Using Rough Set and Support Vector Machine for Network Intrusion Detection System

Rung-Ching Chen and Kai-Fan Cheng
Chaoyang University of Technology
Taichung Country, Taiwan, R.O.C
E-mail: crching@cyut.edu.tw

Ying-Hao Chen and Chia-Fen Hsieh
Chaoyang University of Technology
Taichung Country, Taiwan, R.O.C
E-mail:s9614627@cyut.edu.tw

Abstract—***The main function of IDS (Intrusion Detection System) is to protect the system, analyze and predict the behaviors of users. Then these behaviors will be considered an attack or a normal behavior. Though IDS has been developed for many years, the large number of return alert messages makes managers maintain system inefficiently. In this paper, we use RST (Rough Set Theory) and SVM (Support Vector Machine) to detect intrusions. First, RST is used to preprocess the data and reduce the dimensions. Next, the features selected by RST will be sent to SVM model to learn and test respectively. The method is effective to decrease the space density of data. The experiments will compare the results with different methods and show RST and SVM schema could improve the false positive rate and accuracy.***

Keywords-Rough Set; Support Vector Machine; Intrusion Detection System; Attack Detection Rate;

I. INTRODUCTION

The intrusion behaviors cause the great damage of systems. So enterprises search for intrusion detection systems to protect their systems. The traditional technology such as firewall is used to defense attacks. Thus, the IDS (Intrusion Detection System) is usually used to enhance the network security of enterprises.

The major difference between firewall and IDS system is that firewall is a manual passive defense system. Comparatively, IDS could collect packets online from the network. After collecting them, IDS will monitor and analyze these packets. So, IDS system acts as the "second line of defense". Finally, it will provide the detecting results for managers. The detecting results could be either attack or normal behavior. An ideal IDS system has a 100% attack detection rate along with a 0% false positive rate, but it is hard to achieve. Detecting illegal behaviors of the host or network is the major object of IDS. The IDS is actually such a system to detect some illegal behaviors. One of the ability of IDS is it could monitor various activities on the network. IDS will send a warning message to the managers if it detects an attack. Briefly, the aim of intrusion detection is to identify malicious attacks. There are two main methods of IDS: misuse and anomaly [9]. The idea of misuse detection is to establish a pattern or a signature form so that the same attack can be detected. It could apply to our personal computers, just like the antivirus programs. The main drawback of misuse detection is it cannot detect new attacks but there will be many various new attacks generation. On the other hand, misuse has a pattern database. This database includes possible signature of attacks. If the system matches the data with the attack pattern, the IDS regards it as an attack. Consequently, misuse detection provides a lower false positive rate.

The other idea here is to establish a normal activity profile for system. Anomaly detection utilizes soft computing methods to detect attacks. For instance, Neural network[8], Static analysis, Data mining [6] etc. Anomaly detection can detect new attacks, but it has a higher false positive rate.

Large number of transmission packages will lead to computation overloading and worse performance. In this paper, we purpose an intrusion detection method to reduce the features of transferring packages using the method of Rough Set Theory. If the number of the key features is decreased correctly, the noise of data to affect the systems analysis performance will be lower. Primary experiment results prove our research can improve the attack detection rate.

The remainders of the paper are organized as follows. Section 2 presents the related literature of intrusion detection systems. Section 3 introduces our methodology. Section 4 shows the experiments results and Section 5 is conclusions and future works.

II. RELATED WORKS

A. The type intrusion methods

The IDS improves the attack detection rate (ADR) and decrease the false alarm rate (FAR) [5][6][12]. The Denial-of service (DOS) attacks also called the Distribution Denial-of Service (DDOS) attack is shown in Figure 2. The attacker uses large numbers of computers to login system in a short period of time or to transfer mass numbers of packages. These actions will lead to overloading of the host. The network services of host will stop. Large resource of the host such as the utilization of CPU and memory and the networks bandwidth…etc are consumed by the attack. SYN-flood, Smurf, Teardrop and Ping of Death are belong to the DoS attack.

978-0-7695-3580-7/09 $25.00 © 2009 IEEE
DOI 10.1109/ACIIDS.2009.59

IEEE computer society

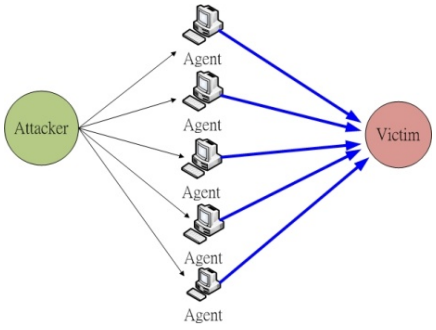

Figure 1 The schema of DDoS/DoS.

The intruders find the user's account or the system setting out of gear. The R2L attack tries to intrusion the system. SQL Injection is one kind of the R2L attacks.

U2R attack uses the unauthorized account to control system. The methods of U2R use the virus, worm, to overload buffer of memory to exceeding the limitation of memory access. For instance, the several of Buffer, Overflow Attacks are typical traditional U2R attack

Hackers scan the protocol of the computer before they launch attacks by port Scan action. The hackers find the weak spots and software design weakness to intrude a system. Port Scanning and the Ping-Sweep are typical traditional port scan attack.

B. Intrusion detection system

IDS system monitors the packages transmissions on the network. While malice behaviors have happen, IDS will send an alert to the network manager or use a related method to defense the attacks. Most intrusion detection systems are classified as either a NIDS (Network Based Intrusion Detection System) or a HIDS (Host Based Intrusion Detection System) [4][11]. In general, NIDS is located between host and firewall. HIDS was usually installed on a server or main computer as shown in Figure 2 NIDS collects and analyzes the information at the host. NIDS could monitor the data real time on the network. If NIDS finds illegal behaviors, it will send messages to the managers. Comparatively, HIDS monitors the activities of the host. So it can determine whether an attack or not. The data of HIDS is caught by the host, so it is not easy to be influenced by some methods, just like encryption. Entropy has been used in intrusion detection for a long time. B. Balajinath et al. used entropy in learning behavior model of intrusion detection in 2001[1]. TF-IDF is often applied to IDS, too. Such as Wun-Hwa Chen et al. compared SVM to ANN for intrusion detection, their methods are based on TF-IDF. The Unauthorized Access from a Remote Machine (R2L), and The Unauthorized Access to Local Super-user Privileges (U2R) both are intrusion behaviors which will be detected by HIDS.

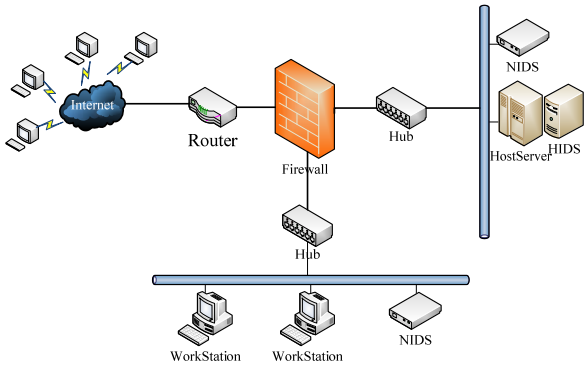

Figure 2 The hybrid system of IDS.

III. THE METHODOLOGY

The flowchart is our intrusion detection method shown as Figure 3 which comprised three steps. First, data preprocessing and data discretion are utilized to do data arrangement. Next, the RST is used to find useful features. Finally, the system uses the SVM to classify the data [1][10], which will be described as follows.

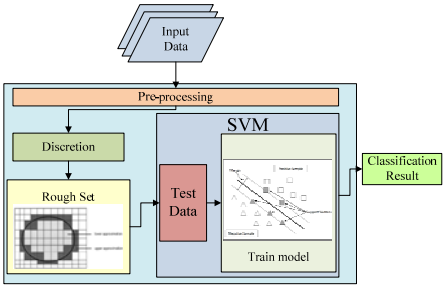

Figure 3 The workflow of the system in training phase.

A. Preprocessing

In this section, we will introduce the KDD cup'99 and use the database to test the system performance. The KDD Cup 1999 Data is original from 1998 DARPA Intrusion Detection Evaluation [7]. A process can be composed of many system calls. A system call is a text record. In this phase, some useless data will be filtered and modified. For example, some text items need to be converted into numbers.

Every process in the database has 41 attributes shown in Table 1.

Table 1 KDD cup'99 features

No.	Features	No.	Features
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login		

This dataset have been used for the Third International Knowledge Discovery and Data Mining Tools Competition. The task of this competition was to build a network detector to find "bad" connections and "good" connections. For "bad" connections, the attack has categories, shown in Table 2.

An example of KDD cup'99 [7] is shown in Figure 4. There are several text words in the dataset. The system will transform text into numeric values in advance. For example, the service type of "tcp" is mapping to 3 and the system will follow Table 3 to transform it into the numeric form, as described in Figure 5.

```
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,normal.
0,tcp,http,SF,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.0
0,0.00,0.00,0.00,1.00,0.00,0.00,19,19,1.00,0.00,0.05,0.00,0
.00,0.00,0.00,0.00,normal.
0,tcp,http,SF,235,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.
00,0.00,0.00,0.00,1.00,0.00,0.00,29,29,1.00,0.00,0.03,0.00,
0.00,0.00,0.00,0.00,normal.
```

Figure 4 The original data of KDD cup'99

```
0,3,19,10,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0,0,0,
0,1,0,0,9,9,1,0,0.11,0,0,0,0,0
0,3,19,10,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0,0,0,0
,1,0,0,19,19,1,0,0.05,0,0,0,0,0
0,3,19,10,235,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0,0,0,
0,1,0,0,29,29,1,0,0.03,0,0,0,0,0
```

Figure 5 After transform from original data

Table 2 Data types and attacks classification

Attack types	class	Attack types	class
Normal	Normal	guess_passwd	R2L
apacha2	DoS	Imap	
back		multihop	
land		named	
mailbomb		phf	
netune		sendmail	
pod		snmpgetattack	
processtable		snmpguess	
smurf		spy	
teardrop		warezclient	
udpstorm		warezmaster	
buffer_overflow	U2R	worm	
httprunnel		xlock	
loadmodule		xsnoop	
perl		Ipsweep	Probe
ps		mscan	
rootkit		nmap	
sqlattack		portsweep	
xterm		saint	
ftp_write	R2L	satan	

Table 3 Transformation table

Types	Class	No.
Protocol_type	TCP	3
Protocol_type	UDP	7
Protocol_type	ICMP	9
Flag	OTH	1
Flag	REJ	2
Flag	RSTO	3
Flag	RSTOS0	4
Flag	RSTR	5
Flag	S0	6
Flag	S1	7
Flag	S2	8
Flag	S3	9
Flag	SF	10
Flag	SH	11
Attack or Normal	Attack	1
Attack or Normal	Normal	0

B. Feature selection by rough set

Using the RST reduces the attributes for SVM operation. Rough Set Theory [2][3][12][13] is one of data-mining methods which reduces the features from large numbers of data. Using RST needs to build the decision table or the information table. The decision table describes the features of processes. Formally, an information system IS (or an approximation space) can be shown as follows.

$$IS = (U, A)$$

Where U is the Universe (a dataset of process, $U=\{x_1,x_2,x_3,x_4,x_5,x_6,\ldots,x_m\}$) and A presents the attributes of a

process, for instances, (A={a_1,a_2,a_3,a_4,a_5}). The definition of an information function is f_a: U→ V_a, V_a is the set of values of the attributes. For example, the values of U and A are listed as follows and they are mapping to V_i.

U= {x_1, x_2, x_3, x_4, x_5, x_6, …, x_m}
A= {a_1, a_2, a_3, a_4, a_5}

V1= {1, 2, 3, 4}
V2= {1, 2, 3, 4, 5}
V3= {1, 2, 3, 4, 5}
V4= {1, 2, 3}

For every set of attributes B ⊆ A, if b (x_i) = b (x_j)(every b $_{\subseteq B}$), there is an indiscernible relation Ind(B). Continuous, to define the basic concepts, namely the Upper approximations and Lower approximations of a set let X represents the elements of subset of the universe U (X $_{\subseteq U}$). The lower approximations of X in B ($B_{\subseteq A}$) represents as $\overline{BX}$ such as follows.

$$\overline{BX}=\{X_i \in U | [X_i]_{ind(B)} \subset X\} \quad (1)$$

The lower approximations of set X of process xi, which contained X of elementary set in the space B.

The upper approximation of set X is BX, BX represents the union of the elementary which is a non-empty intersection with X.

$$BX= \{ X_i \in U | [X_i]_{ind(B)} \cap X \neq 0 \} \quad (2)$$

For any object xi of lower approximation of X ($x_i \in \overline{BX}$), it is certainly belongs to X. For object of xi of upper approximations of X ($xi_{\in BX}$), it is called a boundary of X in U. The difference of upper and lower approximations is:

$$BNP = BX - \overline{BX} \quad (3)$$

If the upper and lower approximations are identical (BX = $\overline{BX}$), the set X is definable; otherwise, set X is indefinable in U. There are four types of the set of indefinable in U. Ø represents an empty set.

If $\overline{BX} \neq$ Ø and BX ≠ U, the set of X represents roughly definable in U;

If $\overline{BX} \neq$ Ø and BX=U, the set of X represents externally indefinable in U;

If $\overline{BX}$ = Ø and BX ≠ U, the set of X represents internally indefinable in U;

If $\overline{BX}$ = Ø and BX ≠ U, the set of X represents totally indefinable in U.

Using all attributes to do intrusion detection is ineffective. In this paper, RST is used to combine the similar attributes and to reduce the number of attributes. So it can enhance the processing speed and to promote the detection rate for intrusion detection. An example of RST is shown in Figure 6.

O:1,0,3,19,10,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0,0,0,0,1,0,0,9,9,1,0,0.11,0,0,0,0,0
O:2,0,3,19,10,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0,0,0,0,1,0,0,19,19,1,0,0.05,0,0,0,0,0
O:3,0,3,19,10,235,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0,0,0,0,1,0,0,29,29,1,0,0.03,0,0,0,0,0

Figure 6 The Rough set input data form

C. *Intrusion estimation*

In this paper, we construct an SVM [5] model for classification. While intrusion behaviors happens. SVM will detect the intrusion. SVM uses a high dimension space to find a hyper-plane to perform binary classification, where the error rate is minimal. The SVM can handle the problem of linear inseparability. The SVM uses a portion of the data to train the system, finding several support vectors that represent the training data. These support vectors will be formed a model by the SVM, representing a category. According to this model, the SVM will classify a given unknown document. A basic input data format and output data domains are listed as follows.

$$(x_i, y_i), \ldots, (x_n, y_n), x \in R^m, y \in \{+1, -1\} \quad (4)$$

Where $(x_i, y_i), \ldots, (x_n, y_n)$ are a train data, n is the numbers of samples, m is the inputs vector, and y belongs to category of +1 or -1 respectively.

On the problem of linear, a hyper plan can divided into the two categories as shown in Figure 7. The hyper plan formula is:

$$(w \cdot x) + b = 0 \quad (5)$$

The category formula is:

$$(w \cdot x) + b \geq \text{if } y_i = +1 \quad (6)$$

$$(w \cdot x) + b \leq \text{if } y_i = -1 \quad (7)$$

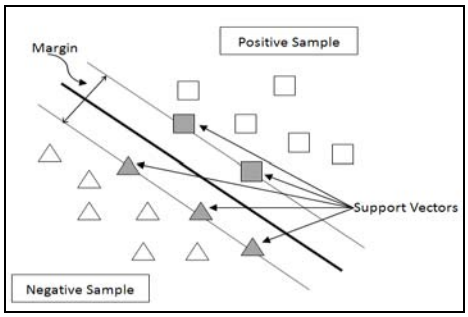

Figure 7 The hyper-plane of SVM.

However, for many problems they are not easy to find hyper planes to classify the data. The SVM has several kernel functions that users can apply to solving different problems. Selecting the appropriate kernel function can

solve the problem of linear inseparability. Another important capability of the SVM is that it can deal with linear inseparable problems. Internal product operations affect the classification function. A suitable inner product function K ($X_i \cdot X_j$) can solve certain linear inseparable problems without increasing the complexity of the calculation.

IV. EXPERIMENTS AND DISCUSSIONS

The experiment used a AMD Athlon™ 64 X2 Dual Core Processor 5000+ 2.59G MHz computer with 512MB RAM, and implemented on a Windows XP Professional operating system.

The data was collected from the MIT Lincoln Lab, 1998 DARPA intrusion detection evaluation program. The data set offers seven weeks of training data and two weeks of testing data. We randomly selected some of the data for training data and testing data. The DARPA data is labeled with a session number, each session including several processes, which in turn consist of system calls. A total of 24,701 processes were used as training data, while 15,551 processes were used as testing data. We have 41 features initially, which are presented and calculated their frequency in this database. Libsvm was used as our classification tool [2].

The total of 41 features, entropy and 29 RST feature values are used to train the three SVM models.

After the preprocessing process, data are formatted for RST. Then, the data of discretion is imported the RST by analysis tool. RST upper and lower approximation is utilized to seek the relation and un-relation of feature attributes. The features were selected by RST which are shown in Table 5.

Table 4. The training data and test data

	Train	Ratio	Test	Ratio
Normal	4863	19.69%	3029	19.48%
Probe	205	0.83%	208	1.34%
DoS	19572	79.24%	11492	73.90%
U2R	2	0.01%	11	0.07%
R2L	56	0.23%	809	5.20%
Total	24701	100.00%	15551	100.00%

Protocol type, Services, and Flag are transformed into numerical values. The LIBSVM data format is:

LIBSVM format: [Label] 1: f1 f2 f3 f4…fn. (8)

Before using SVM classification [14], we need to do "scaling". This action is to increase the accuracy, decrease the overlap, and reduce complexity [5]. Our system uses the kernel of RBF $K(x, y)= \ell^{-\gamma\|x-y\|^2}$.

In the SVM classify, we will take the tool to do classification the attributes of features before the preprocessing step. We use the SVM to train and to test processes. Base on the estimation enhance operate in the intrusion detections, the estimation formula is listed as follows.

The output of SVM is 1 or -1. If the output is 1, it is an intrusion behavior on the model. If the output is -1, it is normal. Using Libsvm tool [5] to classification estimate intrusion or not.

Table 5. The features after RST operation

No.	Features	No.	Features
1	Duration	16	rerror_rate
2	protocol_type	17	same_srv_rate
3	src_bytes	18	diff_srv_rate
4	dst_bytes	19	srv_diff_host_rate
5	wrong_fragment	20	dst_host_count
6	num_failed_logins	21	dst_host_srv_count
7	logged_in	22	dst_host_same_srv_rate
8	num_compromised	23	dst_host_diff_srv_rate
9	root_shell	24	dst_host_same_src_port_rate
10	num_root	25	dst_host_srv_diff_host_rate
11	num_file_creations	26	dst_host_serror_rate
12	num_shells	27	dst_host_srv_serror_rate
13	num_access_files	28	dst_host_rerror_rate
14	Count	29	dst_host_srv_rerror_rate
15	serror_rate		

The research object is to increase the accuracy of SVM. We use the RST to reduce features. SVM use to archive the supervisor learning and find a suitable kernel of SVM classification. Using the test data evaluates system performance. The data distribution of training data and testing data are shown in Table 5. User can use this model to evaluate the IDS. We use (1) all 41 features, (2) entropy features and 29 RST features values to train three SVM models respectively, and to compare the accuracy of the three models.

To estimate the performance of the system, three important formulas are used to evaluate system accuracy [10]; attack detection rate (ADR), false positive rate (FPR) and system accuracy (SA).

Attack Detection Rate=

$$\frac{Total\ number\ of\ attacks}{Total\ number\ of\ detected\ attacks} \times 100\% \quad (9)$$

False Positive Rate =

$$\frac{Total\ number\ of\ misclassified\ processes}{Total\ number\ of\ normal\ processes} \times 100\% \quad (10)$$

Accuracy Rate=

$$\frac{Total\ number\ of\ correct\ classified\ processes}{Total\ number\ of\ processes} \times 100\% \quad (11)$$

The experiments tested attack detection rate, false positive rate and accuracy among 41 features input SVM, Entropy inputs SVM, and after Rough Set reduces feature input SVM. Accuracy of 41 features input to SVM is 86.79% but the ADR is only 70.03%. The result of this study is not well because the feature is not reduction of attributes and the loading is too heavy for the system. The Entropy features input SVM has using reduction of feature and the ADR reaches 92.44% but the accuracy is only 73.83%. The accuracy of our proposed method is the best but its false position rate and attack detection rate are worse than Entropy to SVM. Table 6 shows the results.

Table 6. Comparison of three methods using SVM

	Attack Detection Rate	False Positive Rate	Accuracy
41 features to SVM	70.03%	29.97%	86.79%
Entropy to SVM	92.44%	7.56%	73.83%
Rough Set of SVM	86.72%	13.27%	89.13%

V. Conclusions and Future Works

In this paper, we have proposed an intrusion detection method using an SVM based system on a RST to reduce the number of features from 41 to 29. We also compared the performance of the SVM with that of a full features and Entropy. Our framework RST-SVM method result has a higher accuracy as compared to either full feature or entropy. The experiment demonstrates that RST-SVM yields a better accuracy.

In the future, we will increase number of testing data for our system and to find vary of accuracy. We also hope to combine RST method and genetic algorithm to improve the accuracy of IDS.

References

[1] B. Balajinath, and S. V. Raghavan, Intrusion Detection Through Learning Behavior Model. Computer Communications vol. 24, pp.1202-1212, 2001, 2001.

[2] C. Chang and C. J. Lin, LIBSVM -- A Library for Support Vector Machines.

[3] C. L. Huang and W. C. Tang, The Application of Genetic Algorithm to Attribute Selection and Discretization for Rough Set Theory. Information Management Thesis, Hua Fan University Taiwan, 2003.

[4] D. Y. Yeung and Y. Ding, Host-based intrusion detection using dynamic and static behavioral models. Pattern Recognition, vol. 36, pp. 229-243, 2003.

[5] Guide of LIBSVM. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[6] L. C. Wuu, C. H. Hung and S. F. Chen, Building Intrusion Pattern Miner for Snort Network Intrusion Detection System. Journal of Systems and Software, vol. 80, Issue 10, pp. 1699-1715, 2007.

[7] MIT,http://www.ll.mit.edu/IST/ideval/data/1998/1998_data_index.html, 2008.

[8] N. Toosi and M. Kahani, A New Approach to Intrusion detection Based on An Evolutionary Soft Computing Model Using Neuro-Fuzzy Classifiers. Computer Communications, vol. 30, Issue 10, pp. 2201-2212, 2007.

[9] O. Depren, M. Topallar, E. Anarim and M. K. Ciliz, An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse Detection in Computer Networks. Expert Systems with Applications, vol. 29, pp. 713-722, 2005.

[10] R. C. Chen and S. P. Chen, Intrusion Detection Using a Hybrid Support Vector Machine Based on Entropy and TF-IDF. International Journal of Innovative Computing, Information and Control (IJICIC) Vol. 4, Number 2, pp. 413-424, 2008.

[11] S. Rubin, S. Jha, and B. Miller, Automatic Generation and Analysis of NIDS Attacks. Proceedings of 20th Annual Computer Security Application Conference, IEEE Computer Society, vol. 00, pp. 28-38, 2004.

[12] W. T. Wong and Y. C. Chang, A Hybrid Approach of Rough Set Theory and Genetic Algorithm for SVM-based Intrusion Detection. Information Management Thesis, Chung Hua University Taiwan, 2005.

[13] Z. Pawlak, Rough Sets, International Journal of Computer and Information Sciences, vol. 11, pp. 341-256, 1982.

[14] Z. Zhang and H. Shen, Application of Online-training SVMs for Real-time Intrusion Detection with Different Considerations. Computer Communications, vol. 28, pp. 1428-1442, 2005.