the proposed approach is an efficient way in network intrusion detection.**

Keywords-Intrusion Detection;Support Vector Machine; Genetic Algorithm

I. INTRODUCTION

With the development of network and information technology, network is becoming increasingly important in politics, economy, military affairs and daily life. But we can't turn a blind eye to a fact that more and more network attacks have seriously threatened our networks. In order to protect the network and information, intrusion detection is applied to network security issues. Intrusion detection is a kind of security technology to protect the network against the intrusion attacks. It includes two main categories. One is misuse detection and the other is anomaly detection. Misuse detection can exactly detect the known attacks, but it can do nothing against the unknown attacks. However, anomaly detection can detect the unknown and new attacks. So it is an all-round method to detect attacks.

Nowadays, there are many methods that have been applied to intrusion detection such as wavelet analysis [1], fuzzy data mining [2] and intelligent Bayesian classifier [3]. But many experiments [4], [5], [6] show that there is a high detection accuracy when Support Vector Machine (SVM) is used in intrusion detection. But it is very critical to select features in intrusion detection based on SVM. Because some features in data may be irrelevant or redundant. Furthermore, they may have a negative effect on the accuracy of the classifier. In order to improve intrusion detection performance, we must select appropriate features and optimize them during data-preprocessing step at first. In this paper, we mainly study intrusion detection based on SVM, and use Genetic Algorithm (GA) to select and optimize features at the same time.

The rest of this paper is organized as follows. In section 2, SVM theory and intrusion detection based on SVM are described. In section 3, feature selection and optimization with GA is presented. In section 4, the experimental results using KDD Cup99 are presented. Finally, some concluding remarks are given in section 5.

II. SVM AND INTRUSION DETECTION BASED ON SVM

A. SVM

Support Vector Machine (SVM) is based on the structural risk minimization principle from the statistical learning theory. Its kernel is to control the empirical risk and classification capacity in order to maximize the margin between the classes and minimize the true costs [6]. A support vector machine finds an optimal separating hyper-plane between members and non-members of a given class in a high dimension feature space [4]. Although the dimension of feature space is very large, it still shows good generalization performances. The basic SVM theory is as follows.

First, we are given a set of training examples $S=((X_1,y_1),.....(X_l,y_l))$, $l=1,2,.....n$, $X_l \in R^n$, and $y \in \{+1,-1\}$ where X is the input data and y is output. If y is "1", it means the input example is normal. If y is "-1", it means the input example is abnormal. Suppose this set can be separated by a hyper-plane $W \cdot X + b = 0$. That is, all the training examples satisfy:

$$y_i(\langle W \cdot X_i \rangle + b) \geq 1 \text{, for all } i = 1,......l \tag{1}$$

W is an adjustable weight vector, and b is the bias term.

In Fig. 1, the margin between two hyper-planes

$H_1 : W \cdot X_1 + b = 1$ and $H_2 : W \cdot X_1 + b = -1$ is $2/\|W\|$. And the hyper-plane that maximizes the margin is the optimal separating hyper-plane. Thus, the optimization is now a convex quadratic programming problem.

$$\underset{W,b}{Minimize} \qquad \Phi(W) = \frac{1}{2}\|W\|^2$$

$$subject\ to \qquad y_i(\langle W \cdot X_i \rangle + b) \geq 1 , \quad i = 1,......l . \tag{2}$$

1-4244-1312-5/07/$25.00 © 2007 IEEE

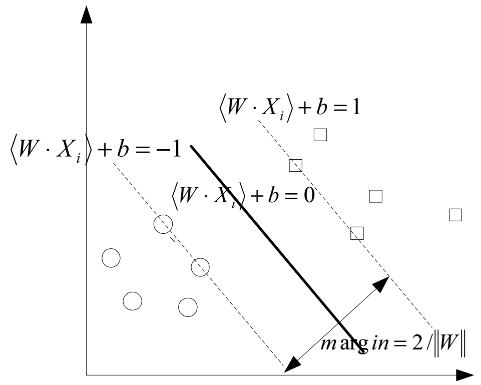
$\langle W \cdot X_i \rangle + b = 1$
$\langle W \cdot X_i \rangle + b = -1$
$\langle W \cdot X_i \rangle + b = 0$
$margin = 2/\|W\|$

Figure 1. Separating hyper-plane between two classes

Finally, we can get the optimal classification function:

$$f(x) = \operatorname{sgn}\{\sum_{i=1}^{l} \alpha_i y_i K(X_i, X) + b\} \tag{3}$$

Where α_i is Lagrange multiplier. When the set is non-linearly separable, $K(X_i, X)$ is kernel function, and it must satisfy the Mercer condition. When the set is linearly separable, $K(X_i, X)$ means inner product $\langle X_i \cdot X \rangle$.

B. *Intrusion Detection Based on SVM*

Intrusion detection is essentially a pattern recognition and classification problem. Its goal is to distinguish the normal data from the abnormal through detection. SVM is a new method for designing classifier based on small sample learning, and is especially applied to small sample data. Furthermore, it is not sensitive to the dimension of data. Therefore, it is absolutely feasible to apply SVM to intrusion detection [7].

The processes of intrusion detection based on SVM are depicted as follows:

Capturing network data. We set the Network Interface Card (NIC) on promiscuous mode, and use libpcap or tcpdump to collect the data stream of a given network.

Data preprocessing. The data applied to SVM have different types. Some are symbolic such as user's command, and some are numeric such as the number of connections. It is necessary to preprocess the data and transform them to the same format. This format must be recognized and dealt with by SVM.

SVM training and test. We first train the selected data with SVM. Then, we can get a set of support vectors after training and put them into support vector database (SVB). Finally, the support vectors in SVB are used to detect the actual network events and the decisions are made.

Responding to network events. If the intrusion detection system (IDS) has detected the attacks, it will adopt some measures to hold them back such as giving an alert, cutting off connections and so on.

III. FEATURE SELECTION AND OPTIMIZATION WITH GA

GA is an adaptive method of global-optimization searching and simulates the behavior of the evolution process in nature. It maps the searching space into a genetic space. That is, every possible key is encoded into a vector called a chromosome. One element of the vector represents a gene. All of the chromosomes make up of a population and are estimated according to the fitness function. A fitness value will be used to measure the "fitness" of a chromosome. Initial populations in the genetic process are randomly created. GA then uses three operators to produce a next generation from the current generation: reproduction, crossover, and mutation. GA eliminates the chromosomes of low fitness and keeps the ones of high fitness. This whole process is repeated, and more chromosomes of high fitness move to the next generation, until a good chromosome (individual) is found.

A. *Chromosome Encoding*

Encoding is the first step in GA. For a data record, we convert each value of its feature into a bit binary gene value, 0 or 1. In our experiments, we choose the subsets of KDD Cup99, 1999 kddcup.data_10_percent and corrected [8], as the training dataset and test dataset. Because the values of feature No.2, No.3, and No.4 are all symbols, it is not necessary to optimize these three features. It is just all right to eliminate them before encoding and add them to the record after optimization. For feature that has a numeric value, if its value isn't 0, we convert it to 1; otherwise we convert it to 0. For example, a record (0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0, 0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00, 0.11,0.00,0.00,0.00,0.00,0.00）can be converted to (0110000 0100000000001100001001110100000) after encoding.

B. *Fitness Function*

We adopt the value of fitness function to decide whether a chromosome is good or not in a population. Equations (4) are used to calculate the fitness value of every chromosome.

$$\begin{cases} F(X) = AX + \beta N_0 \\ A = (\alpha_1, \alpha_2, \cdots, \alpha_n) \\ X = (x_1, x_2, \cdots, x_n)^T \end{cases} \tag{4}$$

Where N_0 is the number of 0 in a chromosome, β is the coefficient, x_n means the nth gene (0 or 1), and α_n means the weight of the nth gene. We use (5) to calculate the weight.

$$\alpha_n = \frac{N_n}{N_{all}} \tag{5}$$

Where N_n means the number of the nth feature in dataset when its value isn't 0, and N_{all} means the total number of the nth feature in dataset.

IV. Experimental Results

A. Feature Selection and Optimization

In our experiments, the total number of population is 100, the length of a chromosome is 38, the number of generation is 300, the crossover rate is 0.8, and the mutation rate is 0.05. The optimal fitness value varies with β, so the features are different after optimization. The results of feature selection and optimization are described in Table Ⅰ.

TABLE I. GA FEATURE SELECTION AND OPTIMIZATION RESULTS

β	0.1	0.3	0.5	0.7	0.9
Amount of features	34	34	26	16	5
Feature No.	1,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20,21,23,24,25,26,27,28,29,30,31,33,34,35,36,37,38,39,41	1,5,6,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,25,26,27,28,29,30,31,33,34,35,36,37,38,39,41	1,5,6,8,9,10,12,13,16,18,25,26,27,28,29,30,31,32,34,35,36,37,38,39,40,41	5,7,8,9,17,25,26,27,28,29,31,34,35,37,38,39	25,27,31,33,38

B. SVM for Classification

We randomly select a set of data from training dataset and test dataset respectively. The training data includes 1934 records, and the test data includes 1944. We first use the training data to create a SVM model. Then, the model is applied to classify the test data. The results are depicted in TableⅡ.

TABLE II. SVM CLASSIFICATION ACCURACY

β	Without feature selection	0.1	0.3	0.5	0.7	0.9
Accuracy	95.22%	95.32%	96.30%	98.46%	98.02%	97.14%

TABLE III. SVM CLASSIFICATION ACCURACY WHEN β IS 0.5

Data	1st	2nd	3rd
Accuracy	98.97%	98.61%	98.92%

It shows that, SVM itself has a good classification performance, and can distinguish the normal data from the abnormal well. Its classification accuracy is as high as 95.22%. Furthermore, with the adoption of GA feature selection and optimization, the classification accuracy increases much. Considering the balance between the number of features and the accuracy (That is, the features selected can not only indicates the basic network state, but also improve the classification accuracy.), we set β to 0.5 finally. At this time, the number of features reduces from 41 to 29, and the accuracy increases to 98.46%. When $\beta = 0.5$, we randomly select another three sets of test data. The first includes 1950 records, the second includes 1944 and the third includes 1943. The results are depicted in TableIII.

V. Conclusions

In this paper, we proposed an intrusion detection method based on SVM and GA. Because some features in data may be irrelevant or redundant, we first apply GA to select and optimize features, and then apply SVM to classify. The experimental results show that SVM can achieve a good classification accuracy, and the accuracy can be improved obviously after feature selection and optimization. Therefore, it is efficient to apply SVM and GA to intrusion detection.

References

[1] Sanjay Rawat, Challa S. Sastry. Network Intrusion Detection Using Wavelet Analysis. In:G. Das and V.P.Gulati. Ed. CIT 2004, LNCS 3356, 2004, pp. 224-232.

[2] Jian Guan, Da-xin Liu, Tong Wang. Applications of Fuzzy Data Mining Methods for Intrusion Detection Systems. In: A. Lagana et al. Ed. ICOIN 2004, LNCS 3045, 2004, pp. 706-714.

[3] Andrea Bosin, Nicoletta Dessi, Barabara Pes. Intelligent Bayesian Classifiers in Network Intrusion Detection. In:M.Ali and F.Esposito. Ed. IEA/AIE 2005, LANI 3533, 2005, pp. 445-447.

[4] Dong Seong Kim, Jong Sou Park. Network-based intrusion detection with support vector machines. In: Kahng H-K. Ed. ICOIN 2003, LNCS 2662, 2003, pp. 747-756.

[5] RAO Xian, DONG Chun-xi, YANG Shao-quan. An intrusion detection system based on support vector machine. Journal of Software. 4 (2003) , pp. 798-803.

[6] ZHANG Kun, CAO Hong-xin, YAN Han. Application of support vector machines on network abnormal intrusion detection. Application Research of computers. 5 (2006) , pp. 98-100.

[7] LI Hui, GUAN Xiao-hong, ZAN Xin. Network intrusion detection based on support vector machine. JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT. 6 (2003) , pp. 800-807.

[8] KDD Cup99 Data. http://kdd.ics.uci.edu/ databases/kddcup99/kddcup 99.html.