False Alarm Minimization Scheme based on Multi-Class SVM

Gil-Han Kim, and Hyung-woo Lee

Division of Computer information and Software, Hanshin University

Summary

The existing well-known network based intrusion detection/ prevention techniques such as the misuse detection technique, etc, are widely used. However, because the misuse detection based intrusion prevention system is proportionally depending on the detection rules, it causes excessive large false alarm which is linked to wrong correspondence. This study suggests an intrusion prevention system which uses *multi-class Support Vector Machines (SVM)* as one of the rule based intrusion prevention system and anomaly detection system in order to solve these problems. When proposed scheme is compared with existing intrusion prevention system, it show enhanced performance result that improve about 20% and propose false positive minimize with effective detection on new variant attacks.

Key words:
False Alarm, Intrusion Detection/Prevention, Multi-class SVM, Network Security.

1. Introduction

The popularization of the Internet is also bringing forth the increase of actions damaging the integrity, confidentiality and availability of computer resources and destroying security policies. Thus, we need to design various intrusion detection/prevention systems to protect system resources and data on network from intrusions.

Firewall, which is the 1st-generation solution of network security, is defenseless to permitted rules and vulnerable to user authentication based on ID and password, likely to expose passwords if a backdoor is installed. On the contrary, intrusion detection system (IDS), which is a 2nd-generation solution, is usually built behind firewall and analyze and detect attacks that detour the firewall, but it plays simply the role of CCTV, having limitations in detecting and preventing attacks at the same time. Thus, to solve these vulnerabilities, it is inevitable to introduce intrusion prevention system (IPS), a network security technology that copes with attacks actively and minimizes damage from the attacks[1].

IPS is an in-line tool that can decide whether to pass traffics or not based on the result of attack detection. Like IDS, IPS is divided into host-based IPS and network-based IPS according to data source. In addition, according to detection model, it is divided into misuse detection IPS and anomaly IPS. Misuse detection IPS detects attacks using rules defined by experts. It is widely used because of the high detection rate but is vulnerable to variant attacks or those not captured by the rules.

In addition, with the increase of rules, the rate of false detection (false positive) that detects a normal behavior as an attack also rises in geometrical progression. In IPS that can make an active response, these problems may cause wrong responses to normal/attack packets, which in turn may disrupt the flow of normal services. Thus, the removal of false alarms (false positive, false negative) is an important issue in improving IPS performance[2].

On the other hand, anomaly IPS detects attacks by analyzing users' patterns and comparing them with input patterns using machine learning or data mining rather than depending on experts' knowledge. This method is flexible in attack detection but its detection rate is low[3]. Thus, the present study proposes a method of minimizing false alarms while maintaining the attack detection rate of misuse detection IPS by applying the mutually complementary features of misuse detection IPS with high attack detection rate and IPS based on learning using multi-class SVM with 'normal' modeling function.

Chapter 2 introduces previous researches to reduce false alarms, and Chapter 3 explains SVM used as a learning tool in this research. Chapter 4 describes the intrusion prevention technique using SVM to be proposed in this paper, and Chapter 5 explains experiment on the proposed model and its results. Lastly, Chapter 6 draws conclusions and discusses future researches.

2. Previous Researches for False Alarm Minimization

In network security system, false alarm means false detection (false positive) that detects a normal behavior as an attack and miss detection (false negative) that judges an attack to be normal. False alarm causes the security system make a unnecessary response, spending resources and impairing the reliability of the security system. Thus, to reduce false alarms, researches have been made as follows.

2.1 System environment setting change

(1) Improving intrusion patterns

Normal packets are misunderstood as intrusion basically because intrusion patterns used in detection are wrong.

Manuscript received March, 2006.
Manuscript revised March 28, 2006

Thus, to reduce false positive, signature should be precise. It is difficult to make false positive that detects intrusions precisely in every situation. Thus, when false positive occurs, it may be informed to developers so that they improve signature and reduce false positive[4].

(2) Setting in accordance with system and network situation and policies
False positive can be reduced not by developers but by system managers through tuning the system in accordance with system and network situation. If rules against intrusions are set by default, normal packets may be misunderstood as hacking. Thus, the system manager should change the signature in accordance with the network setting through trials and errors.
In addition, false positive can be reduced by setting IPS environment fittingly to the protection policies of the corporation or organization and disabling signatures inconsistent with the protection policies.

2.2 Data mining

Data mining is a technology for extracting unknown useful knowledge from a large volume of data. It can be applied to data filtering as well as to sequence extraction for unknown types of attacks. Knowledge-based filter using data mining, which deals with important alarms, is reported to reduce false positive by around 30% in actual systems[5,6].

2.3 Correlation analysis technique

Correlation analysis technique installs different information protection systems and reduces false positive using information from the systems. It maintains a database for the vulnerabilities of each system to be protected and, if there is an alert, it moves to the alert and the resource DB, looks up vulnerabilities and, if there are identical vulnerabilities, it outputs the alert. This technique is dependent on selected attributes and is not suitable for exhaustive detection of the causal relation among alarm data[7,8].

2.4 Behavior-based analysis technique

When misuse detection IDS raises an alarm, this technique decides whether or not the alarm has to be informed to the manager using a learning or clustering method. Because it uses the information of packets commonly used as train data, it has less information loss and can train filtering modules appropriately for the network environment. Previous researches known so far used instance-based learning, which uses learning instances as they are, as their learning technique. However, this technique is disadvantageous in that, to classify new input data, it has to calculate the similarity of the new data to all train data and consequently it is costly and cannot cope with new attacks[9].

2.5 Problems in previous researches

Previous researches to reduce false detection (false positive) and miss detection (false negative) have problems such as vulnerabilities inherent in system management and structure and lack of abilities to cope with new attacks (Table 1). Thus, the present study proposes as a behavior-based analysis method using SVM an IPS model that minimizes false alarms and copes with new variant attacks effectively by analyzing and identifying false alarm patterns in four types of IPS detection results (true positive: identify attack as attack, false positive: identify normal as attack, true negative: identify normal as normal, false negative: identify attack as normal) in order to detect new attacks without the loss of train data.

Table 1: Problems in previous researches

Change of environment setting	Difficult to manage when network environment changes frequently, and difficult to use when protecting individual systems with different degree of vulnerability
Data mining	Cannot avoid information loss due to its structure, neglecting attacks that occur infrequently, so lack abilities to identify anomaly and cope with new patterns of attacks
Correlation analysis	Dependent on selected attributes, and not suitable for exhaustive exploration of causal relations among alarm data
Behavior-based analysis	Cost a lot to calculate, dependent on data, cannot cope with new patterns of attacks

3. Support Vector Machine (SVM)

Support vector machine (SVM) was a learning algorithm developed and proposed by Vapnik in 1995. While traditional learning algorithms are based on empirical risk minimization (ERM) to minimize empirical errors of the learning group, SVM is based on structural risk minimization to minimize the probability of wrong classification of data of fixed but unknown probability distribution[10].

3.1 Linear SVM - separable case

The purpose of SVM is to infer a function that distinguishes two classes with given train data. SVM learning is a process of finding the linear optimal separating hyperplane(OSH) under the constraint of maximizing the distance between the points of the two classes.

To make it possible to separate linearly the sets of learning vectors belonging to the two classes, the system should learn a training data set $\{(x_i, d_i)\}_{i=1}^{N}$ to have hyperplane $(w_0^T \bullet x) + b_0 = 0$ composed of weight vector w and bias b. Here, x_i is an input pattern and d_i is a target value. Hyperplane $(w_0^T \bullet x) + b_0 = 0$ satisfies the condition of Equation (1).

$$\exists w, b \quad s.t. \begin{cases} w^T x_i + b > 0 & for \quad d_i = +1 \\ w^T x_i + b < 0 & for \quad d_i = -1 \end{cases} \qquad (1)$$

In Equation (1), input patterns that satisfy the condition of the equal sign and are positioned closest to the decision surface are called support vectors. Conceptually, because these vectors are closest to the hyperplane, they are difficult to be separated. Thus, learning for separation is to find the optimal hyperplane that satisfies the constraint of Equation (2). This is a problem of optimization with constraints. It is a quadratic problem to find the optimal values of parameter w and b for the optimal hyperplane when training data set $\{(x_i, d_i)\}_{i=1}^{N}$ is given.

$$\begin{cases} Minimize \quad \Phi(w) = \frac{1}{2}\|w\|^2 \\ s.t. \quad d_i(w^T x_i + b) \geq 1 \qquad for \quad i = 1, ...N \end{cases} \qquad (2)$$

Here, the values that have the maximum margin are the optimal values, and the maximum margin hyperplane can separate two classes optimally. Consequently, if the optimal separating hyperplane is expressed as $g(x) = w_0^T \bullet x + b_0$, the distance between a support vector and g(x) is $1/\|w\|$, and the hyperplane that classifies input patterns optimally minimizes cost function $\Phi(w)$ as in Equation (3).

$$\Phi(w) = \frac{1}{2}\|w\|^2 \qquad (3)$$

The cost function Equation (3) is a block function of w, and constraint Equation (2) is linear to w. Summing up SVM for classification described above, this is a problem of optimization to find weight vector w and bias b satisfying constraint Equation (2) with given learning patterns and, here, the optimal separating hyperplane is found by maximizing the distance of separation through minimizing $\|w\|^2$. When the problem of optimization is solved using the Lagrange coefficient method, it is dualized with a Lagrange multiple as in Equation (4) and becomes a quadratic problem as below.

$$\theta(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j = 0$$

$$s.t. \quad \alpha_i \geq 0, \; i = 1, ..., N \;\; and \;\; \sum_{i=1}^{N} \alpha_i d_i = 0 \qquad (4)$$

3.2 Linear SVM - inseparable case

When learning samples cannot be completely separated into two classes by a linear separating hyperplane, it is inevitable to allow wrong classification. For this, slack variable (ξ) is added to the problem of optimization as expressed below.

$$Minimize \quad \tau(w, \xi) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \xi_i \qquad (5)$$

$$s.t. \quad d_i(< w, x_i > + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0, \;\; i = 1, ..., N$$

That is, it allows for objects to be placed in the margin of two parallel hyperplanes but with penalty parameter C.

C, which is a variable working as a penalty on non-separable data, is in a trade-off relation with model complexity. That is, if C is large the trained machine tends to provide a solution forming the optimal hyperplane and it is close to 0 it provides the effect of optimizing the margin maximization term. As a result, it does not give weight to the term of minimizing misclassification errors and thus produces a SVM classifier with a very large margin width.

3.3 Nonlinear SVM

Most patterns are not linearly separable. Thus, to classify nonlinear patterns, we need to convert the input space of nonlinear patterns into a specific space of linear patterns.

$$\theta(\alpha) = \sum_{i} \alpha_i^N - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j K(x_i, x_j)$$

$$s.t. \sum_{i}^{N} \alpha_i d_i = 0, \quad 0 \leq \alpha_i \leq C, \; \forall i. \qquad (6)$$

By obtaining Lagrange multiple i from the model above, we can get Equation (7) below, the most plane function in the specific space.

$$f(x) = \mathrm{sgn}(< w, \phi(x) >, + b)$$

$$= \mathrm{sgn}(\sum_{i=0}^{N} \alpha_i d_i K(x_i, x) + b) \qquad (7)$$

SVM provides kernels as in (Table 2) to support non-linear mapping functions.

Table 2 : Types of kernel functions

Type of kernel	Kernel	Remarks
Dot kernel	x•y	Inner product of x and y
Polynomial kernel	$(x \bullet y+1)^d$	D=1,2,3...

RBF (Radial Basis Function) kernel	$\exp(-g \| x - y \|^{2})$	g is the parameter determining the shape of kernels
Perceptron kernel	$\tanh(ax * y + b)$	a,b are constants satisfying Mercer's condition

4. False Alarm Minimization using SVM

4.1 Structure of SVM-based intrusion prevention system

In order to minimize false detection (false positive) and miss detection (false negative) in misuse detection IPS (Snort_inline), the intrusion prevention system to be proposed in this research is composed of multi-class SVM modules trained with the results of rule-based detection, which are classified into 4 classes (true positive: identify attack as attack, false positive: identify normal as attack, true negative: identify normal as normal, false negative: identify attack as normal). The structure of the system is as in (Figure 1).

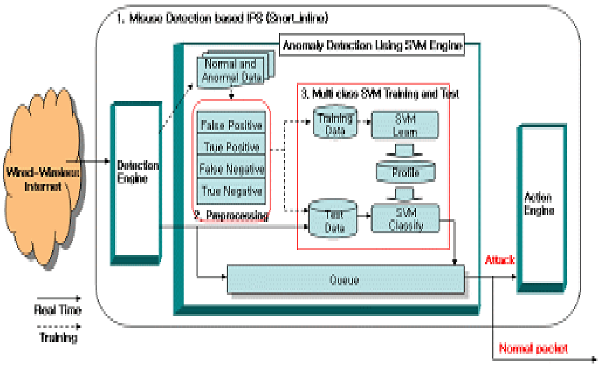

Fig 1. Structure of SVM-based intrusion prevention system

In IPS, network packet collection is made at Snort_inline of promiscuous mode, and whether abnormal traffic or not is determined by rules in the detection engine. Because detected packets include false detection packets as well as miss detection packets, they are filtered through the SVM module for multi-class classification according to the 4 classes trained off-line, and only attack packets identified as true positive and false negative are handled by the action engine in a safe way.

4.2 Misuse detection IPS (Snort_inline)

Snort_inline is one of patch versions of Snort well-known as an intrusion detection system and is used as IDS gateway or NIPS (network intrusion prevention system). Snort_inline inspects abnormal packets using a rule file for attack packets, and uses IPTables because it needs a simple mechanism for packet routing. Packets are obtained from the kernel space by IPTables, and are transferred to Snort_inline through the ip_queue function [11]. This supplements the weak point of Snort that detects intrusions by capturing packets passing through the network at random, enabling real-time packet maintenance and control. In this research, we installed an intrusion prevention system with Snort_inline-2.3.0-RC1 for experiment, and the structure of the system is as in (Figure 2).

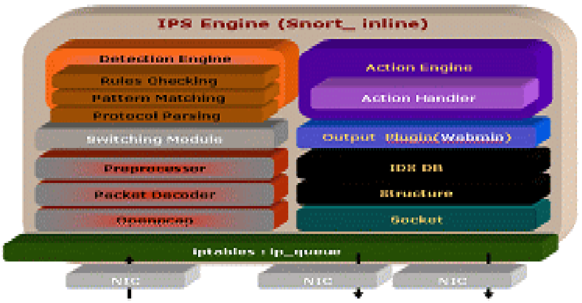

Fig 2. Advanced Snort_inline architecture

4.3 Preprocess data set

(1) Data set of DARPA 1998

The data of DARPA 1998 used in this research is a standard audit data set prepared by MIT Lincoln Labs for developing intrusion detection programs, containing extensive various types of intrusions tested on an artificial experimental military network.

For developing and evaluating intrusion detection systems, DARPA data set provides Solaris-based BSM audit data and tcpdump data for the period from 1998 to 2000, around 7 weeks per year, and 5 days per week (Monday ~ Friday). Tcpdump data itself was made simply by storing packets passing through the network, so it is impossible to distinguish attacks from normal packets just with tcpdump data. Thus, a separate list file, which identifies attacks included in the tcpdump data, is provided together. In addition, the data set of 1998 contains 27 kinds of attacks, which can be classified into four large classes (DoS, Probing, R2L and U2R).

(2) Conversion of SVM train data set

In order to apply the raw tcpdump data provided in DARPA 1998 as training/test data to a SVM learning algorithm and to draw precise classification results, it is most important to select the features of packets used as train data. In this research, to apply behavior models to each packet, we generated a train data per packet of tcpdump type (using -vv option), and a train data is composed of the set of features as in (Table 3). A total of 26 features were extracted from packet headers.

Table 3 : Packet features for SVM learning

Index	Feature	Type value
label		+1, -1
1	ToS	Real
2	TTL	Real
3	IP_ID	Real
4	Offset	Real
5	Fragment	DF, MF etc
6	Protocol	TCP, UDP, ICMP, IGMP, EGP, OSPF, IGRP, GRE, etc
7	IP_Length	Real
8	Source IP Address	Consider DARPA network
9	Source Port	Echo(7), Discard(9), Users(11), Quote(17), Nameserver(53), Bootps(67), Bootpc(68), TFTP(69), RPC(111), NTP(123), SNMP(161), SNMP_trap(162), FTP_data(20), FTP(21), TELNET(23), SMTP(25), DNS(53), Finger(79), HTTP(80). Rlogin(513), Rsh(514), etc
10	Destination IP Address	Consider DARPA network
11	Destination Port	Echo(7), Discard(9), Users(11), Quote(17), Nameserver(53), Bootps(67), Bootpc(68), TFTP(69), RPC(111), NTP(123), SNMP(161), SNMP_trap(162), FTP_data(20), FTP(21), TELNET(23), SMTP(25), DNS(53), Finger(79), HTTP(80). Rlogin(513), Rsh(514), imap(143), ssh(22), etc
12	DgmLength	Real
13	TCP_Length	Real
14	Sequence number	Real
15	Acknowledgement	Real
16	UGR	0,1
17	ACK	0,1
18	EOM	0,1
19	RST	0,1
20	SYN	0,1
21	FIN	0,1
22	Window Size	Real
23	UDP_Length	Real
24	ICMP_Type	Real
25	ICMP_Code	Real
26	ICMP_Length	Real

4.4 SVM learning model for false alarm minimization

The SVM learning model for minimizing false detection and miss detection in misuse detection IPS is applied to SVM to classify four detection types defined by Snort_inline (TP, FP, TN, FN). Basically, however, in order to apply SVM, which performs binary classification, to this study having 4 classes, strategies using binary classification combination should be presented. In this study, we designed and experimented two multi-class SVM models for minimizing false alarms that adopted one-against-all [12] and one-against-one [13] methods composed of the combination of binary SVMs.

(1) Application of one-against-all method
One-against-all (OAA) method uses k binary SVMs to classify k classes. Each SVM is trained with train data to identify a class. In our research problem that distinguishes TP, FP, TN and FN, the structural diagram is as in (Figure 3). The first SVM identifies FP. In order to distinguish class FP from other classes TP, TN and FN, train data corresponding to FP have +1 and the others have -1. Then, for test data, each SVM is given the same input data, and the output values from the SVMs are compared, and the data is identified as the class of the SVM that produced the largest output value.

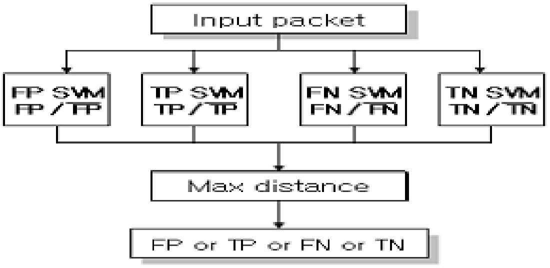
Input packet
FP SVM FP / $\overline{FP}$
TP SVM TP / $\overline{TP}$
FN SVM FN / $\overline{FN}$
TN SVM TN / $\overline{TN}$
Max distance
FP or TP or FN or TN

Fig 3. Structure of multi class SVM using one-against-all method

(2) Application of one-against-one method
Different from OAA, One-against-one (OAO) method uses k(k-1)/2 binary SVM to identify k classes. Each train data is divided into two classes. For our research problem, the structural diagram is as in (Figure 4). The first SVM has train data composed of class FP and class TP, and it also classifies only class FP and class TP in test data. When classifying test data, all of the SVMs performs classification and test data is identified as the class with the largest number of votes.

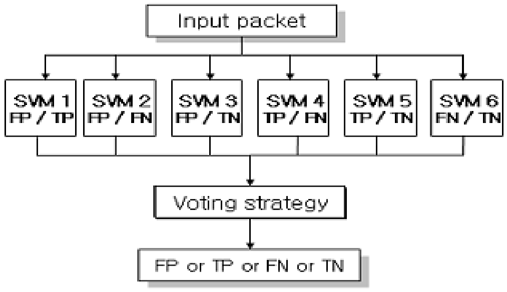

Fig 4. Structure of multi-class SVM using one-against-all method

5. Analysis of Experimental Results

5.1 Composition of experiment data

In order to evaluate and analyze the performance of the intrusion prevention system using SVM proposed in this study to minimize false alarms in existing intrusion prevention systems, first, we need to analyze the results of detection using an intrusion prevention system based on misuse detection (Snort_inline).

From tcpdump files and attack list files by day of each week in DARPA data set to be used as learning and test data, we extracted normal tcpdump data and anomaly tcpdump data by comparing attack time including duration, source IP address and destination IP address. The tcpdump data classified into normal and anomaly was used as input data to identify false alarm patterns and measure precision in Snort_inline, and 4 patterns (TP, FP, TN, FN) of data set are generated by comparing with Alert_list detected by the rules in the detection engine of Snort_inline as in (Figure 5). This data is used as train data for SVMs that identify the 4 classes. To set more specific feature values of packets, we converted the data into tcpdump data using '-vv' option in the experiment.

We obtained packets in the form of tcpdump divided into 4 classes from data for Friday of Week 2, Wednesday and Friday of Week 3, Tuesday and Wednesday of Week 4, Tuesday and Wednesday of Week 6 and Wednesday of Week 7 in DARPA 1998 data set, and took 1000 packets for each class so a total of 4000 packets as SVM train data. In addition, as test data for testing the performance of the classification system, we used data not included in the train data, namely, data for Tuesday of Wee2, Monday and Tuesday of Week 3, Friday of Week 4, Monday and Friday of Week 5, Friday of Week 7 in DARPA 1998 data set. SVM_light[14] was used as a SVM experiment tool.

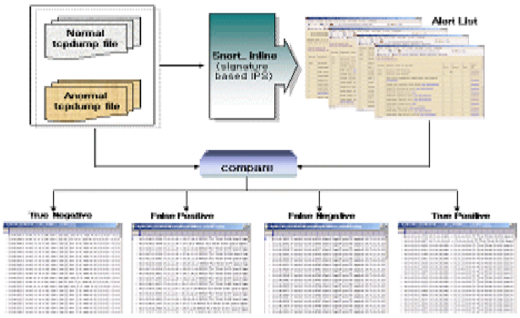

Fig 5. Dividing into 4 classes

Table 4 : Train/test data set for SVM

	TP data	FP data	TN data	FN data	
Train Set	1000 packets	1000 packets	1000 packets	1000 packets	4000 packets
Test Set	300 packets	300 packets	300 packets	300 packets	1200 packets

5.2 Experimental results and analysis

(1) SVM results according to feature selection

For correct classification, it is important to select appropriate features of packets to be classified. (Figure 6) below shows the performance trend of the classifier for the test data when the number of feature parameters of the train data in (Table 3) is 6, 12, 16, 22 and 26. Precision is high when the train data used information of over 16 packet features. This suggests that 16 items of packet header information including TOS, TTL, packet length, fragment, protocol, port number and IP address were considered most importantly.

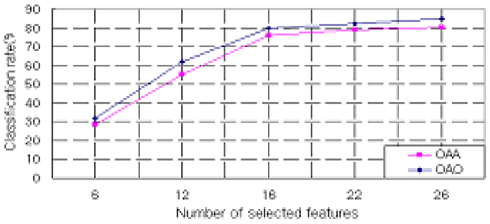

Fig 6. Performance trend of multi-class SVM according to the number of features selected

(2) SVM results according to kernel function

To evaluate the performance of multi-class SVM when the number of parameters in train data was 26, at which classification performance was highest in the prior experiment, we calculated the classification rate of each model and the precision and recall rate of each class classified. (Table 5) shows the results of classification by each model, and Equation (8) expresses the precision and recall rate of Class$_i$ identified by multi-class SVM.

Table 5 : Contingency table for Class$_i$

Class$_i$		True Label	
		Yes	No
Classifier judgement	Yes	C_TP	C_FP
	No	C_FN	C_TN

$$P_i = \frac{C\_TP_i}{C\_TP_i + C\_FP_i} \quad , \quad R_i = \frac{C\_TP_i}{C\_TP_i + C\_FN_i} \qquad (8)$$

In the test environment, we used kernel functions dot, polynomial and RBF (Radial Basis function). We showed the results of applying integer 1 and 4 for the parameter degree(d) of the polynomial kernel and the results of applying real number 0.01 and 0.5 for parameter gamma(g) of the RBF kernel, which showed the best results using empirical information through multiple experiments.

(Table 6) and (Table 7) below are the results of experiment with multi-class SVM model based on OAA and OAO methods for the two kernel functions using train data composed of all of the 26 feature parameters.

According to the results, SVM using OAO method produced somewhat better results than that using OAA method and this is probably because, due to the characteristics of SVM intended for binary classification, classification patterns are calculated more precisely in data composed of classes in equal ratio. In addition, when degree 4 was applied to the polynomial kernel, the classification rate for the 4 classes was highest (84.91%). In addition, precision and recall rate for each class were also as high as 90% on the average but classification was not clear between FN (false negative) and TN (true negative), namely, between attack packets that are not detected in Snort_inline and normal packets. This may be because, in preparing the test data of FN class, we included new attack data not included in the train data to evaluate the ability to detect variant attacks and new attacks.

Table 6 : Result of testing OAA (one-against-all)

Kernel	Parameter	C	C_TP	C_FP	C_FN	C_TN	P(%)	R(%)	A(%)
Dot	-	FN	95	2	205	898	97.93	31.66	82.75
		FP	300	0	0	900	100	100	
		TN	298	204	2	696	59.36	99.33	
		TP	300	1	0	899	99.66	100	
Polynomial	d=1	FN	72	2	228	898	97.29	24	80.83
		FP	300	2	0	898	99.33	100	
		TN	298	226	2	674	56.87	99.33	
		TP	300	0	0	900	100	100	
	d=4	FN	88	3	212	897	96.70	29.33	82.08
		FP	300	4	0	896	98.68	100	
		TN	297	203	3	697	59.4	99	
		TP	300	5	0	895	98.36	100	
RBF	g=0.01	FN	72	2	228	898	97.29	24	80.83
		FP	300	1	0	899	99.66	100	
		TN	298	225	2	685	56.97	99.33	
		TP	300	2	0	898	99.33	100	
	g=0.5	FN	85	3	215	897	96.59	28.33	81.83
		FP	300	0	0	900	100	100	
		TN	297	213	3	687	58.23	99	
		TP	300	2	0	898	99.33	100	

C=class, P=precision, R=recall, A=classification rate=4 class hit number(C_TP) / total test data

Table 7 : Result of testing OAO (one-against-one)

Kernel	Parameter	C	C_TP	C_FP	C_FN	C_TN	P(%)	R(%)	A(%)
Dot	-	FN	96	6	204	894	94.11	32	82.5
		FP	300	2	0	898	99.33	100	
		TN	294	202	6	698	59.27	98	
		TP	300	0	0	900	100	100	
Polynomial	d=1	FN	96	6	204	894	94.11	32	82.5
		FP	300	2	0	898	99.33	100	
		TN	294	202	6	698	59.27	98	
		TP	300	0	0	900	100	100	
	d=4	FN	122	3	178	897	97.6	40.66	84.91
		FP	300	2	0	898	99.33	100	
		TN	297	176	3	724	62.79	99	
		TP	300	0	0	900	100	100	
RBF	g=0.01	FN	94	5	206	895	94.94	31.33	82.41
		FP	300	2	0	898	99.33	100	
		TN	295	204	5	696	59.11	98.33	
		TP	300	0	0	900	100	100	
	g=0.5	FN	87	3	213	897	96.66	29	82
		FP	300	0	0	900	100	100	
		TN	297	213	3	687	58.23	99	
		TP	300	0	0	900	100	100	

C=class, P=precision, R=recall, A=classification rate=4 class hit number(C_TP) / total test data

(3) Comparison of performance

In order to compare the performance of SVM-based IPS proposed in this study to minimize IPS false alarm, we applied the test data used in the multi-class SVM model to case-based training technique using k-NN algorithm. The comparative experiment used TiMBL (Tilburg Memory Based Learner, version 5.1)[15], and the performance of Snort_inline, SVM and k-NN for the test data was expressed in harmonic mean (F-measure) combining precision and recall rate. F-measure is as Equation (9) below.

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} \qquad (9)$$

When comparing F-measure resulting from the experiment, false detection and miss detection rate decreased significantly in existing IPS as shown in (Table 8). Here, the detection rates for attacks and normal packets are not much meaningful because they did not consider false detection and miss detection. The high detection rate of Snort_inline despite many false alarms may be because then number of packets used in the experiment was much larger than that in the comparative experiment.

Table 8 : Results of test classification for comparing performance

	Rate(%)	FP(%)	FN(%)	P(%)	R(%)	F(%)
Snort_ inline	89	6.8	20.36	52.36	79.56	63.15
OAA	82.75	0.33	34	99.49	65.94	79.31
OAO	84.91	0.5	29.6	99.29	70.33	82.33
1-NN	79	0	42	100	58	73.41

Rate=(attack detection + Normal detection)/total test data
FP=false positive/normal, FN=false negative/attack, P=precision, R=recall, F=F-measure

6. Conclusions and future research

To apply SVM, which is intended for binary classification, to the classification of 4 detection patterns in misuse detection IPS, we designed and experimented two multi-class SVM models using one-against-all (OAA) and one-against-one (OAO) methods. According to the results of experiment, classification was more precise in multi-class SVM using OAO method than in that using OAA method.

In conclusion, the classification system reduced the false detection rate for test data by 99% and miss detection rate by 40.6%. In addition, it improved the performance of SVM based on Snort_inline by around 20% compared to that when using only Snort_inline as a single system. Furthermore, the proposed system had not only misuse detection but also anomaly detection effect without sacrificing genuine alarms. Thus, the proposed system processes only genuine alarms in the action engine by minimizing alarms against attacks generated by IPS and, in addition to rule-based detection, can detect new attacks intelligently. However, to apply the model to actual systems, we need to measure its performance quantitatively in real-time systems. In addition, a larger volume of train data is necessary to enhance the efficiency of anomaly detection.

References

[1] Jo Hyeon-jeong, "Intrusion prevention system based on next-generation network security technology," Journal of Information Science Association, Volume 23, No. 1, p21-26, 2005

[2] C.Kruegel and T. Toth "Using decision trees to improve signature-based detection." In6th Symposium on Recent Advances in Intrusion Detection(RAID), Lecture Notes in Computer Science. Springer Verlag. USA. September, 2003

[3] Gary Golomb. IDS v. IPS Commentary, Linuxsecurity.com News, 6/16/2003, http://www.linuxsecurity.com/articles /forums_article- 7476.html

[4] Internet Security System. "The Truth about False Positive." White Technical Report. 2001

[5] R. Lippman et als., "Evaluation intrusion detection system : The 1998 DARPA Off-line intrusion detection evaluation." Proc. Of DARPA Information Survivability Conference and Exposition, pp.12-26, 2000

[6] K. Julisch. "Mining alarm clusters to improve alarm handling efficiency," In 17th Annual Computer Security Application Conference(ACSAC), pp12-21, 2000

[7] Cuppens, F., Miege, A. "Alert correlation in a cooperative intrusion detection framework", In Proceedings of the IEEE Symposium on Security and Privacy, 2002

[8] H. Debar, A.Wespi, "Aggregation and Correlation of intrusion-Detection Alert", In Recent Advances in intrusion Detection, number 2212 in Lecture Notes in Computer Science, p85-103, 2001

[9] S. Manganaris, M. Christensen, D. Zerkle and K. Hermiz, "A Data Mining Analysis of RTID Alarms," In 2nd Work-shop on Recent Advances in Intrusion Detection(RAID99), 1999

[10] Campbell, C and Cristianini, N. "Simple Learning Algorithms for Training Support Vector Machines", Technical report, University of Bristol, 1998

[11] http://snort-inline.sourceforge.net

[12] Hsu, C.W. and Lin, C.J. "A Comparison of Methods for Multi-class Support Vector Machines," IEEE Transaction on Neural Networks. Vol. 13. No.2. pp 415-425, 2002

[13] Knerr. S., Personnaz, L. and Dreyfus, G. Single-layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network. in Neuro-computing: Algorithms. Architectures and Applications. J. Fogelman, Ed. Springer-Verlag, New York, 1990

[14] Christopher J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, 1998

[15] Daelemans, W., Zavrel, J. van der Sloot, K, and van denBosch, A., TiMBL:Tilburg Memory Based Learner, version 5.1, Reference Guide, Technical Report 01-04, Induction of Linguistic Knowledge, Tilburg University, 2001