Finding Granular Features using Rough-PSO in IDS

ANAZIDA ZAINAL, MOHD AIZAINI MAAROF and SITI MARIYAM SHAMSUDDIN
Faculty of Computer Science and Information Technology
Universiti Teknologi Malaysia
81310 Skudai, Johor
MALAYSIA
anazida, aizaini and mariyam@utm.my

Abstract *- Most of the existing IDS use all the features in network traffic to evaluate and look for known intrusive patterns. Unfortunately, such system suffers a lengthy detection procedure. Serious implication may incur to a host computer or network due to delay in diagnosis. Feature reduction improves the speed of data manipulation and classification rate by reducing the influence of noise. Besides, selecting important features from input data leads to a simplification of a problem, faster and more accurate detection rates. The purpose of this paper is to investigate the effectiveness of the Rough Set and Particle Swarm (PSO) in feature selection. Support Vector Machine (SVM) was used as a classifier. Data used in this experiment was originally obtained from dataset created by DARPA in the framework of the 1998 Intrusion Detection Evaluation Program. Six significant features were proposed by Rough-PSO.*

Keywords: Feature selection, intrusion detection, Rough Set, PSO, significant features.

1 Introduction

Generally, research in intrusion detection is aimed at getting a high classification rate. In pursuing high accuracy, most of the reported works fail to address the urgency of such detection. They use all the existing features in the network traffic to match against the known intrusive patterns. This has resulted in a lengthy detection procedure. It is supported by literature which shows that most of the reported works put great emphasis on producing a good classifier that can do accurate detection rather than concentrating on feature selection and feature reduction issue. Various techniques including machine learning and statistical approaches have been implemented and their accuracy are satisfactory. Among them are Artificial Neural Network [1-3], Support Vector Machine (SVM)[1][4-5], Bayesian Network and few others. Recent publications have shown that more researches in IDS have deployed SVM. This is due to its generalization ability and the absence of local minimal and sparse representation of solution [5].

Meanwhile, research in finding best feature subset has been intensified in early 2000. Both statistical and machine learning approaches are popularly used. [6] have used Bayesian Network and Classification and Regression Tree, [7-8] used Flexible Neural Tree and few others have used other types of machine learning techniques.

Particle Swarm Optimization (PSO) is a population-based search algorithm and initialized with a population of particles having a random solution. Each particle in PSO is associated with a velocity [9]. Particles's velocities are adjusted according to historical behavior of each particle and its neighbors while they fly through the search space. The particle swarms find an optimal region of complex search spaces through the interaction of individual in a population of particles. PSO has been successfully applied to a large number of optimization problems such as traveling salesman problem (NP-hard) [10]. Literature also pointed out that the binary version of PSO is often outperformed Genetic Algorithm [11]. With proper adaptation and data representation, PSO can be used to find an optimal feature subset.

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 [12]. According to Pawlak [13], it can be used to find out all possible feature subsets.

The objective of this paper is to propose a minimal set of features for IDS using the 2-tier process; Rough Set and Particle Swarm Optimization. The rest of this paper is organized as follows: Section 2 discusses feature selection, Section 3 describes the three techniques adopted in the study. It gives basic description on Rough Set followed by PSO and its implementation in feature selection problem. SVM is used as classifier and lightly touched at the end of Section 3. Section 4 discusses on experiments and results, finally Section 5 concludes the paper and gives the direction of the next stage of this research.

2 Feature Selection

Feature selection is where a feature subset is selected to represent the data. The significance of feature selection

can be viewed in two facets. First is to filter out noise and remove redundant and irrelevant features. According to [14], feature selection is compulsory due to the abundance of noisy, irrelevant or misleading features in a dataset. Second, feature selection can be implemented as an optimization procedure of search for an optimal subset of features that better satisfy a desired measure [15]. Generally, the capability of an anomaly intrusion detection is often hinders by inability to accurately classify variation of normal behaviour as an intrusion. Additionally, network traffic data is usually huge and according to Sung and Mukkamala [16], one of the main problems with IDSs is the overhead of which can be prohibitively high. Generally, an intrusive behaviour has some patterns or structures or relationship properties that are unique and recognizable. These common properties are often hidden within the irrelevant features and some features contain false correlation [6]. Some of these features may be redundant [17] and may have different discriminative power. Thus, this problem can be addressed as a pattern recognition problem to disclose the hidden significant features from the irrelevant features and later ease the classification task in terms of accuracy and speed. According to [18], the existence of these irrelevant and redundant features generally affects the performance of machine learning or pattern classification algorithms.

A conceptual diagram for feature selection that is often used in pattern recognition and classification is shown in Figure 1. It begins with transformation of n-dimensional observation space (example: initial features consist of 41 features of a network connection) represented by **P**, into a q-dimensional vector (selected features from initial features) represented by f. This transformation has reduced the amount of features need to be analyzed for recognition and classification purposes ($\mathbf{P}>q$). f is then mapped into **m** possible distinguishable classes in the decision space for classification purpose.

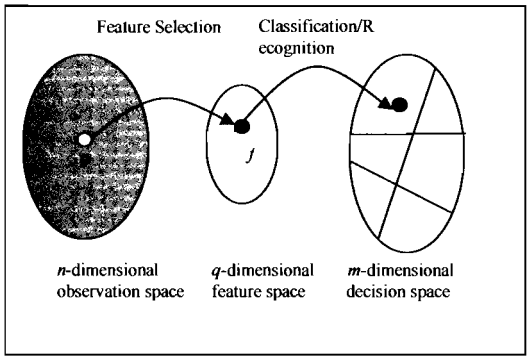

Figure 1. Conceptual diagram for pattern classification/recognition [19]

Figure 2 shows the flow of the feature selection procedure that was adopted in this study. It's structure composes of 2-tier procedure and it has three important phases. The two tiers are; coarse and granular. Coarse Feature Selection tier deploys Rough Set Theory (RST) to filter out the redundant and irrelevant features. Meanwhile, the Granular Feature Selection tier constitutes the deployment of Particle Swarm to further refine the filtration and recommend only the significant features that can classify the data in the dataset.

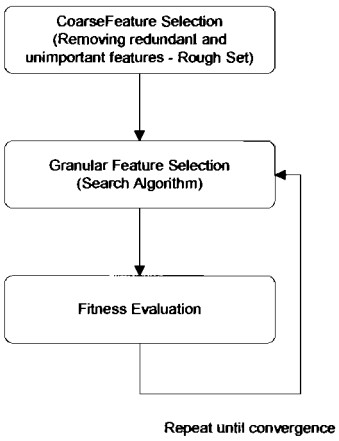

Figure 2. Intrusion Detection Feature Selection Procedure

3.0 Techniques used in the Study

2-tier structure as described in Figure 2, involves two techniques. Rough Set acts as a first filter, followed by PSO to further refine the filtration and finally this feature subset will be classified by one-class SVM. The fitness of the proposed feature subset is evaluated using a fitness function described in Section 3.3.

3.1 Rough Set

Rough set theory (RST) has been used successfully used as a selection tool to discover data dependencies and reduce the number of attributes contained in a dataset by purely structural method [12]. According to [13], it can be used to find out all possible feature subsets.

The main contribution of rough set theory is the concept or reducts. A reduct is a minimal subset of attributes with the same capability of objects classification as the whole set of attributes. Reduct computation of rough set corresponds to feature ranking for IDS. Below is the derivation of how reducts are obtained.

Definition 1 An information system is defined as a four-tuple as follows, $S=<U, Q, V, f>$, where $U=\{x_1, x_2, \ldots, x_n\}$ is a finite set of objects (n is the number of objects); Q is a finite set of attributes, $Q=\{q1, q2, \ldots, qn\}$; $V=\bigcup_{q\in Q}V_q$ and V_q is a domain of attribute q; f:U×Q→V is a total function such that $f(x, q) \in V_q$ for each $q \in Q$, $x \in U$. If the attributes in S can be divided into condition attribute set C and decision attribute set D, i.e. $Q=C \cup D$

and $C \cap D = \Phi$, the information system S is called a decision system or decision table.

Definition 2 Let $IND(P)$, $IND(Q)$ be indiscernible relations determined by attribute sets P, Q, the P positive region of Q, denoted $POS_{IND(P)}(IND(Q))$ is defined as follows:
$POS_{IND(P)}(IND(Q)) = \cup_{X \in U\, IND(Q)} / IND(P) - (X)$.

Definition 3 Let P, Q, R be an attribute set, we say R is a reduct of P relative to Q if and only if the following conditions are satisfied:
(1) $POS_{IND(R)}(IND(Q)) = POS_{IND(P)}(IND(Q))$;
(2) For every $r \in R$ follows that
$POS_{IND(R-\{r\})}(IND(Q)) \neq POS_{IND(R)}(IND(Q))$

Further details can be found in Pawlak [13]. According to Zhang et al. [20], this method produces explainable detection rules and it also has high detection rate for some attacks.

3.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based search algorithm and initialized with a population of particles having a random position (solution). Each particle is associated with velocity. Particles' velocities are adjusted according to historical behaviour of each particle and its neighbours while they fly through search space [15]. Thus, particles have a tendency to fly towards the better and better search area over the course of search process [9]. The calculation of velocity is described as below:

$$V_{id} = wV_{id} + C_1 rand()(P_{id} - X_{id}) + C_2 Rand()(P_{gd} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

Where C_1 and C_2 are positive constants called learning rates. These represent the weighting of the stochastic acceleration terms that pull each particle towards its' *pbest* and *gbest* positions. Low values allow particles to roam far from target regions before being tugged back, while high values result in abrupt movement toward, or past target regions.
rand ( ) and *Rand ( )* are two random functions in the range [0,1] and w is the inertia weight. Suitable selection of the inertia weight provides a balance between global and local exploration, and results in less iteration on average to find a sufficiently optimal solution.
$X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ represents the i^{th} particle and $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ represents the best previous position of the i^{th} particle.
$V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ represents the rate of the position change (velocity) for particle i.

Formula (1) and (2) gives PSO the following capabilities:

1. Memory of the flying particles is given in the first part of the formula.
2. Cognition, which represents the private thinking of the particle, is given in the second part of the formula.
3. Social, represents the collaboration among the particles.

1, 2 and 3 are used to calculate the particle's new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group's best experience [23]. Then the particle flies toward a new position according to equation (2).

3.3 PSO Implementation in Feature Selection

The original PSO is designed for real value problems. Now, the algorithms have been extended to tackle discrete problems. A term 'binary PSO' appeared when PSO is used to solve discrete problem. Various researchers have implemented PSO in feature selection and their applications are diverse. For example, [21] used PSO to select feature subset for classification task and to train RBF neural network simultaneously, [24] used it to diagnose fault in chemical process and [15] implemented PSO to extract features of hyperspectral data for under spilled blood visualization.

Binary PSO also utilizes the formula given in (1) and (2). Generally for feature representation, 1 bit of a particle represents 1 feature. If the feature is selected, the bit is set to 1 and 0 otherwise. Few approaches were used to select features for a particle. Some researches use roulette wheel selection to select features [15] and some randomly select these features [23]. Some reported works implemented selection pressure to control the probability of selecting highly fit features [15]. [25] used velocity as a probability to determine whether X_{id} (a bit) will be in 1 state or 0 state and they used sigmoid function $s(v)=1/(1+\exp(-v))$ to squashed V_{id}. A suitable fitness function will be deployed to evaluate the feature subset proposed.

Apart from feature representation, [23] has proposed the following mechanism for the velocity representation. When particle P is compared to its *lbest* and the *gbest*, sum of -1 and +1 is added. -1 penalty is given when the i^{th} feature in P is chosen but not in *lbest*, and penalty -1 also been given when *gbest* does not contain the feature. +1 is given when *lbest* does have the feature and P does not. Similar procedure goes when comparing between *gbest* and P. Detail procedure of location updating strategy can be found in [23].

Below is the PSO pseudo-code used in this study:

1. Initialize all the possible positions (represent all possible feature subset bands). If the feature is N, thus, there are 2^N possible feature subsets.
2. Introduce m particles, where each will randomly take one position in the feature subset space.

3. Initialize their P_{lbest} for all particles. 1st round, their P_{lbest} = current position.
4. Find their G_{best}
5. Loop (exit when fitness > max_fitness) or (reach max_iter)
    a. Evaluate fitness of each particle's position. Choose the P_{gbest}.
    b. For each particle, check the following :
        i. If $P_{curr} > P_{lbest}$ then $P_{lbest} = P_{curr}$
    c. For each P_{lbest} check the following ;
        i. If $P_{lbest} > P_{gbest}$ then $P_{gbest} = P_{lbest}$
    d. Update velocity for each particle with respect according to formula in (1).
    e. Update the position for each particle according to formula in (2).
6. End.

Here, our N is 15 and the value of M is 5. The iteration of the above pseudo code will continue except when either one of the stopping criteria is met; (i) maximum number of iterations or (ii) the fitness of the proposed feature subset has exceeded the fitness value being set. In most of the feature selection work, a fitness function is normally defined as the correct classification rate using the features picked by each particle. We have adopted the following fitness function in our experiment. The same fitness function was used in [23].

$$\alpha^* \gamma_R(D) + \beta^* \frac{|C| - |R|}{|C|} \quad (3)$$

Where $\gamma_R(D)$ is the classification rate for attribute set R relative to decision D.
|R| is the '1' number of position or the length of selected feature subset. |C| is the total number of features.
α and β are two parameters corresponding to the importance of classification quality and subset length. $\alpha \in [0,1]$ and $\beta = (1- \alpha)$. The classification quality is more important than subset length. The goodness of each position of a particle is measured by this fitness function.

3.4 Support Vector Machine

Support Vector Machine (SVM) is a learning method based on the Structural Risk Minimization principle from statistical learning theory. The principle idea of an SVM is to separate classes with a surface that maximizes the margins between them. It is a powerful classification learning approach which applies the following concept; non-linear input vectors are mapped through a very high dimension feature space where the linear decision of the input vectors is computed in this feature space. By dividing the high-dimensional space into different boundaries or subspaces, SVM maximizes the classification according to the generalized boundary.

[26] performed testing for intrusion detection accuracy on several techniques and claimed that SVM outperformed MARS and ANN, with respect to scalability (SVM can train larger number of patterns while ANN fails to converge) and prediction accuracy. In fact, SVM performed well among the classical intrusion detection algorithms [27]. A few researches used multiclass SVMs [28-29]. SVM is claimed to outperform most of other algorithms [30]. One remarkable property of SVM is its ability to learn can be independent of the feature space dimensionality which means SVM can generalize well in the presence of many features [31]. Here, we used *libsvm* [32] as a classifier.

4.0 Experiment and Results

The original data was obtained from database created by DARPA in the framework of the 1998 Intrusion Detection Evaluation Program (http://www.11.mit.edu/IST/ideval). The raw training data was about 4GB of compressed binary TCP dump data from seven weeks of network traffic. Here we used the KDDCup 1999 data subset that was pre-processed by the Columbia University and distributed as part of the UCI KDD Archive (http://kdd.ics.uci.edu/databases/kddcup1999/kddcup1999.html). Attacks fall into four main categories:

1. DOS - denial-of-service, example syn flood.
2. R2L - unauthorized access from a remote machine, example guessing password.
3. U2R - unauthorized access to local superuser (root) privileges, example various ``buffer overflow'' attacks.
4. Probing - surveillance and other probing, example, port scanning.

For each TCP/IP connection, 41 various quantitative and qualitative features were extracted plus 1 class label. Table 1 shows all the features found in a connection. For easier referencing, each feature is assigned a label (A to AO). This referencing is adopted from [6]. Some of these features are derived features. These features are either nominal or numeric.

Table 1- Network data feature label

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

[16] used three different techniques in ranking the significant 6 features in intrusion detection. The techniques used were Support Vector Machine (SVM), Linear Genetic Programming (LGP) and Multivariate Adaptive Regression Splines (MARS). Each of the techniques produced different feature subsets with few features overlapped. The diagram below depicts the relation between the features chosen by the techniques.

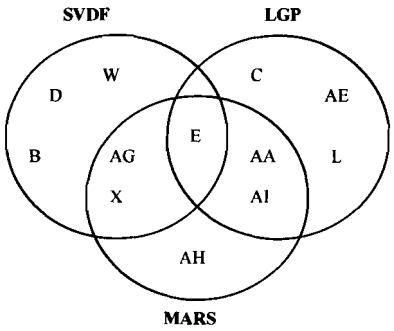

Figure 3. Six significant features from SVDF, MARS and LGP

4.1 Experiment Setup

We used 4 sets of data in which each set contained 4000 records. In all the datasets, 50% to 55% records contained normal data and the remaining were attacks. The first set was the training set and another three sets were used for testing. Test datasets were called dataset_1, dataset_2 and dataset_3. The attack types and their categories are listed in Table 2 below.

Table 2. Attacks and their categories

Category of attacks	Types of attacks
Probe	ipsweep, nmap, portsweep and satan
Denial of Service (DoS)	back, land, Neptune, pod, smurf and teardrop
User to Root (U2R)	buffer_overflow, loadmodule, perl and rootkit
Remote to Local (R2L)	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient and warezmaster

Training dataset was discretized before it was fed to Rough Set tool called Rosetta. Details on Rosetta can be found in [22]. We used Genetic Algorithm to find the reducts. Based on the rules generated, we picked the 15 features that appeared the most in the rules. These features were B, C, D, E, F, L, W, X, AA, AE, AF, AG, AH, AI, and AJ. This feature subset is referred as initial feature subset. This stage is important because Rough Set will eliminate unimportant and redundant features (please refer to Figure 1). This initial feature subset becomes input to the next stage, PSO.

In the second stage particle swarm needs to find an optimal region for complex search spaces. Instead of having all the 41 available features which would have produced 2^{41} possible feature subsets in the search spaces, PSO would now only need to examine 15 features consisting of 2^{15} solution candidates. As described earlier, these 15 features were previously suggested by Rough Set. The reason for having Rough Set filtering at the first stage is to reduce the number of iterations that PSO has to perform in finding an optimum feature subset. We used SVM classifier (libsvm) to classify the data and a fitness function to evaluate the feature subset proposed by Rough-PSO. Based on the pseudo-code given in Section 3.3, Rough-PSO found an optimum solution at the 9^{th} iteration. And the features were: B,D, X, AA, AH, and AI.

4.2 Results and Discussion

As described earlier, [16] had suggested 3 feature subsets based on three techniques. We have used the features proposed by them and trained each of them using our training set. SVM classifier was trained based on each feature subsets and their results were then compared. Here, the detection could either be attack or normal. Table-3 shows their classification rates. The last row is the approach that we have adopted in this study and its' proposed feature subset.

Table 3. Comparison of classification rates for four techniques

Technique	Data1	Data2	Data3	Mean	Std Dev
SVDF (B,D,E,W,X & AG)	89.000	91.275	85.875	88.717	2.214
LGP (C,E,L,AA,AE & AI)	87.775	94.050	96.575	92.800	3.700
MARS (E,X,AA,AG, AH & AI)	79.600	93.300	90.225	87.708	5.869
Rough-PSO (B,D,X,AA, AH & AI)	90.675	95.350	94.200	93.408	1.989

The last two columns show the value of *mean* and *standard deviation* for each of the techniques. Mean gives the average performance of the feature subset proposed by the respective technique on three different test sets. Meanwhile standard deviation is a statistical measure of variance from the mean, representing the dispersion of data (distance) from the mean. Standard deviation is a way of expressing how different the numbers are from the average. Smaller value for standard deviation implies that the feature subset is robust. Which means, despite which dataset is used for testing, the

classification rate does not vary much from its' average performance.

For dataset_1 and dataset_2, PSO has superseded the other three techniques. In dataset_3, LGP performs the best compared to the other three techniques, and PSO is second in the ranking. But the difference was quite small (2.375%). Looking at mean values for each technique, PSO has the highest average classification rate. The last column shows that PSO has the least standard deviation. As a conclusion, PSO can display a consistent performance when different datasets are used for testing.

5.0 Conclusion and Future Work

Based on the datasets used for the experiment, the results indicate that the feature subset proposed by Rough-PSO is superior in terms of accuracy and robustness. PSO has displayed a good performance and in general it takes shorter time (less iteration) to find an optimum feature subset when it is paired with Rough Set. This may be due to the nature of PSO that exploits social behaviour which contributes to faster convergence toward optimum solution.

The finding of this optimum feature subset will lead to the second phase of our work which is to incorporate our IDS with the ability to learn and adapt. It is hoped that this adaptive feature will significantly reduce false positive rate.

References

[1] A. H. Sung, and S. Mukkamala. "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. 2003

[2] J. Li, G. Y. Zhang, and G. C. Gu. "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. 2004.

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

[4] X. Xu, and X. Wang. "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. / 2005 July 22-24, 2005.

[5] H. Gao, H. Yang and X. Wang. "Kernel PCA Based Network Intrusion Feature Extraction and Detection Using SVM." *ICNC 2005, LNCS 3611, Springer Verlag.* pp. 89-94. 2005.

[6] S. Chebrolu, A. Abraham and J. P. Thomas. "Feature Deduction and Ensemble Design of Intrusion Detection Systems." *Journal of Computers and Security.* Vol 24, Issue 4. pp. 295-307. June 2005.

[7] Y. Chen, A.Abraham and J. Yang. "Feature Selection and Intrusion Detection Using Hybrid Flexible Neural Tree." *ISNN 2005. LNCS 3498*, pp. 439-444. 2005.

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

[9] Y. Shi. "Particle Swarm Optimization." Feature Article, *IEEE Neural Networks Society.* pp. 8-12. 2004.

[10] K. Wang, L. Huang, C. Zhou and W. Pang. "Particle Swarm Optimization for Traveling Salesman Problem." Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an. 2-5 November 2003.

[11] J. Kennedy and W. M. Spears. "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. 1998.

[12] R. Jensen, and Q. Shen, "Finding rough set Reducts with Ant Colony Optimization." Proceedings 2003 UK Workshop on Computational Intelligence. 2003.

[13] Z. Pawlak. *Rough Sets, Theoretical Apects of Reasoning about Data.* Kluwer Academic Publishers, Boston, MA. 1991.

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

[15] S. T. Monteiro, K. Uto, Y. Kosugi, N. Kobayashi, E. Watanabe and K. Kameyama. "Feature Extraction of Hyperspectral Data for Under Spilled Blood Visualization Using Particle Swarm Optimization." *International Journal of Bioelectromagnetism*, Vol. 7(1), pp. 232-235. 2005.

[16] A. H. Sung and S. Mukkamala. "The Feature Selection and Intrusion Detection Problems". Proceedings of Advances in Computer Science - ASIAN 2004: Higher-Level Decision Making. 9th Asian Computing Science Conference. Vol. 3321. pp. 468-482. 2004.

[17] R. W. Swiniarski and A. Skowron. "Rough set Methods in Feature Selection and Recognition." *Pattern Recognition Letters* 24, pp. 833-849. 2003.

[18] B. Chakraborty. "Feature Subset Selection by Neuro-rough Hybridization." *RSCTC 2000. LNAI 2005*, Pp. 519-526. 2005.

[19] A. Hassan. "On-line Recognition of Developing Control Chart Patterns." PhD Thesis. Universiti Teknologi Malaysia. 2002.

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

[21] Y. Liu, Z. Qin, Z. Xu and X. He. "Feature Selection with Particle Swarms." CIS 2004, LNCS 3314. Springer Verlag, pp. 425-430. 2004.

[22] A. Øhrn. ROSETTA Technical Reference Manual, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. 66 pages. 2000.

[23] X. Wang, J. Yang, X. Teng, W. Xia and R. Jensen. "Feature Selection based on Rough Sets and Particle Swarm Optimization." *Pattern Recognition Letters,* Vol. 28, Issue 4, pp. 459-471, I March 2007.

[24] L. Wang, and J. Yu. "Fault Feature Selection Based on Modified Binary PSO with Mutation and Its Application in Chemical Process Fault Diagnosis." *ICNC 2005, LNCS 3612. Springer Verlag*, pp. 832-840. 2005.

[25] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers: San Francisco, United States. 2001.

[26] S. Mukkamala, A. H. Hung, and A. Abraham. "Intrusion detection using an ensemble of intelligent paradigms." *Journal of Network and Computer Applications.* Vol. 28. pp. 167-182. 2005.

[27] S. Mukkamala, and A. H. Sung. "Feature ranking and Selection for Intrusion detection Systems." Proceedings of International Conference on Information and Knowledge Engineering. Las Vegas, USA. 2002.

[28] H. Lee, J. Song, and D. Park. "Intrusion Detection System based on Multiclass SVM." Proceedings of 10^{th}. International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. Part II. Vol. 3642. pp. 511-519. 2005.

[29] X. Xu, and X. Wang. "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 2005, Wuhan, China, Volume 3584 /. pp. 696-703. July 22-24, 2005.

[30] C. Burges. "A tutorial on Support Vector Machines for Pattern Recognitlon," *Journal of Data Mining and Knowledge Discovery*. Vol. 2. pp. 121-167. 1998.

[31] W. H. Chen, S. H. Hsu and H. P. Shen. "Application of SVM and ANN for Intrusion Detection." *Journal of Computers & Operations Research.* Vol. 32. Pp. 2617-2634. 2005.

[32] C. Chih and J. Chih. "LIBSVM : A library for support vector machines." Tutorial and software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 2001.