A Rough-Fuzzy Hybrid Algorithm for Computer Intrusion Detection

Witcha Chimphlee[1], Abdul Hanan Abdullah[2], Mohd Noor Md Sap[2], Siriporn Chimphlee[1], and Surat Srinoy[1]
[1]Faculty of Science and Technology, Suan Dusit Rajabhat University, Thailand
[2]Faculty of Computer Science and Information Systems, University Technology of Malaysia, Malaysia

Abstract: *In this paper, we propose an intrusion detection method that combines rough sets theory and fuzzy c-means for anomaly detection. The first step consists of attribute selection which is based on rough set theory for each of the 5 classes of intrusions in the Defense Advanced Research Projects Agency (DARPA) data is identified. The next phase is clustering by using fuzzy c-means; we are using rough sets for cleaning and to filtering out redundant, spurious information. Fuzzy c-means allow objects to belong to several clusters simultaneously, with different degrees of membership. Our method is an accurate model for handling complex attack patterns in large networks. We used data set from 1999 Knowledge Discovery and Data mining (KDD) intrusion detection contest. The main goal of this paper is to apply this method to increase the efficiency of a given intrusion detection model and to be able to reduce the data set by looking for overlapping categories and also to filter in the desired ones.*

Keywords: *Network security, intrusion detection system, anomaly detection, rough sets, fuzzy c-means.*

Received December 31, 2005; accepted June 282006

1. Introduction

Networks today face an unprecedented range of threats and vulnerabilities. The risks have never been greater, and the penalties have never been more severe. Computer security is defined as the protection of computing systems against threats to confidentiality, integrity, and availability. Confidentiality (or secrecy) means that information is disclosed only according to policy, integrity means that information is not destroyed or corrupted and that the system performs correctly, availability means that system services are available where they are needed [2]. An Intrusion Detection System (IDS) is a program that analyzes what happens or has happened during an execution has been misused. An intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource. A fundamental problem in intrusion detection is what matrices(s) can be used to objectively evaluate an intrusion detection system in terms of its ability to correctly classify events as normal or intrusion [15].

Intrusion detection system aims at detecting intruders who elude "first line" protection and acts as the "second line of defence" placed inside a protected network, looking for known or potential threats in network traffic and/or audit data recorded by hosts [8, 13]. Intrusion detection techniques can be categorized into misuse detection and anomaly detection.

Misuse detection uses the patterns of well-known attacks or vulnerable spots in the system to identify intrusions. Anomaly detection attempts to determine whether deviations from the established normal usage pattern can be flagged as intrusions [25]. An ideal intrusion detection system is one that has a high attack detection rate along with a 0% false positive rate.

Attack type fall into four main categories:

- DoS: Denial of Service.
- R2L: Unauthorized access from a remote machine.
- U2R: Unauthorized access to local super user (root) privileges.
- Probing: Surveillance and other probing.

Intrusion detection systems are categorized according to the kind of audit source location they analyze. Most intrusion detection systems are classified as either a network-based or a host-based approach to recognize and detect attacks.

A network-based intrusion detection system performs traffic analysis on a local area network. A host-based intrusion detection system places its reference monitor in the kernel/user layer and watches for anomalies in the system call patterns [6].

This paper is organized as follows. In section 2, we brief related works. In the subsequence section, we considered proposing rough set. In section 4, we stated the fuzzy *c*-means. In section 5, we describe the study data and design of the experiment. In section 6, we reported experiments results. Finally, section 7 concludes our work and discusses future research plans.

2. Related Works

This works closely related to our intrusion detection project include identifying unauthorized use, misuse and attacks on information systems is defined as intrusion detection [12]. The most popular way to detect intrusions has been done by using audit data generated by operating systems and by networks. Soft computing techniques are being widely used by the IDS community due to their generalization capabilities that help in detecting know intrusions and unknown intrusions or the attacks that have no previously described patterns [17]. Several researchers proposed data mining techniques to identify key patterns that help in detecting intrusions [23]. Models of the intended behaviour of users and applications and interpret deviations from this 'normal' behaviour as evidence of malicious activity [16].

Feature selection techniques aim at reducing the number of unnecessary features in classification rules. The features will be used to classify unseen instances into different classes based on the value of the classifier [27]. The concepts in rough set theory are used to define the necessity of features. The measures of necessity are calculated by the functions of lower and upper approximation.

3. Rough Sets

Rough Set Theory (RST) has been used successfully as a selection tool to discover data dependencies and reduce the number of attributes contained in a dataset by purely structural methods [9].

Rough sets remove superfluous information by examining attribute dependencies. It deals with inconsistencies, uncertainty and incompleteness by imposing an upper and a lower approximation to set membership. Given a dataset with discretized attribute values, by the use of rough sets it is possible to find a subset (termed a reduct) of the original attributes using rough sets that are the most informative; all other attributes can be removed from the dataset with minimal information loss [18].

Chakraborty G. and Chakraborty B. [5] proposed positive and negative as shown in search Figure 1 two partitions induced by the decision attribute, positive is for "Decision Yes" and negative is for "Decision No". The partitions X5, X6, and X7, X8 induced by condition attributes are clearly positive or negative side of the partition induced by the decision attribute. In addition, partitions X2 and X3 are clear boundary cases, and no decision can be made out of the corresponding partitions. But partition X4 is mostly positive, and partition X1 is mostly negative. Working with different real world data, we have seen that partitions like X1 and X4 are very common.

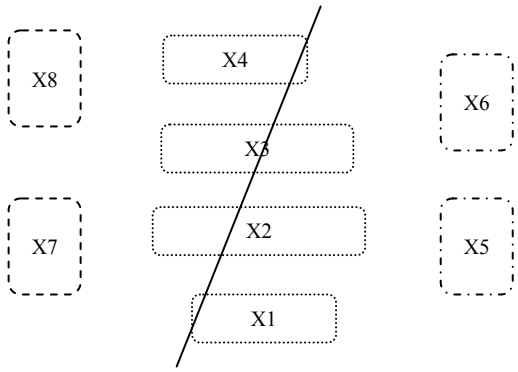

Figure 1. Positive or negative region [5].

The rough sets theory has been developed for knowledge discovery in databases and experimental data sets. An attribute-oriented rough sets technique reduces the computational complexity of learning processes and eliminates the unimportant or irrelevant attributes so that the knowledge discovery in database or in experimental data sets can be efficiently learned.

A rough set is an approximation of a vague concept by a pair of precise concepts, called lower and upper approximations (which are a classification of the domain of interest into disjoint categories).

The classification formally represents knowledge about the problem domain. Objects belonging to the same category characterized by the same attributes (or features) are not distinguishable [19]. Let I = (U, A) be an information system, where U is a non-empty set of finite of objects (the universe). A is a non-empty finite set of attributes such that $a: U \rightarrow V_a$.

Table 1. Decision table for rough set.

Instance	Attributes			Attacks?
	Service	Count	SrvC_ount	
1	http	1	4	Yes
2	ftp_data	2	3	Yes
3	Private	1	5	No
4	http	1	1	Yes
5	Domain_u	2	3	No
6	http	0	2	No

For every $a \in A$; V_a is the value set for attribute a. In a decision system, $A = \{C \cup D\}$ where C is the set of conditional attributes and D is the set of decision attributes. With any $P \subseteq A$ there is an associated equivalence relation *IND*(P):

$$IND(P) = \{(x, y) \in U^2 \mid \forall_a \in P_a(x) = a(y)\} \quad (1)$$

If $(x, y) \in IND(P)$, then x and y are indiscernible by attributes from P. The partition of U, generated by *IND*(P) is denoted U/P and can be calculated as follows:

$$U / P = \otimes\{a \in P : U / IND(\{a\})\}, \text{ where} \quad (2)$$

$$A \otimes B = \{X \cap Y : \forall X \in A, \forall Y \in B, X \cap Y \neq \varnothing\} \quad (3)$$

To illustrate the operation of Rough Set Attribute Reduction (RSAR), an example dataset is presented as in Table 1.

4. Fuzzy C-Means (FCM) Clustering

The fuzzy membership functions corresponding to the informative regions are stored as cases. A collection of fuzzy sets, called fuzzy space, defines the fuzzy linguistic values or fuzzy classes. A sample fuzzy space of five membership function is shown in Figure 2.

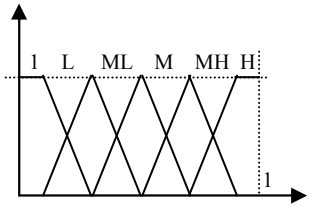

Figure 2. A fuzzy space of five membership function.

The most well-known fuzzy clustering algorithm is fuzzy c-means, a modification by Bezdek J. [4] of original crisp clustering methodology. Bezdek introduced the idea of a *fuzzification* parameter (m) in the range [*1*, *n*], which determines the degree of fuzziness in the clusters. When $m = 1$, the effect is a crisp clustering of points, but when $m > 1$ the degree of fuzziness among points in the decision space increases [10]. The fuzzy c-means algorithm itself is fairly straightforward and involves the basic parameters outlined in Figure 3. Essentially, the parameter m controls the permeability of the cluster horizon.

The fuzzy c-means algorithm uses a fixed number of cluster centers, indicated by the parameter (p).

x: A vector of training data, where *i=1,2,…,n.* These are the clusterattributes selected from the source data elements.

$d_{i,j}$: The distance of the *i*-th data point from the *j*-th cluster center. We use the Euclidean distance.

p: The number of fuzzy clusters specified as part of the algorithm.

m: A fuzzification parameter in the range [*>1, <w*], indicating the width of the *n*-dimensional cluster perimeter. The larger the number the more fuzzy the point assignments into each cluster. Normally *m* is the range [1.25, 2] inclusive.

c_j: The center (or centroid) of a fuzzy cluster (j = 1 ,2 , ..., p). This value is repeatedly calculated by the algorithm (see equation (5)).

$\mu_j(x_i)$: A fuzzy membership qualification indicating the membership of sample x_i to the *j*-th cluster.

Figure 3. Basic parameters of the fuzzy c-means algorithm.

Fuzzy c-means clustering involves two processes: The calculation of cluster centers and the assignment of points to these centers using a form of Euclidean distance. This process is repeated until the cluster centers have stabilized. Fuzzy c-means imposes a direct constraint on the fuzzy membership function associated with each point, as follows [10].

$$\sum_{j=1}^{p} \mu_j(x_i) = 1; \quad i = 1, 2, 3, .., k \tag{4}$$

where:
p: The number of specified clusters.
k: The number of data points.
x_i: The *i*-th data point.
$M_j()$: The function that returns the membership of x_i in the *j*-th cluster.

The goal of the fuzzy c-means algorithm is the assignment of data points into clustered with varying degrees of membership. This membership reflects the degree to which the point is more representative of one cluster than another (or all the others). In effect, we are attempting to minimize a standard loss function, expressed simply as

$$l = \sum_{k=1}^{p} \sum_{i=1}^{n} [\mu_k(x_i)]^m \; ||x_i - c_k||^2 \tag{5}$$

where:
l: The minimized loss value.
n: The number of data points.
x_i: The *i*-th data point
$\mu_k()$: (as in equation (4)) a function that returns the membership of x_i in th *k*-th cluster.
m: the fuzzification parameter.
c_k: The center of the *k*-th cluster.

From this we can derive the two fundamental equations necessary to implement the fuzzy c-means clustering algorithm. Equation (6) is used to calculate a new cluster center value.

$$c_j = \frac{\sum_i [\mu_j(x_i)]^m x_i}{\sum_i [\mu_j(x_i)]^m} \tag{6}$$

where:
c_k: The center of the *j*-th cluster
$\mu_j(\;)$: (as in equation (5)) a function that returns the membership of x_j in th *j*-th cluster
x_i: The *i*-th data point
m: The fuzzification parameter

The second step, determining the cluster membership for a sample point, is only slightly more complicated. We first need to know the distance from a point x_i to each of the cluster centers $c_{1\ldots j}$. This is done, as illustrated in equation (7), by taking Euclidean distance between the point and the cluster center.

$$d_{ji} = ||x_i - c_j||^2 \tag{7}$$

where:
c_j: The center of the *j*-th cluster.
x_i: The *i*-th data point
d_{ji}: The distance of *xi* from the center of cluster c_j.

Since the fuzzy c-means algorithm constrains the total cluster membership for a point to one [1], we calculate

a point's membership as the fractional part of the total possible memberships assigned to the current point. Equation (8) shows how the membership in the *j*-th cluster is calculated.

$$\mu_j(x_i) = \frac{\left(\frac{1}{d_{ji}}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{p}\left(\frac{1}{d_{kj}}\right)^{\frac{1}{m-1}}} \quad (8)$$

where:
$\mu_j()$: The membership of x_i in th j-th cluster.
d_{ji}: The distance of x_i from the center of cluster c_j.
d_{ki}: The distance of x_i from the center of cluster c_k..
m: The fuzzification parameter.
p: The number of specified clusters.

Performance depends on initial centroids. For a robust approach, there are two ways which is described below [3].

1. Using an algorithm to determine all of the centroids. (for example: arithmetic means of all data points).
2. Run FCM several times each starting with different initial centroids.

5. Study Data and Design of the Experiment

The steps in our method as shown in Figure 4 are:

1. Clean data and handle missing and incomplete data.
2. Select the best attribute or feature selection, The first two steps using rough set theory operations were done in ROSETTA [24].
3. Cluster group of data by using fuzzy *c*-means.

The first two step performs the following tasks [7]:

1. Identifies the attributes and their value.
2. Converts categorical to numerical data.
3. Normalizes the data.
4. Performs redundancy check and handle null value.
5. Initializes the necessary parameters such as importance attribute, number of cluster.

We ran our experiments on a system with a 1.5 GHz Pentium IV processor and 512 MB DDR RAM running Windows XP with MATLAB®.

5.1. Data Preparation

The data we used in our experiments originated from MIT's Lincoln Lab. It is developed for intrusion detection system evaluations by DARPA and is considered a benchmark for intrusion detection evaluations [20]. The dataset includes a wide of intrusions together with normal activities simulated in a military network environment that is a common benchmark for evaluation of intrusion detection techniques. The simulated attacks fall in one of four major categories: DoS, R2L, U2R and Probing. Testing data use filename "*corrected.gz*" contains a total of 38 training attack types. It consists of approximately 300,000 data instances, each of which is a vector of extracted feature values from a connection record obtained from the raw network data gathered during the simulated intrusion and is labelled normal or a certain attack type. The distribution of attacks in the KDD Cup dataset is extremely unbalanced. Some attacks are represented with only a few examples, e.g. the *phf* and *ftp_write* attacks, whereas the *smurf* and *neptune* attacks cover millions of records. In general, the distribution of attacks is dominated by probes and *Denial-of-Service (DoS)* attacks; the most interesting and dangerous attacks, such as compromises, are grossly under represented [21].

The data set has 41 attributes for each connection record plus one class label. There are 24 attack types, but we treat all of them as an attack group. A data set of size *N* is processed. The nominal attributes are converted into linear discrete values (integers). After eliminating labels, the data set is described as a matrix X, which has *N* rows and $m = 41$ columns (attributes). There are $md = 8$ discrete-value attributes and $mc = 33$ continuous value attributes. We select a testing data set which contained 18,216 records as shown in Table 2 and more detail in Figure 5.

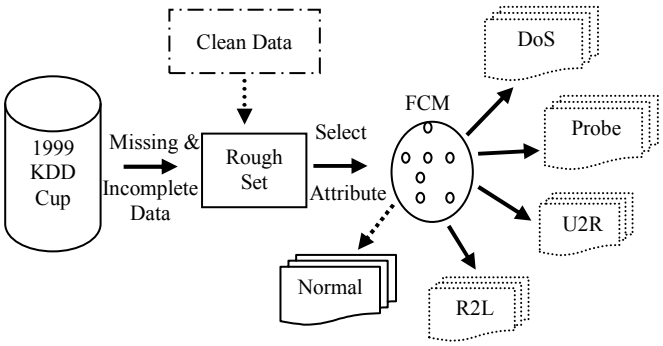

Figure 4. Rough-fuzzy hybrid model.

5.2. Features Selection

Feature selection refers to the problem of selecting those input features that are most predictive of a given outcome. The main aim of feature selection is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features [11]. An IDS must therefore reduce the amount of data to be processed. Data that is not considered useful can be filtered, leaving only the potentially interesting data. Data can be grouped or clustered to reveal hidden patterns. Finally, some data sources can be eliminated using feature selection. Features may contain false correlations, which hinder the process of detecting intrusions. Further, some features may be redundant since the information they add is contained in other features [2].

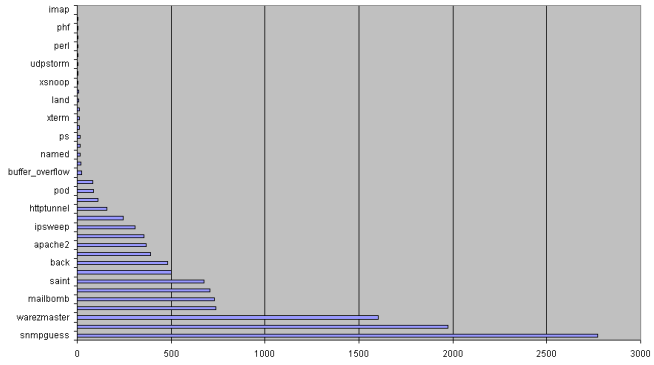

Figure 5. Intrusion detection data distribution for this work.

Feature selection improves classification by searching for the subset of features, which best classifies the training data. In this paper we use rough set techniques for cleaning data and feature selection. The subset of selected features is then used to detect intrusions. To filter out redundant, spurious information, and significantly reduce the number of computer resources, both memory and CPU time, required to detect an attack. Narrowing input data yields the additional benefit of alleviating intrusion detection. Log files are naturally represented as a table, a two dimensional array, where rows stand for objects and columns for their attributes. The process of reducing the rows and columns of a table call object reduction and attribute reduction [14].

Table 2. Dataset for attack distribution.

Attack Types	% Occurrence	Number of Records
Normal	31.64	5,763
Probe	11.88	2,164
DoS	19.38	3,530
U2R	0.38	70
R2L	36.72	6,689
Summary	100%	18,216

5.3. Performance Measure

Two indicators, *detection rate* and *false alarm rate*, were used to measure the accuracy of the method. The detection rate shows the percentage of true intrusions that have been successfully detected. The false alarm rate is defined as the number of normal instances incorrectly labelled as intrusion divided by the total number of normal instances. A good method should provide a high detection rate together with a low false alarm rate [26]. The tools that can be used for evaluating detection model are as follows [1, 22]:

- Confusion Matrices allow the precise measurement of the performance of a model for each modality (events occurrence number, reliability–proportion of the cases with a correct prediction, and precision–proportion of the cases where the true value is correctly predicted);
- Receiver Operating Characteristic (ROC) curves, which are centered on the parameter value of the target node (activity_type).
- Lift charts, which are centered on the parameter value of the target node as well.

6. Results and Discussion

In our experiments, we consider a subset of the KDD data, which consists of 18,216 instances. Except for a few seed points, the labelling information is not utilized in the experiments. Using the important features gives the most remarkable performance in terms of training and testing time. We reduced the dimensionality of this data set (by using rough set and use Johnson's algorithms) from 42 to 11 attributes are *duration, service, src_bytes, dst_byte, count, srv_count, dst_host_count, dst_host_srv_count, dst_host_diff_srv_rate, dst_host_same_src_port_rate, and dst_host_serror_rate.* Our results are summarized in the Table 3 is comparison of different 5 classes. Define membership function equal than 0.5. It is, however, used after our experiments are performed, to evaluate the results and calculate the *detection rate* and *false alarm rate*. As shown in Table 3, the *probe* attack type had detection rate is high as 100% and *DoS* attack type had low detection rate as 82.05%. The results show that the performance of a proposed approach based on fuzzy *c*-means after using rough set for select important attribute is good.

Using rough sets, we are able to successfully identify pieces of information that succinctly characterize computer activity without missing information.

Table 3. Experiment results of fuzzy *c*-means with 11 # of features.

Attack Types	# of Records	Hit	Miss	Detection Rate	False Alarm Rate
Normal	5,763	5,749	14	99.76%	0.24%
Probe	2,164	2,164	0	100%	0%
DoS	3,530	2,897	634	82.05%	17.96%
U2R	70	67	3	95.71%	4.29%
R2L	6,689	6,145	544	91.87%	8.13%
Summary	18,216	17,022	1,195	93.45%	

7. Conclusions and Future Works

In this paper we apply rough set based methods with data reduction and to identify subset of features for network security and using fuzzy *c*-means to intrusion detection to avoid a hard definition between normal class and intrusion class. Features selection methods aim at selecting a small or prespecified number of features leading to the best possible performance of the entire classifier. The subset of selected features is then used to detect intrusions and to filter out redundant, spurious information.

The task of identifying and selecting a useful subset of features, used to represent patterns from a larger set of often mutually redundant or even irrelevant features. Therefore, the main goal of feature subset selection is to reduce the number of features used in classification while maintaining acceptable classification accuracy.

The experiment results show that rough set method is suitable for cleaning data and attribute selection. This method is efficient and reduces the amount of data set for handle data. The advantage of using fuzzy c-means is that it allows one to represent concepts that could be considered to be in more than one category (or from another point of view it allows representation of overlapping categories). It is shown that rough set and fuzzy c-means based IDSs using a reduced number of features can deliver enhanced performance).The result attained for detection of different intrusion/attacks may not be high in some cases but the proposed model can complement existing systems when similar cases arise.

Intrusion detection model is a composition model that needs various theories and techniques. One or two models can hardly offer satisfying results. We plan to apply other theories and techniques to operate in a high accurate and low false alarm rate in intrusion detection in our future work.

Acknowledgements

This work has been partially funded by Suan Dusit Rajabhat University, Bangkok Thailand. (http://www.dusit.ac.th). The authors would like to thank the anonymous reviewers for their helpful and insightful comments and suggestions.

References

[1] Abouzakhar N. S. and Manson G. A., "Evaluation of Intelligent Intrusion Detection Models," *International Journal of Digital Evidence*, vol. 3, no. 1, pp. 1-20, 2004.

[2] Abraham A., Grosan C. and Chen Y., "Cyber Security and the Evolution of Intrusion Detection Systems," *Journal of Educational Technology*, *Special Issue in Knowledge Management*, vol. 1, no. 1, pp. 74-81, 2005.

[3] Albayrak S. and Amasyali F., "Fuzzy C-Means Clustering on Medical Diagnostic Systems," *in Proceedings of the 12th Turkish Symposium on Artificial Intelligence and Artificial Neural Networks (TAINN'2003)*, Turkey, pp. 1-6, 2003.

[4] Bezdek J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[5] Chakraborty G. and Chakraborty B., "A Rough-GA Hybrid Algorithm For Rule Extraction From Large Data," *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA'2004)*, Boston, MA, USA, pp. 85-90, 2004.

[6] Chen W. H., Hsu S. H., and Shen H. P., "Application of SVM and ANN for Intrusion Detection," *Computers and Operations Research*, vol. 32, pp. 2617-2634, 2005.

[7] Chimphlee W., Abdullah A. H., Sap M. N. M., Chimphlee S., and Srinoy S., "Integrating Genetic Algorithms and Fuzzy c-Means for Anomaly Detection," *in Proceedings of the IEEE Indicon 2005 Conference*, Chennai, India, pp. 575-579, 2005.

[8] Chimphlee W., Sap M. N. M., Abdullah A. H., Chimphlee S., and Srinoy S., "An Integrated Model of Intrusion Detection Based on Fuzzy Clustering and Rule Learning for Identify Attacks Classes," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 5, no. 12, pp. 82-88, 2005.

[9] Chouchoulas A. and Shen Q., "Rough Set-Aided Keyword Reduction for Text Categorization," *Application Artificial Intelligence*, vol. 15, no. 9, pp. 843-873, 2001.

[10] Cox E., *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*, Elsevier, 2005.

[11] Dash M. and Liu H., "Feature Selection for Classification," *Intelligent Data Analysis*, vol. 1, no. 3, pp. 131-156, 1997.

[12] Denning D. E., "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222-232, 1987.

[13] Giacinto G., Roli F., and Didaci L., "Fusion of Multiple Classifiers for Intrusion Detection in Computer Networks," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1795-1803, 2003.

[14] Godínez F., Hutter D., and Monroy R., "Attribute Reduction for Effective Intrusion Detection," *in Proceedings of the Advances in Web Intelligence, 2nd International Atlantic Web Intelligence Conference (AWIC'2004)*, Cancun, Mexico, pp. 74-83, 2004.

[15] Gu G., Fogla P., Dagon D., Lee W., and Skoric B., *An Information-Theoretic Measure of Intrusion Detection Capability*, College of Computing, Georgia Institute of Technology, 2005.

[16] Helmer G., Wong J. S. K., Honavar V. G., Miller L., and Wang Y., "Lightweight Agents for Intrusion Detection," *Journal of Systems and Software*, vol. 67, no. 2, pp. 109-122, 2003.

[17] Ilgun K., "USTAT: A Real-Time Intrusion Detection System for UNIX," *in Proceedings of the 1993 IEEE Symposium on Security and Privacy*, Oakland, California, USA, pp. 16-28, 1993.

[18] Jensen R. and Shen Q., "Fuzzy-Rough Data Reduction with Ant Colony Optimization," *Fuzzy Sets and Systems*, vol. 149, pp. 5-20, 2005.

[19] Jensen R. and Shen Q., "Rough and Fuzzy Sets for Dimensionality Reduction," *in Proceedings of the 2001 UK Workshop on Computational Intelligence*, UK, pp. 69-74, 2001.
[20] KDD, *The 3rd International Knowledge Discovery and Data Mining Tools Competition (KDDCup'1999)*, University of California, Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, California, Irvine, 2005.
[21] Laskov P., Rieck K., Schäfer C., and Müller K. R., "Visualization of Anomaly Detection Using Prediction Sensitivity," *in Proceedings of Sicherheit*, GI, pp. 197-208, 2005.
[22] Marchette D. J., *Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint*, Springer Verlag, New York, USA, 2001.
[23] Mukkamala S. and Sung A. H., "Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines," *Journal of the Transportation Research Board of the National Academies*, vol. 1, no.1822, pp. 33-39, 2003.
[24] Ohrm A., *ROSETTA Technical Reference Manual*, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, 2000.
[25] Wagner D. and Soto P., "Mimicry Attacks on Host-Based Intrusion Detection Systems," *in Proceedings of the 9th ACM Conference on Computer and Communications Security*, Washington DC, USA, pp. 255-264, 2002.
[26] Wang Q. and Megaloikonomou V., "A Clustering Algorithm for Intrusion Detection," *in Proceedings of the SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, SPIE, Orlando, Florida, USA, pp. 31-38, 2005.
[27] Zhang M. and Yao J., "A Rough Sets Based Approach to Feature Selection," *in Proceedings of the 23rd International Conference of NAFIPS*, Banff, Canada, pp. 434-439, 2004.

Witcha Chimphlee received his BSc degree in computer science from Phetburi Rajabhat University, Thailand in 1992, and his MSc degree in computer science from National Institute of Development Administration (NIDA), Thailand in 1999. Since 1992, he has been working as a lecturer at Suan Dusit Rajabhat University. Currently, he is a PhD candidate in computer science in the Faculty of Computer Science and Information Systems at University Technology of Malaysia. His research interests include intrusion detection system, data mining, and fuzzy clustering. He is a member of IEEE Computer Society.

Abdul Hanan Abdullah received his BSc and MSc in computer science from the University of San Francisco and his PhD degree in the same field from the Aston University, United Kingdom in 1995. Currently, he is a professor and the dean of the Faculty of Computer Science and Information Technology, Universiti Teknologi Malaysia. His research interests include network security, grid computing and active network. Currently, he is supervising six PhD students.

Mohd Noor Md Sap received his PhD degree in computer science from the University of Strathclyde, United Kingdom in 1994. Currently, he is an associate professor and the head of the Database Laboratory (DATALAB), at the Faculty of Computer Science and Information Technology, Universiti Teknologi Malaysia. He has published over 75 journal and conference papers. He has also received research grants from the UTM and MOSTE. He is currently the head of editorial board of Journal of Information Technology. He received 1998's Excellent Teacher IT Award from Malaysian National Computer Confederation (MNCC). He is a member of ACM Computer Society. His research interests are in knowledge discovery, information retrieval, machine learning, information security, and decision support.

Siriporn Chimphlee received the BNS degree in nursing from Kuakarun College of Nursing, Thailand in 1996, and MS degree in applied statistics from National Institute of Development Administration (NIDA), Thailand in 2001. Since 2000, she has been working as a lecturer at Suan Dusit Rajabhat University. Currently she is a PhD candidate in computer science in Faculty of Computer Science and Information Systems at University Technology of Malaysia. Her research interests include data mining, web mining, and soft computing.

Surat Srinoy received the MSc degree in computer science from Faculty of Computer Engineering Chulalongkorn University, Thailand in 1999. He holds vice director of the Office of Academic Resources and Information Technology at Suan Dusit Rajabhat University. Currently, he is a PhD student in computer engineering at Mahanakorn University of Technology, Bangkok, Thailand. His research interests include network security, intrusion detection system, and immunoinformatics.