CLASS SPECIFIC FEATURES FOR ATTACKS IN NETWORK INTRUSION DETECTION SYSTEM

Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin
Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia.
{anazida@utm.my, aizaini@utm.my and mariyam@utm.my}

Abstract: Most of the existing Intrusion Detection System (IDS) uses all the features to determine whether an input does have an intrusive pattern or otherwise. Some of these features are redundant and some have little contribution to the detection process. The purpose of this study is to identify small number of significant features that can represent most of the attack types. Here, we used Kohonen SOM to classify the input data into their respective attack categories. Empirical results indicate that generic feature subset previously obtained is not suitable to represent all the attack categories. Instead, different categories of attacks best represented using different significant feature subset.

Keywords : Intrusion Detection System, Feature selection, Class Specific Features, Classification and Kohonen SOM.

1. INTRODUCTION

The primary goal of any intrusion detection system is to be able to do correct detection in a reasonably short period so that an attack can be stopped or counter measure can be taken before a computer network is compromised. Various intelligent paradigms have been used in intrusion detection. Among them are Neural Network [1-4], Support Vector Machine [1][5-6] and Artificial Immune System [7-8]. Statistical methods have also been explored to solve problems of accuracy in IDS. [9-11] use Hidden Markov Model and [12] uses Chi Square and Canberra Distance. Though accuracy is a primary concern, lengthy detection is not desirable. Often accuracy is traded-off with high processing time. Since the ability of IDS to identify intrusive pattern in real time with accuracy is a concern [13], this paper addresses the issue of selecting significant features for intrusion detection. Reducing these input features may lead to faster training and possibly more accurate result.

Research in finding best feature subset has been intensified in early 2000. Both statistical and machine learning approaches were popularly used. [14] have used Bayesian Network and Classification Regression Tree and [15-16] used Flexible Neural Tree. Their purpose is to

uncover only the meaningful features from abundant data. Particle Swarm Optimization (PSO) is a population-based search algorithm and initialized with a population of particles having a random solution. PSO has been successfully applied to a large number of optimization problems such as [18] travelling salesman problem (NP-hard). Literature also pointed out that the discrete PSO (DPSO) is often outperformed Genetic Algorithm [19]. Meanwhile, Rough Set Theory (RST) has been successfully used as a selection tool to discover data dependencies and reduced the number of attributes contained in a dataset by purely structural method [20]. The objective of this paper is to propose minimal significant feature subsets for different attack categories. Rough-DPSO was used as feature selection tool and classification was done using Kohonen SOM.

The organization of this paper is as follows: Section 2 discusses the importance of feature reduction in IDS, Section 3 presents data used in this study, Section 4 discusses feature selection in IDS, Section 5 presents classification using SOM. Section 6 discusses the experiment conducted. Section 7 offers the results and findings. Finally, Section 8 concludes the paper and gives the direction of this study.

2. IMPORTANCE OF FEATURE REDUCTION

IDS normally deals with abundant of data and analysis becomes complex when extraneous features need to be examined in order to look for known intrusive patterns. This huge network traffic data causes a prohibitively high overhead and often becomes a major problem in IDS [13]. Input features may be redundant and complex relationships may exist between these features. Usually, an intrusive behavior has some patterns that are unique and recognizable. But, these common properties are often hidden within the irrelevant features and some features contain false correlation [14]. Therefore, these input features have to be reduced in order to disclose the hidden significant features. Thus, an accurate and fast classification can be achieved.

3. DATA

KDD Cup 1999 was obtained from 1998 DARPA intrusion detection evaluation program, an environment which was set up to acquire raw TCP/IP dump data for a network simulating a typical U.S. Air Force LAN. The LAN was operated as a real environment and injected with multiple attacks. Each TCP/IP connection has 41 qualitative and quantitative features. Some are derived features. We used the labeled data for both training and testing. The total data amounting to 485,797 and 80% of them are attacks. The distribution of attack is given in the figure below.

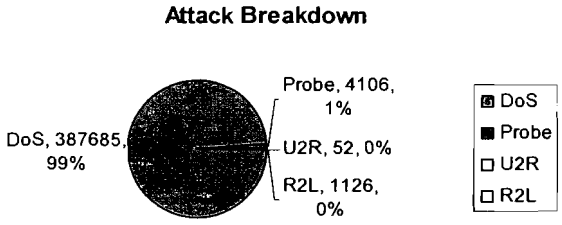

Figure 1 : Data Distribution

Attacks fall into four categories:

i) **Dos: Denial of Service.** This kind of attack consumes a lot of computing and memory resources and denying the legitimate requests. The means of achieving this are varied from buffer overflows to flooding the systems resources. Examples are Back, Land, Neptune, Pod, Smurf and Teardrop

ii) **U2R: User to Root** (unauthorized access to super user privilege). Attack starts out with normal user accessing the system and gradually exploiting system vulnerabilities to gain super user access. Examples are Buffer_overflow, Loadmodule, Perl and Rootkit.

iii) **Probe** (surveillance)

Attacker scans the network to gather information about the network and find the system's known vulnerabilities. These vulnerabilities will be exploited to attack the system. Examples are Ipsweep, Portsweep, Satan and Nmap.

iv) **R2L: Remote to Local** (unauthorized access from a remote to local machine). An attacker who does not have an account exploits some systems' vulnerabilities to gain local access. Examples are ftp_write, guess_passwd, imap, multi_hop, phf, spy, warezclient and warezmaster.

4. FEATURE SELECTION

Feature selection can be used to filter out noise and remove redundant and irrelevant or misleading features in a dataset [21]. For better representation, features are labeled as in Table 1.

Table 1: Feature Labeling

Label	Network Data Features	Label	Network Data Features	Label	Network Data Features
A	duration	**O**	su_attempted	**AC**	same_srv_rate
B	protocol_type	**P**	num_root	**AD**	diff_srv_rate
C	service	**Q**	num_file_creations	**AE**	srv_diff_host_rate
D	flag	**R**	num_shells	**AF**	dst_host_count
E	src_byte	**S**	num_access_files	**AG**	dst_host_srv_count
F	dst_byte	**T**	num_outbound_cmds	**AH**	dst_host_same_srv_count
G	land	**U**	is_host_login	**AI**	dst_host_diff_srv_count
H	wrong_fragment	**V**	is_guest_login	**AJ**	dst_host_same_src_port_rate
I	urgent	**W**	count	**AK**	dst_host_srv_diff_host_rate
J	hot	**X**	srv_count	**AL**	dst_host_serror_rate
K	num_failed_login	**Y**	serror_rate	**AM**	dst_host_srv_serror_rate
L	logged_in	**Z**	srv_serror_rate	**AN**	dst_host_rerror_rate
M	num_compromised	**AA**	rerror_rate	**AO**	dst_host_srv_rerror_rate
N	root_shell	**AB**	srv_rerror_rate		

4.1 Existing Work

The work of [22] showed that rough set classification attained high detection accuracy (using GA) and the feature ranking was fast. But they did not mention the features that they obtained. Similarly, [14] tackled the issue of effectiveness of an IDS in terms of real-time and detection accuracy from the feature reduction perspective. In their work, features were reduced using two techniques, Bayesian Network (BN) and Classification and Regression Trees (CART). They have experimented using four sets of feature subset which are 12, 17, 19 and all the variables (41) from one network connection. The work suggested that there is no generic feature subset. Instead, different features with different length were proven to be good for different types of attacks. Details of their findings can be found in [14]. Using the same dataset, Sung and Mukkamala [13] ranked six significant features. They used three techniques; Support Vector Decision Function Ranking (SVDF), Linear Genetic Programming (LGP) and Multivariate Regression Splines (MARS) and compared their performances on classification accuracy. For detail results, please refer to [16]. Similar to [6], [23] proposed different feature subset to best represent different types of attacks. The trust of their work was on maximizing the inter-classes seperability using genetic algorithm. Based on these reported works, it can be concluded that there are features that really significant in classifying the data.

4.2 Rough-Discrete Particle Swarm Optimization

Our approach utilized both Rough Set Concept and Discrete Particle Swam to perform feature selection. We called it Rough-DPSO. Figure 2 illustrates the flow of our feature selection procedure. RSC and DPSO were implemented in hierarchical manner.

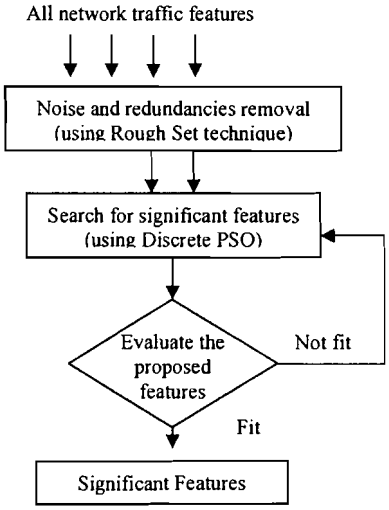

Figure 2: Feature Selection Procedure

The novel aspect of our work is the combination of both statistical and evolutionary techniques in selecting significant features for intrusion detection. Synergizing these two techniques leads to faster convergence and optimum solution is found in lesser iterations. Details can be found in [24-25].

5. CLASSIFICATION USING SOM

Based on the significant features obtained in Phase I, Kohonen SOM was used to classify the data into the respective categories.

5.1 Self Organizing Map (SOM)

The SOM algorithm performs a topology preserving mapping of the input data from its high dimensional data space onto a two dimensional grid. So that the relative distances between data points are preserved. Data points, which closely resemble each other are located to nearby map nodes that form 2-dimensional lattice. The pseudo code of Kohonen SOM is given below:

1. Each node's weights are initialized
2. For *N* training data
    a. An input vector from the training set is chosen and presented to the lattice.
    b. Determine the radius of the neighbourhood of the BMU.

c. Each neighbouring node's weights are adjusted to make them more like input vector. The closer a node is to Best Matching Unit (BMU), the more of its weights get altered.

3. Training complete, we have m x n well trained SOM lattice map.

Here, n and m were both set to 20.

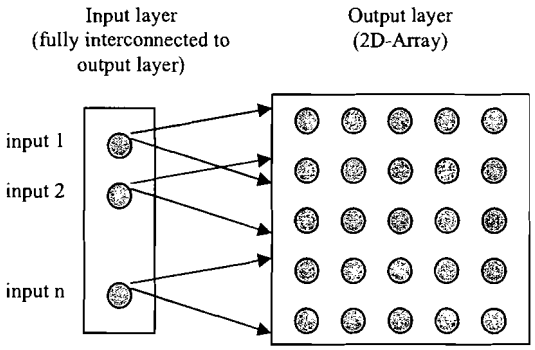

Figure 3: Kohonen SOM

i) Best Matching Unit

Euclidean Distance is used to find BMU. The formula is given below:

$$\text{Distance} = \sqrt{\sum_{i=0}^{i=n}(V_i - W_i)^2} \tag{3}$$

The node which resembles the input vector is chosen to be the BMU.

ii) Radius of the neighbourhood (decay function)

A unique feature of the Kohonen learning is that the area of the neighbourhood shrinks over time. To accomplish that, we used the exponential decay function described below:

$$\sigma(t) = \sigma_0 \exp\left(\frac{-t}{\lambda}\right) \text{ where } t = 0,1,2,3..n \tag{4}$$

t denotes the iteration. It goes from first iteration to n^{th}. iteration and λ is a time constant. Meanwhile, σ_0 is the width of lattice at t=0. Here, our $\sigma_0 = 10$.

iii) Weight adjustments on the neighbouring nodes

Every node resides in the BMU's neighbourhood will have its weight adjusted according to the following equation.

$$W(t+1) = W(t)\theta(t)L(t)(V(t) - W(t)) \tag{5}$$

Where θ represents the amount of influence a node's distance from BMU has on its learning and $\theta(t)$ is given in the equation as follows:

$$\theta(t) = \exp\left(\frac{-dist^2}{2\sigma^2(t)}\right) \quad t=1,2,3,\ldots n \tag{6}$$

Where *dist* is the distance a node is from BMU and σ is the width of the neighbourhood calculated using Equation (2).

Here, the lattice size used was 20x20. The maximum number of training iteration depends on the amount of training data. In this study the amount of training data were 1600 for U2R, Probe and R2L and 2400 for DoS.

6. EXPERIMENT

The experimental flow conducted in this study is summarized in Figure 4 below.

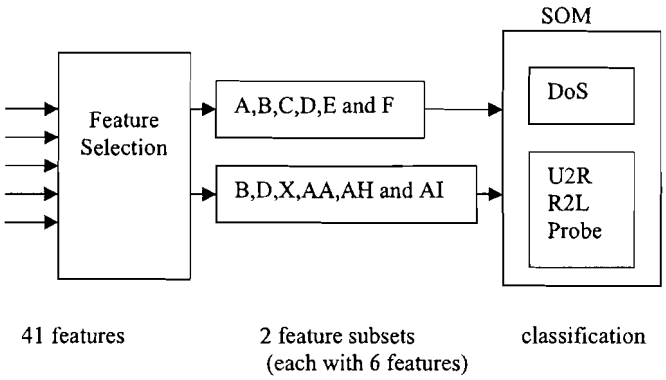

Figure 4: Experimental Flow

The experiment can be partitioned into 2 parts. First part deals with finding the significant features to represent each class of the attacks. The second part deals with Kohonen SOM classification where quantization error is used to measure similarities.

i) Phase I (to obtain significant generic feature subset)

The flow of the Phase I experiment is given in Figure 2. Phase I has produced a feature subset consisting of: *B, D, X, AA, AH* and *AI*. Details of the work and results can be found in [25]. From here onwards, we refer these features as generic feature subset.

ii) Phase II (to do classification)

The purpose of Phase II experiment was to fine-tune and further test the capability of the generic feature subset on class specific attacks. Generic feature subset was used to classify all the attacks. DoS was also trained and tested using the following features: *A, B, C, D, E* and *F*

known as basic features [26] since its' result using generic feature subset was poor. This is discussed later in this section. The classification was done using Kohonen SOM and the quantization error was used as a yardstick to measure the similarities between the input data and the clusters formed as a result of training. By doing this, we also avoid the need to set a threshold value. Quantization error and similarity are inversely proportionate. The least error indicates high resemblance.

20x20 SOM_Probe was trained with 1600 data from Probe category. 400 data were each randomly selected from Ipsweep, Nmap, Portsweep and Satan. For smaller data, we replicated the data to reach 400. Similarly, 20x20 SOM_U2R was trained with 1600 data from U2R category. 400 data were each from Buffer_overflow, Loadmodule, Perl and Rootkit. Since the sizes of these attacks were small, each of the attacks was replicated once. 20x20 SOM_R2L was also trained with 1600 data from R2L category. Since there were eight known attacks which fall under this category, we took 200 data from each attack. We replicated the data few times since their sizes were extremely small (except for Warezclient, its size was big). As for 20x20 SOM_DoS, the size of training data was 2400 where 400 data were randomly taken from Back, Neptune, Smurf and Teardrop. As for Land and Pod, we replicated the data to make their amounts equivalent to other attacks. Our test dataset contained 392,971 data and its' distribution can be found in Figure 1.

7. RESULTS AND DISCUSSION

Results are discussed in two segments; generic feature subset obtained from Phase I and classification rates for all attack categories in Phase II.

i) Phase I

Phase I has proposed a generic feature subset (*B,D,X,AA,AH* and *AI*). This feature subset was compared to the features obtained by Sung and Mukkamala [13] for validation. Phase I of our work only involved two class classification; Normal and Attack, using SVM classifier. Feature subset produced by our approach produced the highest mean (93.408%) and the least standard deviation (1.989). Least standard deviation indicates that the feature subset obtained by Rough-DPSO is robust. Its performance was quite consistent for multiple sets of data tested. Further information on results from Phase I can be found in [24].

ii) Phase II

Result from Phase II was on classification of specific category of attacks. Results showed that R2L, Probe and UR2 were best described using the generic feature subset (*B,D,X,AA,AH* and *AI*) whereas DoS is best represented using six basic features (A,B,C,D,E and F). The results of their classifications are depicted in Figure 5 to 8.

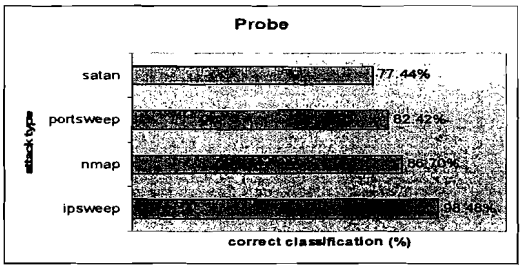

Figure 5: Classification for Probe

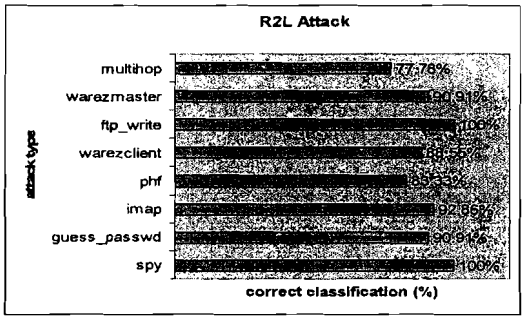

Figure 6: Classification for R2L

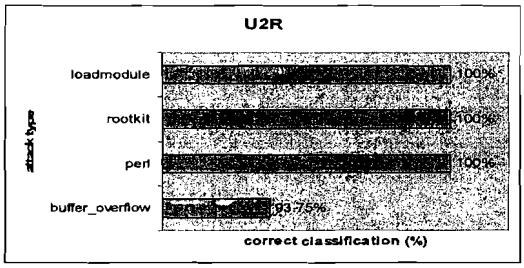

Figure 7: Classification for U2R

We found out that DoS was not suitable to be represented by the generic feature subset. Thus, more feature subsets were explored. Below is the comparison of two feature subsets used to classify DoS.

Table 2: Classification of Dos Using Kohonen SOM

Attacks	B,D,X,AA,AH and AI (generic feature subset)	A,B,C,D,E and F (basic feature subset)
Teardrop	63.27	100
Smurf	99.92	100
Pod	50.00	97.74
Neptune	80.91	99.85
Land	95.6	100
Back	0.32	91.90
Mean	65.00	**98.27**

It is obvious that the result based on the generic feature (1st column) is not consistent unlike when basic features were used to describe DoS. *Smurf* and *Land* show good performance because their patterns are consistent in most of the features thus made them easy to be recognized.

Since basic feature subset gave a superior performance when compared to generic feature subset, we have used the former subset to classify DoS. The classification result using the basic features (*A,B,C,D,E* and *F*) is shown in Figure 8 below.

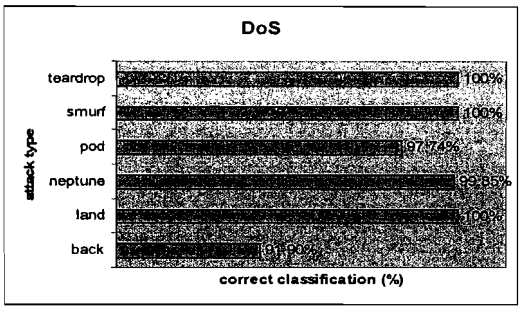

Figure 8: Classification of attacks for DoS

Table 3 below summarizes two class specific features that best used to characterize all the four types of attacks. Probe, R2L and U2R share common generic feature subset while DoS has its' own feature subset with only two of the latter features (B and D) overlapped with the features from the generic feature subset. We also found that all the attack categories can be sufficiently represented by only six significant features as claimed by Sung and Mukkamala [13].

Probe, R2L and U2R are content-based attack and can be detected indirectly especially when derived features are used together with some basic features. Meanwhile, DoS is best described with six basic features that describe the connection itself.

Table 3: Two class-specific Features

Class	Features Selected	Meaning
Probe (6)	protocol_type(B), flag(D), srv_count(X), rerror_rate(AA), dst_host_same_srv_rate(AH), dst_host_diff_srv_rate(AI)	**B**: protocol specific attack **D**: connection status **AA**: connections that have REJ errors **X,AH,AI**: related to multi target attacks
R2L (6)	- same -	
U2R (6)	- same -	
DoS (6)	duration(A), protocol_type(B), service(C), flag(D), src_byte(E), dst_byte(F)	**A**: continued duration, short interval **B,C**: indicates nature of attack which is protocol-service specific **D**: connection status **E,F**: contain little to no data both at source and destination.

The classification rate for each of the category is shown in the table below. Comparison was done with the results on class-specific model reported by [23] for known attacks. They also used KDD Cup 1999 data but their test set was smaller than ours. Theirs was 6,039 while ours was 392,971 (65 times larger) covering all the attacks with all sorts of patterns available in the KDD Cup 1999 labeled data. By taking large sample, random variation is better treated as such the result of this study should be reliable for comparison.

Table 4: Detection performance

Attack Category	Classification (%)	
	Our approach	Work of [23]
R2L	90.54	38.7
Probe	86.26	99.6
U2R	98.44	64.1
DoS	98.27	98.4
Mean	**93.38**	75.2

From Table 4 above, our two class specific features have a higher mean which indicates superior performance when compared to the class specific model proposed by [23]. Significant difference can be observed in R2L category. Our generic feature subset was 50% more superior that their features (B,C,D,J,S,W,AD,AF and AM). This is most probably due to inappropriate features were selected in their R2L model.

8. CONCLUSION AND FUTURE WORK

It is inherent that specific attack types can be represented by different features. Here, we have shown that minimal significant features (6) are sufficient to classify the attacks. Probe, U2R and R2L can be represented by the following features: *B, D, X, AA, AI* and *AI*. Whereas DoS is best represented by the following features: *A, B, C, D, E* and *F* known as basic features. The difference in features is due to unique characteristics of each attack. For example, DoS can be associated with multiple connections in a short time frame as opposed to other types of attacks.

Feature Selection and Classification discussed in this paper are two important components in our Adaptive IDS Model. Feature Selection is crucial in obtaining a fast and accurate detection. Whereas classification by the trained 20x20 SOM for all the attack categories will later become the baseline comparison which is the hard core of our adaptive IDS model. Adaptability will be based on quantization error. Smaller quantization error will indicate a higher degree of similarity between the input data to the particular cluster in the SOM lattice map. The future work will concentrate on unknown attacks and extensive testing will be done using our baseline models developed at this phase II of our study.

REFERENCES

[1] Sung, A. H. and Mukkamala, S. (2003): "Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks." *Proceedings of the 2003 Symposium on Applications and the Internet (SAINT'03).* pp. 209-216.

[2] Li, J.; Zhang, G.Y and Gu, G.C. (2004):"The Research and Implementation of Intelligent Intrusion Detection System Based on Artificial Neural Network." *IEEE Proceedings of the 3rd. International Conference on Machine Learning and Cybernetics.* pp. 3178-3182.

[3] Liu, G.; Yi, Z. and Yang, S. (2007):"A Hierarchical Intrusion Detection Model Based on the PCA Neural Networks." *Journal of Neurocomputing,* Vol. 70, pp. 1561-1568, 7-9 Mar.

[4] Zhang, C.; Jiang, J. and Kamel, M. (2005) "Intrusion Detection using Hierarchical Neural Networks." *Pattern Recognition Letters* Vol. 26 pp. 779-791.

[5] Xu, X. and Wang, X. (2005) "An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines." *Proceedings of First International Conference on Advanced Data Mining and Applications ADMA*, Wuhan, China, Volume 3584. pp 696-703.

[6] Gao, H.; Yang, H. and Wang, X. (2005): "Kernel PCA Based Network Intrusion Feature Extraction and Detection Using SVM." *Lecture Notes in Computer Science,* Vol. 3611. Springer-Verlag, Berlin Heidelberg New York. pp. 89-94.

[7] Kim, J. W. (2002): "Integrating Artificial Immune Algorithms for Intrusion Detection." PhD Thesis, department of Computer Science, University College of London.

[8] Hofmeyr, S. A. (1999): "An Immunological Model of Distributed Detection and Its Application to Computer Security", PhD Thesis, Computer Scienec Dept of University of New Mexico, United States.

[9] Bojanic I. (2005): "Online Adaptive IDS Scheme for Decetiong Unknown Network Attacks Using HMM Models." MSc Thesis, Department of Electrical and Computer Engineering, University of Maryland, United States.

[10] Zhong, A. and Jia, C. (2004): "Study on the Applications of Hidden Markov Models to Computer Intrusion Detection." *IEEE Proceedings of the 15th World Congress on the Intelligent Controls and Automation*, Hangzhou, China. pp. 4352-4256.

[11] Gao, B.H.; Ma, Y. and Yang, Y.H. (2002): "HMM (Hidden Markov Models) Based on Anomaly Intrusion Detection Method." *IEEE Proceedings of the 1st. International Conference on Machine Learning and Cybernetics*. pp. 381-385.

[12] Ye, N.; Chen, Q. and Borror, C.M. (2004): "EWMA Forecast of Normal System Activity for Computer Intrusion Detection." *IEEE Transactions on Reliability*. Vol. 43(4). pp. 557-566.

[13] Sung, A. H. and Mukkamala, S. "The Feature Selection and Intrusion Detection Problems." *Lecture Notes in Computer Science,* Vol. 3321. Springer-Verlag, Berlin Heidelberg New York pp. 468-482. 2004

[14] Chebrolu, S.; Abraham, A. and Thomas, JP. (2005): "Feature Deduction and Ensemble Design of Intrusion Detection Systems. *Journal of Computers and Security.* Vol 24, Issue 4 pp. 295-307.

[15] Chen,Y.; Abraham A. and Yang, J.(2005): "Feature Selection and Intrusion Detection Using Hybrid Flexible Neural Tree. *Lecture Notes in Computer Science*, Vol. 3498. Springer-Verlag, Berlin Heidelberg New York . pp. 439-444.

[16] Chen, Y.; Abraham, A. and Yang, J. (2006): "Feature Selection and Classification Using Hybrid Flexible Neural Tree." *Journal of Neurocomputing* Vol 7 pp. 305-313. 2006

[17] Shi Y. (2004): "Particle Swarm Optimization. Featured Article, IEEE Neural Networks Society. Pp.8-12.

[18] Wang, K.; Huang, L.; Zhou, C. and Pang, W. (2003): "Particle Swarm Optimization for Traveling Salesman Problem." *Proceedings of the Second International Conference on Machine Learning and Cybernetics,* Xi'an. 2-5 November 2003. pp. 1583-1585.

[19] Kennedy, J. and Spears, W.M. (1998): "Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal

Problem Generator." *Proceedings of International Conference on Evolutionary Computation.* pp. 78-83.

[20] Jensen, R. and Shen, Q.(2003): "Finding rough set Reducts with Ant Colony Optimization." *Proceedings 2003 UK Workshop on Computational Intelligence.* Pp. 15-22.

[21] Jensen, R. and Shen, Q. (2005): "Fuzzy-rough Data Reduction with Ant Colony Optimization." *Journal of Fussy Sets and Systems* vol. 149. pp. 5-20.

[22] Zhang, L. H.; Zhang, G. H.; Yu, L.; Zhang, J., and Bai, Y.C.(2004): "Intrusion Detection Using Rough Set Classification." *Journal of Zheijiang University Science.* Vol. 5(9). pp. 1076-1086. 2004.

[23] Sung, W. Shin and Chi H. Lee (2006): "Using Attack-Specific Feature Subsets for Network Intrusion Detection." Springer Verlag LNAI 4304, pp. 305-311.

[24] Anazida Zainal; Mohd Aizaini Maarof and Siti Mariyam Shamsuddin (2007). "Finding Granular Features Using Rough-PSO in IDS." 5th International Conference on Information Technology in Asia 2007 (*CITA'07), Kuching Sarawak, Malaysia. July 10 -12th.* pp. 51-57.

[25] Anazida Zainal; Mohd Aizaini Maarof and Siti Mariyam Shamsuddin (2007): "Feature Selection Using Rough-DPSO in Anomaly Intrusion Detection." ***To appear*** *in International Conference on Computational Science and Its Applications (ICCSA 2007).Kuala Lumpur, Malaysia. August 26-29.*

[26] Kayacik, H. Gunes; Zincir-Heywood, A. N. and Heywood, M. I. (2007): "A hierarchical SOM-based intrusion detection system." *Journal of Engineering Applications of Artificial Intelligence* Vol. 20 (2007). pp. 439-451.