Ensemble of One-class Classifiers for Network Intrusion Detection System

Anazida Zainal[1], Mohd Aizaini Maarof[2], Siti Mariyam Shamsuddin[3] and Ajith Abraham[4]
[1,2,3]*Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia*
[4]*Center of Excellence for Quantifiable Quality of Service (Q2S), Norwegian University of Science and Technology, Trondheim, Norway*
Email: {anazida, aizaini, mariyam}@utm.my, ajith.abraham@ieee.org

Abstract

To achieve high accuracy while lowering false alarm rates are major challenges in designing an intrusion detection system. In addressing this issue, this paper proposes an ensemble of one-class classifiers where each uses different learning paradigms. The techniques deployed in this ensemble model are; Linear Genetic Programming (LGP), Adaptive Neural Fuzzy Inference System (ANFIS) and Random Forest (RF). The strengths from the individual models were evaluated and ensemble rule was formulated. Empirical results show an improvement in detection accuracy for all classes of network traffic; Normal, Probe, DoS, U2R and R2L. RF, which is an ensemble learning technique that generates many classification trees and aggregates the individual result was also able to address imbalance dataset problem that many of machine learning techniques fail to sufficiently address it.

1. Introduction

The recent growth in Internet has also created many problems concerning security. Intrusion detection is last mechanism that can be implemented to safeguard the network. In general, it will analyze the network traffic and look for potential threats. Two types of intrusion detection system (IDS) are; misuse and anomaly. Misuse looks for known attacks called attack signatures while anomaly is based on model of normalcy. A significant deviation from this model of reference, indicates a potential threat. Both approaches suffer several drawbacks. Misuse detection requires frequent updates of signatures to ensure a good detection while anomaly suffers a high false positive rate. Thus, the challenge is to surpass these two problems and come up with solution that can give a good accuracy while retaining low false positive rate. Various intelligent paradigms have been used in intrusion detection. Among them are Neural Network [1], Support Vector Machine [1] and Artificial Immune System [2]. Statistical methods have also been explored to solve problems in IDS.

The purpose of this paper is to address the issue of accuracy and false alarm rate in IDS. Here we employed two means; first is to select the relevant significant features, which represent patterns of the traffic and second is to engineer multiple classifiers with different learning paradigms to form an ensemble classifier model. The organization of this paper is as follows: Section 2 discusses the background and related works on ensemble approach in IDS. Section 3 present the various techniques used and Section 4 describes the flow of the experiment. Section 5 presents the results and discussion on findings. Finally, Section 6 concludes the paper.

2. Related Research

The problem of huge network traffic data size and the invisibility of intrusive patterns which normally are hidden among the irrelevant and redundant features have posed a great challenge in the domain of intrusion detection [3]. One way to address this issue is to reduce these input features in order to disclose the hidden significant features. Thus, an accurate classification can be achieved. Besides identifying significant features that can represent intrusive patterns, the choice of classifier can also influence the accuracy and classification of an attack. The literature suggests that hybrid or assembling multiple classifiers can improve the accuracy of a detection [1][4]. According to Chebrolu et al. [4], an important advantage for combining redundant and complementary classifiers is to increase robustness, accuracy and better overall generalization. Mukkamala et al. [5] demonstrated the use of ensemble classifiers gave the best accuracy for

each category of attack patterns. Ensemble methods aim at improving the predictive performance of a given statistical learning or model fitting technique. The general principle of ensemble methods is to construct a linear combination of some model fitting method, instead of using a single fit of the method. In designing a classifier, the first step is to carefully construct different connectional models to achieve best generalization performance for classifiers. Chebrolu et al. [4] proposed CART-BN approach, where CART performed best for *Normal*, *Probe* and *U2R* and the ensemble approach worked best for *R2L* and *DoS*. Meanwhile, Abraham et al. [6] illustrated that ensemble Decision Tree was suitable for *Normal*, LGP for *Probe*, *DoS* and *R2L* and Fuzzy classifier was for *R2L*. Abraham et al. [7] also demonstrated the ability of their proposed ensemble structure in modeling light-weight distributed IDS. Meanwhile, Mukkamala et al. [1] proposed three variants of Neural Networks, SVM and MARS as components in their IDS. This combining approach has demonstrated better performance when compared to single classifier approach. Here, we have chosen three soft computing techniques to develop our classifiers and they are: Linear Genetic Programming, Adaptive Neural Fuzzy Inference and Random Forest.

3. Computational Intelligent Techniques

We used a hybrid Rough Set and Discrete Particle Swarm (DPSO) to form a 2-tier feature selection process and came up with five different feature subsets. Each represents one of five different classes of network traffic. We built our ensemble classifier model using three different machine learning techniques and they are; Linear Genetic Programming (LGP), Adaptive Neural Fuzzy Inference System (ANFIS) and Random Forest (RF). The subsequent subsections will briefly describe these techniques

3.1. Linear Genetic Programming

Recent developments in GP, which include increased speed through use of linear genomes constructed of machine code instructions and development of homologues crossover operators have motivated the study in network security issues [8].

Genetic programming is a technique to automatically discover computer programs using the principles of Darwinian evolution [9]. It can create a working computer program from a high-level problem statement of the problem and breeds a population of programs to solve a problem. GP iteratively transforms a population of computer programs into a new generation of program by applying genetic operations. These genetic operations include crossover, mutation, reproduction, gene duplication and gene deletion [9]. The fitness of the resulting solutions is evaluated and suitable selection strategy is then applied to determine which solutions will be maintained into the next generation [7]. GP algorithm can be found in [10].

Linear genetic programming is a variant of the GP technique which uses a specific linear representation of computer programs. Abraham et al. [7] demonstrated the capability of three GP variants in the application of IDS where Multi Expression Programming (MEP) outperformed the rest in 3 cases except Probe and DoS. It also came up with very few discriminative features (3, 4, 6, 2 and 7) in which its classification score is above 95% in all cases. Meanwhile Hansen et al. [8] claimed that GP could be executed in realtime due to its detection speed and high level of accuracy. LGP could outperform SVM and ANN in terms of detection accuracy if the population size, program size, crossover rate and mutation rate are appropriately chosen [5].

3.2. Adaptive Neuro- Fuzzy Inference System

Due to complex relationships that exist between the features and the nature of the traffic data which has the grey boundary between normal and intrusive, fuzzy inference system is among the recent approaches which were deployed in intrusion detection. Similar to the work by Toosi and Kahani [11], we deployed ANFIS due to difficulty in determining the parameters associated with variations in the data values to the chosen membership function. ANFIS is the hybrid of approximate reasoning method with the learning capabilities of neural network. In ANFIS, the learning mechanism is implemented using a hybrid supervised learning approach.

Figure 1 shows the structure of ANFIS. The square and circle nodes are for adaptive nodes with parameters and fixed nodes without parameters, respectively. The first layer consists of square nodes that perform fuzzification with chosen membership function. The parameters in this layer are called premise parameters. In the second layer T-norm operation is performed to produce the firing strength of each rule. The ratio of i^{th} rule of the firing strength to the sum of all rules' firing strength is calculated in the third layer, generating the normalized firing strengths. The fourth layer consists of square nodes that perform multiplication of normalized firing strengths with the corresponding rule. The parameters in this layer are called consequent parameters. The overall output is calculated by the sum of all incoming signals in the fifth layer [12].

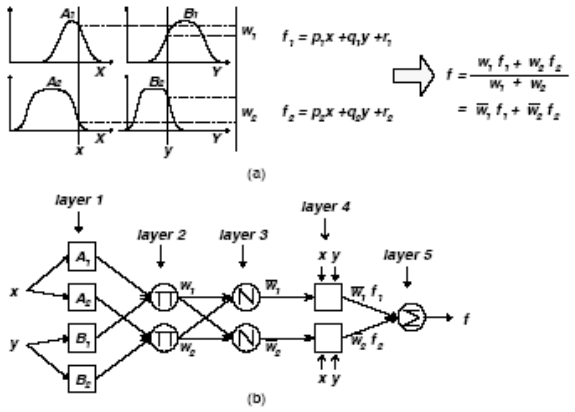

Figure 1. **(a) Sugeno Fuzzy Reasoning; (b) equivalent ANFIS structure [12]**

Toosi and Kahani [11] applied ANFIS in doing classification for KDDCup 1999 dataset and used all the features (41) in coming up with five FIS. Genetic Algorithm (GA) was used to optimize the structure of their fuzzy decision engine. Different learning style of fuzzy inference system was deployed by Abadeh et al. [13] where GA based learning was adopted and their experiment was to discriminate between normal and attack.

3.2. Random Forest

The random forests [14] are an ensemble of unpruned classification or regression trees. In general, random forest generates many classification trees and a tree classification algorithm is used to construct a tree with different bootstrap sample from original data using a tree classification algorithm. After the forest is formed, a new object that needs to be classified is put down each of the tree in the forest for classification. Each tree gives a vote about the class of the object. The forest chooses the class with the most votes [15]. By injecting randomness at each node of the grown tree, it has improved accuracy. RF algorithm is given below [16]:

1. Build bootstrapped sample B_i from the original dataset D, where $|B_i| = |D|$ and examples are chosen at random with replacement from D.
2. Construct a tree τ_i, using B_i as the training dataset using the standard decision tree algorithm with the following modifications:
   a. At each node in the tree , restrict the set of candidate attributes to a randomly selected subset ($x_1, x_2, x_3, \ldots, x_k$), where k = *no. of features*.
   b. Do not prune the tree.
3. Repeat steps (1) and (2) for $i = 1, \ldots,$ *no. of trees*, creating a forest of trees τ_i, derived from different bootstrap samples.
4. When classifying an example x, aggregate the decisions (votes) over all trees τ_i in the forest. If $\tau_i(x)$ is the class of x as determined by tree τ_i, then the predicted class of x is the class that occurs most often in the ensemble, i.e. the class with the majority votes.

Random Forest has been applied in various domains such as modelling [17][18], prediction [19] and intrusion detection system [15][20]. Zhang and Zulkernine [15] implemented RF in their hybrid IDS to detect known intrusion. They used the outlier detection provided by RF to detect unknown intrusion. Its ability to produce low classification error and to provide feature ranking has attracted Dong et al. [20] to use the technique to develop lightweight IDS, which focused on single attack.

4. Experimental Setup

This study used KDD Cup 1999 data set that was extracted 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 operated as a real environment and injected with multiple attacks. Each TCP/IP connection has a total of 41 qualitative and quantitative features where some are derived features. Features were labeled from 1 to 41. The type of attacks belongs to four main categories, namely, Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probing.

The training and testing data used in this study comprises of 5,092 and 6,890 records respectively as shown in Table 1. The composition of these sample data maintains the actual distribution of KDD Cup 1999 data.

Table 1: Training and Testing Data

	Normal	**Probe**	**DoS**	**U2R**	**R2L**
Training	1000	500	3002	27	563
Testing	1400	700	4202	25	563

Experiments presented in this paper are of supervised training and its flow is depicted in Figure 2.

The process to obtain important features was done offline. Each of the classifiers was trained using the same training data.

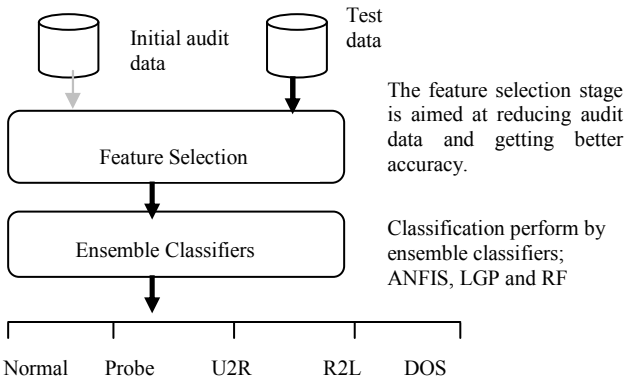

Figure 2: Experimental Flow

Rough-Discrete Particle Swarm Optimization (Rough-DPSO) was used to selectively choose significant features. Detail feature selection procedure of can be found in [21]. The reduced features are:

1. Normal (8 features) :
   f12, f31, f32, f33, f35, f36, f37 and f41
2. Probe (6 features) : f2, f3, f23, f34, f36 and f40
3. DoS (8 features) :
   f5, f10, f24, f29, f33, f34, f38 and f40
4. U2R (6 features) : f3, f4, f6, f14, f17 and f22
5. R2L (6 features) : f3, f4, f10, f23, f33 and f36.

The *neuro-fuzzy* (ANFIS) classifier was trained at 300 epochs of learning and two membership functions (MF) in the form of Bell-shape were used for the input and output fuzzy sets. Five ANFIS were produced to individually represent the five classes of the network traffic. For LGP classifier, we used the population size of 2048 and below, the mutation rate between 78.1% to 96.7%, and the crossover rate from 30.1% to 71.7%. Meanwhile, we used three features as a node split factor in building the trees in RF. The performance of each classifier was individually evaluated prior to their ensemble construction. The strength of individual classifier was used as a basis to assign the individual weight in the ensemble model. The individual performance of the classifiers is shown in Figures 3 and 4. Further discussion is given in Section 5. We have evaluated several weights for the classifiers and found that the following expression gives a good performance in the ensemble model:

$$D_{prob} = (0.5 \text{x} \text{LGP}_{prob}) + (0.1 \text{x} \text{ANFIS}_{prob}) + (0.4 \text{x} \text{RF}_{prob})$$

where 0.5, 0.1, and 0.4 are the weights. D_{prob} is the accumulated decision and LGP_{prob}, ANFIS_{prob} and RF_{prob} are the scores from the respective classifiers.

5. Results and Discussion

The results for the individual classifier and ensemble classifiers are summarized in Table 2. The accuracy, False Positive and True Positive are calculated based on the following equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$False\ Positive = \frac{FP}{FP + TN}$$

$$True\ Positive = \frac{TP}{Total\_class\_samples}$$

The above True Positive calculation would give an indicator of how well a classifier can recognize class specific input being investigated. This is to avoid misleading true positive performance due to imbalance testing data. The results obtained are tabulated in Table 2. We further analyzed the results to explore the discriminative powers of each technique. Figure 3 shows accuracy rate of each technique plotted against each class of traffic; class 1 denotes Normal, class 2 Probe, class 3 DoS, class 4 U2R and class 5 R2L. In general, the performance of LGP is superior when compared to the other two classifiers while both ANFIS and RF are almost at par with each other. In general, their performances are poor for DoS. Two possibilities that can explain this situation; firstly it may be due to the DoS class-specific feature which may not be well selected. Secondly, it may be due to the imbalanced data problem which will be explained later.

Table 2: Performance of the three classifiers and the ensemble model

Classes	LGP			ANFIS			Random Forest			Ensemble Model		
	Accuracy	FP	TP	Accuracy	FP	TP	Accuracy	FP	TP	Accuracy	FP	TP
Normal	98.83	0.0029	0.9971	96.31	0.0029	0.9631	93.16	0.0029	0.9970	**99.27**	0.0029	0.9971
Probe	99.68	0.0000	0.9986	95.41	0.0000	0.5557	95.76	0.0000	0.9990	**99.88**	0.0000	0.9914
DoS	97.45	0.0000	0.9743	92.66	0.0007	0.8877	91.45	0.0121	0.9055	**98.26**	0.0000	0.9743
U2R	99.91	0.0000	0.8000	99.77	0.0000	0.4400	99.13	0.0007	0.8800	**99.96**	0.0000	0.8800
R2L	99.63	0.0000	0.9858	99.49	0.0000	0.9503	98.87	0.0000	0.9965	**99.79**	0.0000	0.9858

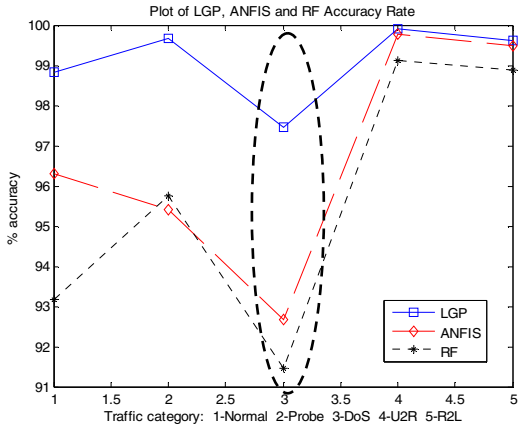

Figure 3: Individual performance based on accuracy rate

Figure 4 shows the true positive performance of all classifiers. The illustration reveals that LGP and ANFIS have poor performance on class 4 (U2R) whereas the performance of RF is relatively better. Figures 3 and 4 suggest that both class 3 (DoS) and class 4 (U2R) are relatively difficult to classify. DoS, which constitutes the largest number of sample data (58.96%) and U2R has the least sample data (0.53%) represent two extreme situations, thus imposing an imbalanced data problem. Data imbalance occurs when either the number of patterns of a class is much larger or smaller than that of the other classes. This study reveals that the performance of RF is relatively stable throughout all classes.

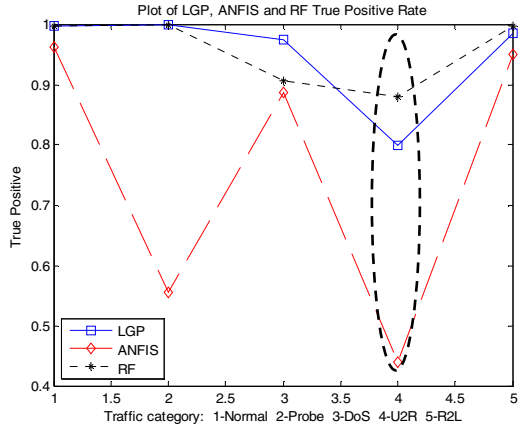

Figure 4: Individual performance based on true positive rate

According to [22] data imbalance is one of the causes that degrade the performance of machine learning algorithms in classifications. This study confirms that both LGP and ANFIS fail to perform well when dealing with imbalanced dataset. On the other hand, RF performs reasonably well relative to others particularly in small data category (U2R). The empirical results conforms the claim made by [16] in which they conclude that RF is robust and it can handle imbalanced data problem. [16] argued that the robustness of RF lies on random selection of features at the node and its bootstrapping strategy during the creation of trees.

Figure 5 compares the accuracy performance of our ensemble model against the best individual classifier, LGP. The ensemble behaves very similar to LGP with slight performance improvement in all the classes. This finding suggests that the ensemble model is the best approach to provide high accuracy while keeping low false positive. This is perhaps due to the complementary role from each of the members in the ensemble model.

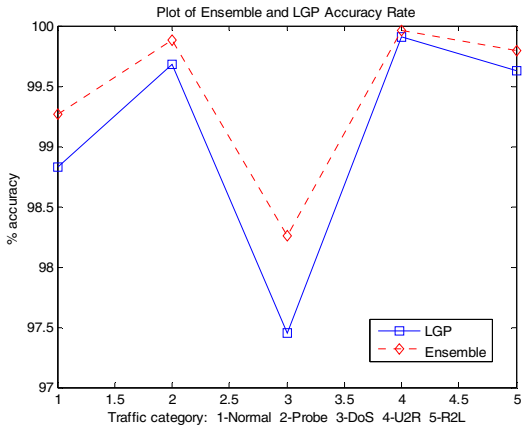

Figure 5: Accuracy rate of ensemble vs LGP

6. Conclusions

In this paper, we have demonstrated that ensemble of different learning paradigms can improve the detection accuracy. This was achieved by assigning proper weight to the individual classifiers in the ensemble model. Based on our experiment, LGP has performed well in all the classes except the U2R attacks. In contrary, RF shows a better true positive rate for U2R class. Thus, by including the RF in the assemble model, the overall performance particularly the result for U2R class has improved.

The assignment of the weights to the individual classifier in the ensemble model is very important. We plan to investigate a more systematic method that can explicitly give the correlation among the weight values and investigate how the values influence the classification result.

7. Acknowledgments

We would like to thank Universiti Teknologi Malaysia, Ministry of Higher Education Malaysia and Ministry of Science and Innovation Malaysia for sponsoring this study.

8. References

[1] S. Mukkamala, A.H. Hung and A. Abraham, "Intrusion Detection Using an Ensemble of Intelligent Paradigms." *Journal of Network and Computer Applications*, Vol. 28(2005), 167-182.

[2] J.W.Kim, "Integrating Artificial Immune Algorithms for Intrusion Detection." PhD Thesis, Department of Computer Science, University College of London, 2002.

[3] 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(2004) , 468-482.

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

[5] S. Mukkamala, A.H. Sung and A. Abraham, "Modeling Intrusion Detection Systems Using Linear Genetic Programming Approach." LNCS 3029, Springer Hiedelberg, 2004, pp. 633-642.

[6] A. Abraham and R. Jain, "Soft Computing Models for Network Intrusion Detection Systems." Soft Computing in Knowledge Discovery: Methods and Applications, Springer Chap 16, 2004, 20pp.

[7] A. Abraham, C. Grosan, and C.M. Vide, "Evolutionary Design of Intrusion Detection Programs." *International Journal of Network Security*, Vol. 4, No. 3, 2007, pp. 328-339.

[8] J.V., Hansen, P.B. Lowry, R.D. Meservy and D.M. McDonald, "Genetic Programming for Prevention of Cyberterrorism through Dynamic and Evolving Intrusion Detection." *Journal of Decision Support Systems* Vol. 43, 2007, pp. 1362-1374.

[9] Koza J.R. and Poli. R. 2003. A Genetic Programming Tutorial. http://www.genetic-programming.com/jkpdf /burke2003tutorial.pdf

[10] K.M. Faraoun and A.Boukelif, "Genetic Programming Approach for Multi-Category Pattern Classification Applied to Network Intrusions Detection." *International Journal of Computational Intelligence* Vol. 3, No. 1(2006) 79-90.

[11] A.N. Toosi, and M. Kahani, "A new approach to intrusion detection based on a evolutionary soft computing model using neuro-fuzzy classifiers." *Journal of Computer Communications*, Vol. 30, 2007, pp. 2201-2212.

[12] J.R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System." *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 3, May 1993, pp. 665-685.

[13] M.S. Abadeh, J. Habibi and C. Lucas, "Intrusion Detection Using a Fuzzy Genetics-based Learning Algorithm." *Journal of Network and Computer Applications* Vol 30, 2007, pp. 414-428.

[14] L. Breimann, 2001, "Random Forests." Journal of Machine Learning, Kluwer Academic, Netherland, Vol.45, 2001, pp. 5-32.

[15] J. Zhang, and M. Zulkernine, 2006. A Hybrid Network Intrusion Detection Technique Using Random Forests. In Proceedings of the IEEE First International Conference on Availability, Reliability and Security (ARES'06).

[16] T.M. Khoshgoftaar, M. Golawala and J. Van Hulse, "An Empirical Study of Learning from Imbalanced Data Using Random Forest." Proceedings of the 19th. IEEE Conference on Tools with Artificial Intelligence. 2007, pp. 310-317.

[17] P. Xu, and F. Jelinek, " Random Forests and the Data Sparseness Problem in Language Modeling." *Journal of Computer Speech and Language*, Vol. 21, Issue 1(Jan 2007) pp. 105-152.

[18] J. Peters, B. De Baets, N.E.C Verhoest, R. Samson, S. Degroeve, P. De Becker and W. Huybrechts, "Random Forests as a Tool for Ecohydrological Distribution Modelling." *Journal of Ecological Modelling*, Vol 207, Issue 2-4, October 2007, pp. 304-318.

[19] B. Lariviere, and D. Van den Poel, "Predicting Customer Retention and Profitability by Using Random Forests and Regression Forests Techniques." *Journal of Expert Systems with Applications*, Vol. 29, Issue 2, (August 2005) pp. 472-482.

[20] S.K. Dong, M.L. Sang, and S.P. Jong, "Building Lightweight Intrusion Detection System Based on Random Forest." LNCS 3973, Springer-Verlag, Berlin Heidelberg (2006) pp. 224-230.

[21] A. Zainal, M.A. Maarof and S.M. Shamsuddin, "Feature Selection Using Rough-DPSO in Anomaly Detection." LNCS 4705, Part 1 Springer Hiedelberg (2007) pp. 512-524.

[22] P.Kang, and S.Cho, "EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems." ICONIP 2006, LNCS 4233, Part 1 Springer Hiedelberg, 2006, pp. 837-846