http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (1): 185-189, 2008
ISSN 1812-5638

Minimax Probability Machine with Genetic Feature Optimized for Intrusion Detection

Zhen-Guo Chen and Shu Wang
Department of Computer Science and Technology,
North China Institute of Science and Technology, East Yanjiao, Beijing 101601, China

Abstract: This research presents an intrusion detection method for network datasets using Minimax Probability Machines (MPM) and genetic algorithm. The minimax probability machines can achieve the comparative performance with the Support Vector Machine (SVM). To do more accurate data classification and decrease the training time of classifier, we present a genetic feature optimized method for minimax probability machines. Genetic algorithm is used to optimize the feature so as to generate newly features to boost minimax probability machines do more accurate classification and need less training time. A new classifier model based on minimax probability machines with genetic feature optimized is proposed and is applied to intrusion detection in this paper. The experimental results show that the classification method with genetic feature optimized has better performance than the traditional learning method.

Key words: Intrusion detection, MPM, genetic algorithms, data classification

INTRODUCTION

The number of intrusions into computer systems and network is growing. It remains to be solved how to find out new intrusion behaviors effectively, which is important to protect the resource of the system and network. Intrusion detection system which is a component of network security measure is more and more concerned.

The main goal of intrusion detection is to detect unauthorized use, misuse and abuse of computer systems by both system insiders and external intruders. Over the past decade many intrusion detection techniques have been used to capture the system or user's normal usage pattern and classify new behavior as either normal or abnormal. Forrest *et al.* (1996) introduced the idea of building program profiles with short sequences of system calls issued by running programs for intrusion detection (Forrest *et al.*, 1996). Wenke Lee's research (Lee and Xiang, 2001) focuses on theoretical measures for anomaly detection. In contrast, unsupervised schemes can make appropriate labels for a given dataset automatically. Anomaly detection methods with unsupervised features are explained in (Ertoz *et al.*, 2004; Eskin *et al.*, 2002; Stamford *et al.*, 2002). MINDS (Ertoz *et al.*, 2004) are based on data mining and data clustering methods. The researches of Eskin *et al.* (2002) are used to detect anomalous attacks independent of pre-existing knowledge. Mukkamala *et al.* (2002) apply algorithm of Support Vector Machine (SVM) to intrusion detection and get better performance.

The MPM is a state-of-the-art classification algorithm. It can achieve the comparative performance and the same complexity of time with the SVMThe MPM method finds a bound on the misclassification probabilities. On the other hand, SVM finds a hyperplane that maximizes the distance between two classes. It was even as SVM showed, the computational complexity of MPM classifier is mainly rest with the scale of question. Moreover, when the numbers of features of network dataset become very large, the detection rates of IDS can be degraded, since it should process the large number of features of vast amount of dataset. So we tried to find out important features among the whole features of network dataset to minimize processing burden and maximize detection rates. To decrease the computational complexity and get more accurate data classification, we propose the genetic feature optimized algorithm to optimize network dataset in this paper. Whereafter we can construct MPM classifier with optimized dataset.

THE NECESSARY BACKGROUND

The Minimax Probability Machines (MPMs): The MPM is based on a theorem which, assuming positive definite covariance matrices for each of the two classes but

Corresponding Author: Zhen-Guo Chen, Department of Computer Science and Technology,
North China Institute of Science and Technology, East Yanjiao, Beijing 101601, China

without making any other statistical assumptions such as Gaussian distribution, guarantees a lower bound on the probability of misclassification for the classification approach of linear projection to one dimension followed by comparison to a decision boundary.

Let x and y denote random vectors in a binary classification problem, w ith mean vectors and covariance matrices given by X~ $(\bar{x}\ \Sigma_x)$ and covariance matrices for each of the two classes but y~$(\bar{y}\ \Sigma_y)$, respectively, where ~ means that the random variable has the specified mean and covariance matrix but that the distribution is otherwise unconstrained. Note that x, x, x, $\bar{x}$, y, $\bar{y} \in R^n$ and Σ_x, $\Sigma_y \in R^{n\times n}$

We want to determine the hyperplane $a^T z = b$ (a, z $\in R^n$ and b$\in$ R) that separates the two classes of points with maximal probability with respect to all distributions having these means and covariance matrices. This boils down to:

$$\max_{\alpha, a, b} \alpha \ s.t \inf \Pr\{a^T x \geq b\} \geq \alpha \tag{1}$$

$$\inf \Pr\{a^T y \leq b\} \geq \alpha$$

or,

$$\begin{aligned} \max_{\alpha, a, b} \alpha \ s.t & \ 1-\alpha \geq \sup \Pr\{a^T x \leq b\} \\ & 1-\alpha \geq \sup \Pr\{a^T y \geq b\} \end{aligned} \tag{2}$$

Consider the second constraint in formula (2). Recall the result of Bertsimas and Sethuraman (2000):

$$\sup \Pr\{a^T y \geq b\} = \frac{1}{1+d^2},$$

With

$$d^2 = \inf_{a^T y \geq b} (y-\bar{y}) \sum_y^{-1} (y-\bar{y}) \tag{3}$$

We obtain the following transformed Second Order Cone Programming:

$$\min \left\| \sum_x^{\frac{1}{2}} a \right\|_2 + \left\| \sum_y^{\frac{1}{2}} a \right\|_2 \ s.t$$

$$a^T (\bar{x}-\bar{y}) = 1. \tag{4}$$

We can solve this problem in various ways which yield a worst-case complexity of O (n^3). This is the same complexity as the quadratic programs one has to solve in support vector machine (Lanckriet *et al*., 2002).

Brief introduction to genetic algorithm: Genetic Algorithm (GA) is a kind of search and optimized algorithm that have been produced from simulating biologic heredities and long evolutionary processes of creatures. It stimulates the mechanism of "survival competitions; the superior survive while the inferior are eliminated, the fittest survive. The mechanism searches after the optimal subject by means of a successive iterative algorithm. Ever since the late 80s, GA, as a new cross discipline which has drawn people's attention, has already shown its increasing vitality in many fields (Yanfang and Zhongtuo, 1995).

GA stimulates reproduction, mating and dissociation in natural selection and natural heredity procedures. Each possible solution to problems is taken as an individual among population and each individual is coded as character string; each individual is evaluated in response to predefined objective functions and a flexibility value given. Three of its elemental operators are selection, crossing and mutagenesis (Genshe and Xinhai, 1994).

GENETIC FEATURE OPTIMIZED ALGORITHM

To improve the performance of MPMs, we propose GA to optimize feature set and get optimization feature.

Determination of the genetic encoding scheme: First, we should give the determination of a genetic encoding scheme, namely to denote each possible point in the problem's search space as a characteristic string of defined length. In order to simplify and improve the algorithm's global search ability, feature set are mapped to this very chromosome.

There are several ways to encode a chromosome; for example, Binary encoding, Real encoding and Order encoding. In this research, binary encoding is our choice. In binary encoding, GA applies strings of binary bits (1s and 0s) representing chromosomes to describe multiple points in the search space of the problem domain. They simulate natural evolution on populations of chromosomes during the search process. It is worth noting that GA solves the optimization problem by manipulation the genes in the chromosomes blindly without any knowledge about the information of the problem. The only information given is an evolution of the chromosome which is used to decide the selection of chromosomes so that better chromosomes can be reproduced. These features lead GA to be applicable to a wide variety of global optimization problems (Lis and Eliben, 1997; Sheble and Brittig, 1995). In this case, bit 1 means the feature is selected and bit 0 means the feature is not selected.

As we all known, the KDDCUP99 dataset has three symbolic features and 38 numerical features. According to our encoding scheme, each binary bit of chromosomes denotes the state of a feature. In the present study, we will use fixed length chromosomes. Therefore the length of chromosomes is 41. For example:

Let $x = (a_1, a_2, a_3, a_4, a_5, a_6)$, is a feature set, if the feature a_1, a_2, a_5, a_6, are selected, a_3, a_4, are not selected, then a chromosome can be expressed as 110011.

Fitness and performance evaluation of MPMs: The second key step is the definition of the fitness function to evaluate the problem-solving ability of the MPMs, which is denoted by a certain specific chromosome string. In this study, objective function is generated from the training time and false positive rate of the MPMs and then converted into a function via reciprocal transformation. Its computational formula is as follows:

$$f = (T - T_k) + \frac{N+1}{N_k+1} \quad (5)$$

Where:

T, N = Separately false positive number and training time before optimized

T_k N_k = Separately false positive number and training time at k times

Three of GA's elemental operators: In the system, genetic algorithms operations follow the algorithms described in literature(Michalewicz, 1996).

In select operation, the chromosomes are selected according to:

$$P_i = \frac{f_i}{\sum_{i=1}^{N} f_i} = \frac{f_i}{f_{sum}} \quad (6)$$

Where:

f_i = Fitness value of individual I

f_{sum} = Total fitness value of population

P_i = Selective probability of individual

In the crossover operation, two chromosomes are first selected randomly as the parents and then a single point is randomly chosen. Two parents then take part in the crossover at the crossover point to generate two children. The crossover rate in the experiment is 0.75.

In the mutation operation, some chromosomes are randomly chosen. Some genes of the chosen chromosomes are randomly selected and replaced by a random value within the range of (0, 1). The mutation rate is 0.15 in the experiment.

Table 1: MPMs with genetic feature optimized

Step 1:	Initializing chromosome according feature set and every parameter.
Step 2:	The affinity function is designed according to the formula (5).
Step 3:	Do Select Crossover and Mutation operation.
Step 4:	training MPMs according k-iter optimized feature set and get training time and false positive number computer affinity function.
Step 5:	Repeat Until i=*iterativeness* or not any improvements in the continuous t iterations.
Step 6:	Get final optimized feature set and employ MPMs with optimized feature to test.

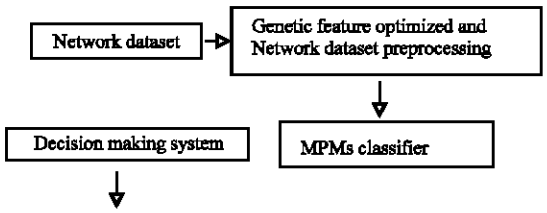

Fig. 1: The intrusion detection system models based on MPMs with genetic feature optimized

MPMs with genetic feature optimized: The learning procedure of the MPMs with genetic feature optimized is shown in Table 1.

In this study, we provide a definition of the intrusion detection model based on MPMs classifiers with genetic feature optimized.

The model is independent of any particular system, application environment, system vulnerability, or type of intrusion, thereby providing a framework for a general-purpose intrusion-detection system. Figuer 1 shows this model.

SIMULATIONS AND RESULTS ANALYSIS

Dataset for experiments: In this research, we use the dataset from KDD Cup 1999. The network traffic data are connection-based, meaning that each data record corresponds to one network connection. There are three symbolic features and 38 numerical features. In addition, a label will be provided to indicate whether the record is normal or abnormal. We will represent each symbolic feature by a group of binary-valued features.

Attacks fall into four main categories: DoS: A Denial of Service (DoS) attack is a class of attacks in which an attacker makers some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

R2L: A remote to user attack is a class of attacks in which an attacker sends packers to a machine over a network but who does not have an account on that machine; Exploits some vulnerability to gain local access as a user of that machine.

Table 2 Optimized features

Class	No.	Optimized result
Normal	25	10101101110001101001
		11011111000101101101
Probe	7	00101100000000000000
		01100000001100000000
DOS	19	10101101000000000100
		01111110001101101111
U2Su	8	00001100000001101000
		00010000001100000000
R2L	6	00101100000000000000
		00100000001100000000

U2R: User to root exploits are a class of attacks in which an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system.

Probing: Probing is a class of attacks in which an attacker scans a network of computes to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits.

Extract optimized feature: Here section, we get optimized feature set through the algorithm of genetic (Table 2). If the value is 1, then the feature is selected, if the value is 0, then the feature is not selected.

In Table 2, the number of feature has been decreased greatly. Certainly the result will be dissimilarity each time.

Experiments results: We use 5900 randomly sampled normal connection as training data to evaluate the performance of a model. Another 8000 randomly sampled normal connections form the threshold determination set, which has no overlap with the training set (Table 3).

Since the KDD Cup 1999 is concerned with multi-classification. So we construct separately five classifiers to train dataset.

In this study, we use two performance measures in our experiments. The True Acceptance Rate (TAR) measures the percentage of normal connections in the test set that are classified as normal and the True Detection Rate TDR measures the percentage of intrusive connections in the test set that are detected as intrusions.

To evaluate our method, we compare the result based on MPM-GA with the result based on traditional MPM, SVM and multi-layer MPM (Liu Fang and Zhenguo, 2004) in Table 3. We can see the intrusion detection system based on MPM+GA has better performance than other methods. Table 4 shows our method needs less training time than traditional MPM and SVM methods. But the whole time need a little more than multi-layer MPM (Liu Fang and Zhenguo, 2004), because of the genetic feature optimized algorithm need some time in training phase.

Table 3: Comparison of four methods in average performance

	TAR (%)		TDR (%)		
Method	Normal	Probing	DoS	U2R	R2L
MPM	97.71	92.71	93.35	95.85	96.85
SVM	97.73	92.70	93.45	95.87	96.83
multi-layer MPM	97.74	92.71	93.38	95.85	96.86
MPM+GA	98.85	95.70	93.94	96.86	96.82

Table 4: Comparison of four methods in training time

	Training time (sec)				
Method	Normal	Probing	DoS	U2R	R2L
MPM	7.88	49.85	22.96	3.79	11.81
SVM	7.86	49.73	22.99	3.78	11.84
multi-layer MPM	2.62	16.55	7.56	1.23	4.01
MPM+GA	3.61	23.98	11.88	1.82	5.78

CONCLUSION

We apply the genetic algorithm to optimize feature. Therefore, we employ MPMs with optimized feature to intrusion detection. The experiment results show that ours method achieves the better performance and decreases training time.

REFERENCES

Bertsimas, D. and J. Sethuraman, 2000. Moment Problems and Semidefinite Optimization. In: Handbook of Semidefinite Programming. Vol. 27. Int. Ser. Oper. Res. Manage. Sci. Kluwer Acad. Publ., Boston, MA., 27: 469-509.

Ertoz, L. *et al.*, 2004. MINDS-Minnesota Intrusion Detection System, Next Generation Data Mining Chapter 3.

Eskin, E., A. Arnold, M. Prerau, L. Portnoy and S. Stolfo, 2002. A Geometric Framework for Unsupervised Anomaly Detection Detecting. In: Intrusions in Unlabeled Data. Kluwer, Barbara, D. and S. Jajodia (Eds.). Applications of Data Mining in Computer Security.

Forrest, S., S.A. Hofmeyr, A. Somayaji and T.A. Longstaff, 1996. A Sense of Self for Unix Process, Proceeding of 1996. IEEE Symposium on Securiry and Privacy. IEEE Comput. Soc. Press, Los Alamitos, CA, pp: 120-128.

Genshe, C. and Z. Xinhai, 1994. Study and development of genetic algorithm. J. Inform. Control, 23: 215-221.

Lanckriet, G.R.G., L.E. Ghaoui and C. Bhattacharyya *et al.*, 2002. Minimax probability machine (A). Proceeding Advances in Neural Information Processing System (C), Berkeley: Department of EECS University of California, pp: 1-7.

Lee, Wenke and D. Xiang, 2001. Information-theoretic measures for anomaly detection (C). In: IEEE Symposium on Security and Privacy. Oakland, pp: 130-143.

Lis, J. and A.E. Eliben, 1997. A multi-sexual genetic algorithm for multiobjective optimization. In: Proc. IEEE Int. Conf. Evol. Comput., pp: 59-64.

Liu Fang, L and C. Zhenguo, 2004. Intrusion detection based on multi-layer minimax probability machine classifier. In: Proceedings of International Conference on Machine Learning and Cybernetics, Shanghai, China, 2580-2585.

Michalewicz, Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs, Springer, Verlag, Berlin, 1996.

Mukkamala, S., G.I. Janoski and A.H. Sung, 2002. Intrusion Detection Using Support Vector Machines, Proceedings of the High Performance Computing Symposium-HPC, San Diego, pp: 178-183.

Sheble, G.B. and K. Brittig, 1995. Refined genetic algorithm-economic dispatch example. IEEE Trans. Power Sys., 10: 117-124.

Staniford, S., J. Hoagland and J. McAlerney, 2002. Practical automated detection of stealthy portscans. J. Comp. Secu., 2002, 10: 105-136.

Yanfeng, S. and W. Zhongtuo, 1995. Parallel genetic algorithm. J. Syst. Eng., 13: 14-6.