Network-Based Intrusion Detection with Support Vector Machines

Dong Seong Kim[1] and Jong Sou Park[1]

Hankuk Aviation University
200-1, Hwajon-dong, Doekyang-gu, Koyang, Kyounggi-do, Korea
{dskim,jspark}@mail.hangkong.ac.kr

Abstract. This paper proposes a method of applying Support Vector Machines to network-based Intrusion Detection System (SVM IDS). Support vector machines(SVM) is a learning technique which has been successfully applied in many application areas. Intrusion detection can be considered as two-class classification problem or multi-class classification problem. We used dataset from 1999 KDD intrusion detection contest. SVM IDS was learned with triaing set and tested with test sets to evaluate the performance of SVM IDS to the novel attacks. And we also evaluate the importance of each feature to improve the overall performance of IDS. The results of experiments demonstrate that applying SVM in Intrusion Detection System can be an effective and efficient way for detecting intrusions.

1 Introduction

We propose network-based intrusion detection systems using Support Vector Machines(SVM). Intrusion is generally defined as violating confidentiality, Integrity, and Availability of computer or computer network system[1]. Intrusion detection system(IDS) detects automatically computer intrusions to protect computers and computer networks safely from malicious uses or attacks[2]. IDS should cope with a novel attacks as well as previously known attacks or misuses. Also IDS should give minimum overhead to computer system and IDS itself for processing audit data. But existing intrusion detection systems tend to have a low detection rates at novel attacks and high overhead itself to process audit data. SVM is relatively a novel classification technique proposed by Vapnik[3]. In many applications, SVM has been shown to provide higher performance than traditional learning machines, and has been introduced as powerful tools for solving classification problems such as pattern recognition and speech recognition fields [4,5,6]. Intrusion generally can be considered as a binary classification problem, distinguishing between normal and attacks. Intrusion can also be considered as multi-class classification problems. So we propose Network-based Support Vector Machine Intrusion Detection System(SVMIDS) . We used 1999 KDD dataset which was used in contest of intrusion detection areas [7,8]. We performed experiments on network-based SVM IDS with the various Kernel functions and regularization parameter C values. We also performed experiments with various

H.-K. Kahng (Ed.): ICOIN 2003, LNCS 2662, pp. 747–756, 2003.

numbers of features of dataset to evaluate the characteristics of feature values. Through analysis of features by using SVM, IDS can process network packet more effectively. The rest of this paper is organized as follows. A brief description of the Intrusion detection and the theory of SVM will be described in Section 2. The proposed structures of network-based SVM IDS will be suggested in Section 3. The various experiments of SVM IDS and their results are presented in Section 4. Some concluding remarks are given in Section 5.

2 Related Works

2.1 Intrusion Detection

Intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource. Intrusion detection system(IDS) attempts to detect an intruder break into computer system or legitimate user misuse system resources in real-time. Intrusion detection techniques based on intrusion model can be divided into two major types, misuse detection and anomaly detection[9,10]. In misuse detection, intrusions are well defined attacks on known weak points of a computer system. Misuse can be detected by watching for certain action being performed on certain object, concretely by doing pattern matching on audit-trail information. Misuse detection can cope with a known attack but not novel attacks because misuse detection depends on the intrusion database for known attack patterns. In anomaly detection, intrusions are based on observations of deviations from normal system usage patterns. They are detected by building up a profile of the system being monitored, and detecting significant deviations from this profile. A model is built which contains metrics that are derived from system operation. Metrics are computed from available system parameters such as average CPU load, number of network connections per minute, number of processes per user, etc. Anomaly detection also contains some problems. Since anomaly detection techniques use significant variation from user profiles or normal profile, intruders can manipulate to modify system profiles gradually. And Intrusion detection techniques based on data-source can be divided into two main types, host-based detection and network-base detection [10]. Host-based intrusion detection use audit data such as user's CPU usage time, command log, system call of system, etc. Because host-based intrusion detection can not have any information of network events in lower layers, it usually does not detect network-based attacks. According to the operating system or the platform of each host requires a different system, so it takes higher cost to develop. And a host-based intrusion detection use resources of itself, such as CPU time, storage, etc., so it lowers the performance of the system. On the other hand, network-based intrusion detection (NIDS) use audit data such as network packet data, especially network packet headers and payloads. Most NIDS generally collects network packets through using Network Interface Card (NIC) and takes passive analysis, so NIDS do not require software to be loaded and managed on a variety of hosts. NIDS detects attacks that host-based detection systems miss, e.g., many IP-based Denial-Of-Service(DoS)

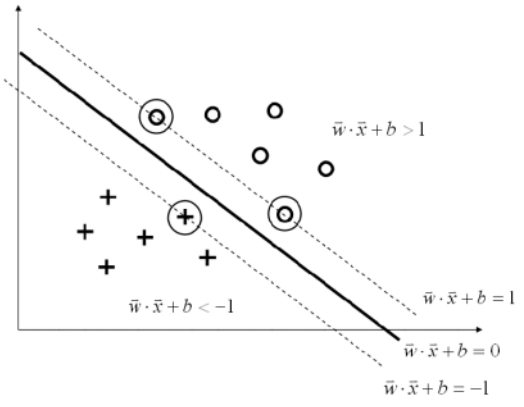

Fig. 1. Optimal seperating hyperplane is the one that seperates the data with maximal margin[3]

attacks and fragmented packet(Teardrop) attacks. We propose network-based intrusion detector which can detect both anomaly and misuse intrusions using Support Vector Classification Methods.

2.2 Support Vector Classification

Support Vector Machines (SVM) have been successfully applied to a wide range of pattern recognition problems[4,5,6]. SVM are attractive because they are based on well developed theory. A support vector machine finds an optimal separating hyper-plane between members and non-members of a given class in an high dimension feature space. Although the dimension of feature space is very large, infinite sometimes, they still show good generalization performances. This conflicts with our sense about "curse of dimension" and over-fit. Theories have been used to interpret this paradox, but more works still have to be done for these kinds of learning machines [5]

Two-Class Classification. Support vector machines are based on the structural risk minimization principle[11] from the statistical learning theory. In their basic form, SVM learn linear decision rules described by a weighted vector and a threshold . The idea of structural risk minimization is to find a hypothesis for which one can guarantee the lowest probability of error. Geometrically, we can find two parts representing the two classes with a hyper-plane with normal and distance from the origin as depicted in Figure 1.

$$D(\overrightarrow{x}) = sign\{\overrightarrow{w}\cdot\overrightarrow{x}+b\} = \begin{cases} +1, \text{ if } \overrightarrow{w}\cdot\overrightarrow{x}+b>0 \\ -1, \qquad \text{else} \end{cases} \tag{1}$$

Suppose such separating hyper-planes exist. We define the margin of a separating hyper-plane as the minimum distance between all input vectors and the

hyper-plane. The learning strategy of support vector machines is to choose the classifier hyper-plane with maximal margin. Although this might seem a reasonable heuristic approach, it is actually well founded in statistical learning theory. Basically, bounds on the generalization error are given in terms of the classification function's complexity, which we can minimize for linear functions by maximizing the margin. In particular, these bounds do not depend on the dimension of the input space, which makes this strategy appealing for high-dimensional data. To modify this approach for linearly inseparable data, we introduce slack variables and simultaneously maximize the margin and minimize the classification error on the training set. This includes introducing a regularization parameter that controls the trade-off between these two objectives. The resulting optimization problem is a quadratic program, usually solved in the form of its Lagrangian dual :

$$Min \quad \alpha \qquad \sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j x_i \cdot x_j - \sum_{i=1}^{m} \alpha_i \tag{2}$$

$$s.t. \qquad \sum_{i=1}^{m} y_i \alpha_i = 0, \qquad 0 \le a_i \le C, \quad i = l, L, M \tag{3}$$

In our experiment to solve two-class classification problem for intrusion detection, we used the well-known software SVMlight [19] for two-class intrusion detection classification.

Multi-class Classification. Support Vector Machines were originally designed for two-class classification, commonly called as binary classification. How to effectively extend it for multi-class classification is still an on-going research issue [12]. Currently there are two types of approaches for multi-class SVM, one-against-all[13] and one-against-one approaches[14]. One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation. More detailed description for multi-class classification methods are presented in [12]. To solve multi-class attack problems in our experiments, we used libsvm[20] which use one-against-one methods.

3 Network-Based Support Vector Machines Intrusion Detection System

3.1 Structure of Network-Based SVM IDS

The overall structure and component of SVM IDS are depicted in Figure 2[18]. We describe the overall structure of SVM IDS briefly. First, training dataset and test dataset is collected. We used training set and test dataset from KDD 99 dataset in intrusion detection areas [7]. These sets are preprocessed and to be used as standard input form of SVM. In this progress, we can select the features of the training set and test set. As the result of feature selection, we can estimate

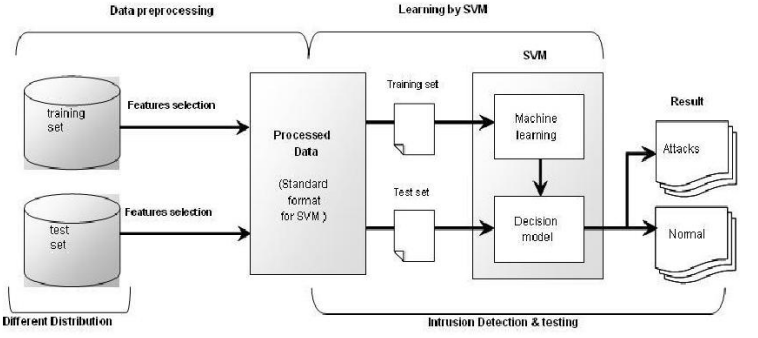

Fig. 2. Structure of Network-based SVM IDS

what features is important and we can use these characteristics of feature to improve the performance of SVM IDS. Training dataset is divided into two sets as learning dataset, validation dataset. With learning set, SVM is learned and as a result of learning, a decision model is generated. Actually decision model is a decision function, called hyper-planes in feature space, with a weight vector value and some numbers of support vectors. With validation dataset, which are different from learning set, we can evaluate how the decision model we already made perform classification task well. With test set, we can evaluate the performance of SVM decision model such about various novel attacks. The performance of a decision model depends on the detection rates, false positive rates, number of misclassifications.

3.2 Dataset and Preprocessing

Dataset. We used the datasets used in the 1999 KDD intrusion detection competition [7]. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between "bad"connections, called intrusions or attacks, and "good" normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment. More detail descriptions are presented in [8]. In their tasks, attacks fall into four main categories [16] :

- DOS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local root privileges, e.g., various "buffer overflow" attacks;
- Probing: surveillance and other probing, e.g., port scanning.

And Stolfo et. al. defined higher-level features that help in distinguishing normal connections from attacks[15]. There are several categories of derived features, such as "same host" features, "same service" features, "content" features. And the distribution of training set is different from that of test set because test set contains additionally 14 novel attacks that do not appear in training set.

Preprocessing. Those datasets were labeled on individual connections records called attack instances whether it's normal or attacks. In dataset, derived feature values are denoted as a discrete type, e.g., tcp, ftp, SF or continuous type, e.g., 5, 0.01, etc. So we preprocessed dataset to correspond with the standard input form of SVM. In experiments, we deleted features or add some features to evaluate the characteristics of the deleted or added features.

3.3 Learning and Testing of SVM

Learning Processes. Through training process, SVM became a decision model, actually a decision function with some numbers of support vectors and weight vector values. We used two datasets, learning sets and validation set. Learning sets are used to generate a decision function of SVM. And we can evaluate the generated decision function of SVM by using the validation set. Learning set and validation sets are made from training set randomly, but learning sets are different from the validation sets. We can evaluate the performance of decision function of SVM through validating the values of detection rate and misclassification rates. If detection rates are too low to accept, new learning process must be done. This process was iterated until a high detection rates are obtained. In learning process, we used various C values, e.g., 1, 500, 1000 and kernel functions such as linear, 2-poly and Radial Basis Function (RBF). Through modification of regularization parameter C values and kernel function, we can validate which kernel function is an effective and efficient one in this application area.

Test Processes. The distributions of training sets are different from test sets. In test set, there are about 14 new attacks which are not contained in training set. So we can evaluate the model made in learning progress how the models cope with various novel attacks. In KDD'99 contest, the evaluations of detection rates and false alarm rates were carried out, and the results were open to the public [8]. So we compared our results to KDD'99 results relatively and we demonstrated the improvement of our methods.

4 Experiments of SVM IDS and Results

We now describe the processes of experiments as three phases. In first phase, we processed the training sets and test sets suitable for the standard form of SVM. In second phase, SVM IDS was under learning using the preprocessed learning

sets, we validated a decision model by using the preprocessed validation sets. In third phase, we evaluated the detection rates of the decision model that had constructed in second phase by using test sets. In addition to these processes, we manipulated the number of the input features to investigate the importance of each input features. We also performed experiments on 5 classes classifications classified by DARPA projects.

4.1 Data Preprocessing

We used KDD 1999 dataset in intrusion detection of application areas. These dataset are for the purpose of evaluation to various proposed intrusion detection systems. The dataset are only classified dataset which labeled as normal or attack name, so these dataset is not suitable for standard form of SVM input. We transformed these dataset into a standard form of SVM input. The total numbers of training set is 4,898,431 instances and the total number of 10% labeled test set is 311,029 instances. We extracted learning set and validation set from training set randomly. But learning set is different from validation set. And we also extracted sub test sets from test set randomly

4.2 SVM Learning and Validation Experiments

We used SVMlight[19] for binary classification. We used linear, 2poly, and RBF functions as kernel functions. Our input dataset have 41 features and dimension is large, so we used various combination of kernel function and C values to find out the most suitable and efficient kernel to our application field. And the C values are regarded as a regularization parameter. This is the only free parameter in the SVM formulation. This parameter provides us trade-off between margin maximization and classification violation. Generally speaking, smaller parameter C makes a decision model simpler, if parameter C is infinite; all training data are classified correctly. The result of validation experiments are displayed in Table 1.

Table 1. The results of validation experiments

kernel	C	val 1.	val 2.	val 3.	val 4.
linear	0	93.56%	93.12%	93.45%	90.52%
2poly	0	49.97%	50.03%	50.02%	50.07%
	100	48.65%	49.97%	49.97%	49.97%
	1000	49.97%	50.03%	50.02	50.07%
RBF	0	50.03%	50.02%	49.97%	50.02%
	100	85.61%	85.69%	86.85%	82.59%
	1000	86.04%	85.63%	87.49%	82.35%

Table 2. The results of testing experiments

kernel	C	test 1.	test 2.	test 3.	test 4.
linear	0	90.16%	89.51%	87.71%	88.84%
RBF	1000	78.10%	77.23%	76.24%	76.51%

Table 3. The results of features deletion experiments

feature.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
det.(%)	77.20	78.10	76.24	77.23	78.05	76.55	78.05	76.10	77.20	76.23	78.10	77.40	78.30	76.11
feature.	15	16	17	18	19	20	21	22	23	24	25	26	27	28
det.(%)	76.2	78.05	78.34	78.05	76.40	75.96	78.03	76.24	76.51	78.05	77.23	78.22	77.34	78.25
feature	29	30	31	32	33	34	35	36	37	38	39	40	41	
det.(%)	76.1	76.68	78.05	78.05	78.05	78.05	77.32	76.70	79.00	78.14	77.86	76.12	76.65	

Table 4. The results of five-classes classification experiments

	All	Normal	Probe	DOS	U2R	R2L
Validation	95.58%	99.2%	29.54%	94.5%	16%	32%
Testing	93.59%	99.3%	36.65%	91.6%	12%	22%

4.3 SVM Testing Experiments

We used the model which had highest detection rates in section 4.2. There are 14 new attacks in test sets, they were not contained in the training set. We displayed the results of detection rates to the test data in Table 2.

4.4 Feature Deletion Experiments

We performed features deletion experiments and the results are summarized in Table 3. Through this feature deletion, we tried to find out the importance of features in intrusion detection system. But the results of these experiments are not so promising to find out which input features are important or useless.

4.5 Five Class Classification Experiments

We performed experiments on five class classifications. The results are displayed in Table 4. Detection rates about Normal, Probe, and DOS are comparatively high, but detection rates about U2R and R2L are so low because the numbers of instances in R2L and U2R attack are so small.

4.6 Comparison of Training and Testing of Results to '99 KDD Contest

We analyzed the performance of IDS using SVM previously. But for more correct verification, comparable method to our approach is needed. But in this paper,

Table 5. The performance comparisions between SVM IDS and KDD '99 winner

	SVM IDS	KDD 99 winner
Normal	99.3%	99.5%
Probe	36.65%	83.3%
DOS	91.6%	97.1%
U2R	12%	13.2%
R2L	22%	8.4%

we did not perform those work, so we compared our performance of SVM IDS to that of KDD 1999 contest winner relatively.

5 Conclusion and Future Works

In this paper, we proposed a network-based SVM IDS, and demonstrated it through results of 3 kinds of experiments. And we show that SVM IDS can be an effective choice of implementing IDS. In this paper, we used labeled dataset, it does not mean that our approach is not real-time intrusion detection. If network packet can be preprocessed in real-time, our approach can be use in real-time applications. And because there are a large mount of network packets to be processed in a real-time environment, we plan to implement hardware based SVM IDS chip to process faster than software. And there are some miss-classified input vectors and those degrade the performance of SVM IDS, so we need to improve the performance by applying Genetic Algorithm (GA) based feature extraction to enhance the performance of SVM. We can also apply decision tree method to get feature extraction instead of GA.

Acknowledgements

This work is supported by "Integrated Circuit Design Education Center(IDEC)", "Information Technology Research Center(ITRC)" supported by the Ministry of Information &Communication of Korea(supervised by IITA),"Internet Information Retrieval" Regional Research Center Program supported by the Korea Science and Engineering Foundation.

References

[1] J. P. Anderson., : Computer Security Threat Monitoring and Surveillance, James P Anderson Co.., Technical report, Fort Washington, Pennsylvania, April (1980)

[2] D. E. Denning. : An Intrusion Detection Model, IEEE Trans. S. E., (1987)

[3] V. Vapnik. : The Nature of Statistical Learning Theory, Springer, Berlin Heidelberg New York (1995)
[4] SVM Application List, http://www.clopinet.com/isabelle/Projects/SVM/applist.html
[5] Christopher J. C. Burges. : A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery (1998)
[6] . G. Guo, S. Z. Li, and K. Chan. : Face Recognition by Support Vector Machines, Fourth IEEE International Conference on Automatic Face and Gesture Recognition,(2000)196-201
[7] KDD Cup 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[8] Results of the KDD '99 Classifier Learning Contest, http://www-cse.ucsd.edu/users/elkan/clresults.html
[9] S. Kumar.: Classification and Detection of Computer Intrusions. Ph.D. Dissertation (1995)
[10] H. Debar, M. Dacier and A.Wespi. : A revised taxonomy for intrusion-detection systems, IBM Research Technical Report. (1999)
[11] V. N. Vapnik. : Statistical Learning Theory. John Wiley & Sons (1998)
[12] Chih-Wei Hsu, Chih-Jen Lin : A comparison of Methods for Multi-class Support Vector Machines, National Taiwan University.(2001)
[13] B. Schoelkopf, C. Burges, and V. Vapnik.: Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, Proceedings, First International Conference on Knowledge Discovery&Data Mining. AAAI Press, MenloPark, CA (1995)
[14] S. Knerr, L. Personnaz, and G. Dreyfus. : Single-layer learning revisited: a stepwise procedure for building and training a neural network. In J. Fogelman, editor, Neurocomputing: Algorithms, Architectures and Applications. Springer-Verlag. (1990)
[15] Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K. Chan. : Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project, Technical Rep. (2000)
[16] Intrusion Detection Attacks Database, http://www.cs.fit.edu/~mmahoney/ids.html
[17] Aurobindo Sundaram : An Introduction to Intrusion Detection, ACM crossroad Issue 2.4 April (1996)
[18] J. Lee, D. S. Kim, S. Chi, J. S. Park : Using the Support Vector Machine to Detect the Host-based Intrusion, IRC 2002 international conference (2002)
[19] SVM-Light Support Vector Machine, http://svmlight.joachims.org
[20] LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/