Journal of Information Assurance and Security 4 (2009) 217-225

Ensemble Classifiers for Network Intrusion Detection System

Anazida Zainal[1], Mohd Aizaini Maarof[2] and Siti Mariyam Shamsuddin[3]

[1,2] Information Assurance and Security Research Group (IASRG)
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
anazida@utm.my, aizaini@utm.my

[3]Soft Computing Research Group (SCRG)
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
mariyam@utm.my

Abstract: Two of the major challenges in designing anomaly intrusion detection are to maximize detection accuracy and to minimize false alarm rate. In addressing this issue, this paper proposes an ensemble of one-class classifiers where each adopts 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. Prior to classification, a 2-tier feature selection process was performed to expedite the detection process. Empirical results show an improvement in detection accuracy for all classes of network traffic; Normal, Probe, DoS, U2R and R2L. Random Forest, 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.

Keywords: ensemble, ANFIS, genetic programming, random forest, intrusion detection and classification.

1. Introduction

The recent growth in Internet has also created many problems concerning security. Various security strategy were put forth to safeguard a network. Firewall as a basic packet filter alone is not sufficient to provide a secured network environment. Intrusion detection when coupled with firewall can provide a better and safer network. In general, an intrusion detection system (IDS) will analyze the network traffic and look for potential threats. Two types of intrusion 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. Graphical approach like Junction Tree (JT) was also found to be useful to segregate between normal and attack patterns. One particular advantage is its ability to illustrate the inter relation between attributes [3]. In recent years, the approach of using multiple classifiers were widely being used to solve many classification problems including IDS [4,5,6]. With a proper voting system and weighting assignment, this approach seems to improve the classification rate. Meanwhile, when dealing with a domain which deals with huge data size like network traffic, usually resources and time are greatly affected.

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: Major portion of section 2 discusses the background and related works on ensemble approach in IDS. Section 3 presents the various techniques used in this study 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 Works

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 [7]. 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 [1,6]. Classifier ensembles also known as committees are aggregations of several classifiers whose individual predictions are combined in some manner (e.g., averaging or voting) to form a final prediction [8]. An important advantage for combining redundant and complementary classifiers is to increase robustness, accuracy and better overall generalization in most applications [4,8]. Mukkamala et al. [9] 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,

Received June 10, 2009

1554-1010 $ 03.50 Dynamic Publishers, Inc

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. [6] 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. [10] illustrated that ensemble Decision Tree was suitable for *Normal*, LGP for *Probe*, *DoS* and *R2L* and Fuzzy classifier was for *R2L*. In their later work, Abraham et al. [11] 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. Giacinto et al.[5] took a slightly different approach. Their anomaly IDS was based on modular multiple classifier system where each module was designed for each group of protocols and services. Each module might contain either individual or combination of different classifiers. The modular architecture would allow putting a rejection threshold of each module as to optimize the overall attack detection rate given a desired total false alarm rate for the ensemble. They reported that there was an improvement on attack detection rate and significant reduction on false alarm.

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 Intelligence Techniques

Network traffic data is usually associated with large volume and having numerous fields that require careful examination by IDS. To alleviate the overhead problem, feature selection was performed prior to classification. Besides, selecting the significant features which signify each traffic class is to find the intrusive patterns or common properties are which often hidden within the irrelevant features [6]. They further commented that there are features that contain false correlation. Some of these features also may be redundant [12] and may have different discriminative power. Therefore, the aim of feature selection is to disclose these hidden significant features from the irrelevant features. Thus, an accurate and fast classification can be achieved. Each represents one of five different classes of network traffic. Meanwhile, the ensemble classifier model was built using three different machine learning techniques and they are; Linear Genetic Programming (LGP), Adaptive Neural Fuzzy Inference System (ANFIS) and Random Forest (RF). The hybridization of these intelligences was aimed at improving the classification capability of the IDS. The subsequent subsections will briefly describe these techniques

3.1 Preprocessing Stage

Feature Selection process implemented in this study utilized a hybrid approach where Rough Set Technique and Binary Particle Swarm (BPSO) were structured in hierarchical manner to form a 2-tier feature selection process. Features were obtained based on class-specific characteristics, thus each class had one specific feature set. Since BPSO uses heuristic technique and the initial feature candidates are 41, Rough Set techniques was used to eliminate the redundant features and rank the top 15 features for each classes of traffic (Normal, Probe, DoS, U2R and R2L). These significant features are termed as reducts.

3.1.1 Rough Set Technique (RST)

Pawlak [13] introduced rough set theory which assumes that every objective within the universe of discourse is associated with some information. RST has been used to solve problems in various areas. Among them are; uncertainty in electricity load analysis [14], fault diagnosis on diesel engine [15], feature extraction [11], knowledge discovery for diabetic children [17] and many others. Four basic concepts of RST are:

i) Indiscernibility of objects
ii) Lower and upper approximation
iii) Attribute Reduction
iv) Induction of Decision Rule

Subsequent paragraph gives an overview of the related Rough Set Theory taken from Hassanien et al. [17].

Definition 1 (information system).
Information system is a tuple (U,A) where U consists of objects and A consists of feature. Every $a \in A$ corresponds to the function $a : U \rightarrow V_a$, where V_a is value set of a. In applications, we often distinguish between conditional features C and decision features D, where $C \cap D = \emptyset$. In such cases, we define decision systems (U, C, D).

Definition 2 (indiscernibility relation).
Every subset of features $B \subseteq A$ induces indiscernibility relation:

$$Ind_B = \{(x, y) \in U \times U : \forall_{a \in B} \quad (1)$$

For every $x \in U$, there is an equivalent class $[x]_B$ in the partition of U defined by Ind_B.

Inconsistency in the decision table happen when two or more similar objects with matching descriptions but they belong to different classes.

Definition 3 (lower and upper approximation)
Given a set $B \subseteq A$, the lower and upper approximations of a set $Y \subseteq U$ are defined by, respectively,

$$BY = \cup_{x:[x]_B \subseteq X}[x]_B \quad (2)$$

$$BY = \cup_{x:[x]_B \cap X \neq \emptyset}[x]_B \quad (3)$$

Attribute reduction which is the third concept of RST is of the interest since it has the capability to eliminate the redundant and unimportant features.

Definition 7 (reducts)
Given a classification task related to the mapping a set of variables C to a set of labeling D, a reduct is a subset $R \subseteq C$ such that

$$\gamma(C, D) = \gamma(R, D) \quad (4)$$

and none of proper subsets of R satisfies analogous equality.

Definition 8 (reduct set)
Given a classification task mapping a set of variables C to a set of labeling D, a reduct set is defined with respect to the power set $P(C)$ as the set $R \subseteq P(C)$ such that $R = \{A \in$

$P(C) : \gamma(A, D) = \gamma(C, D)\}$. That is the reduct set is the set of all possible reducts of the equivalence relation denoted by *C* and *D*.

Definition 9 (minimal reduct)
A minimal reduct $R_{minimal}$ is the reduct such that $\|R\| \leq \|A\|$, for all $A \in R$. That is, the minimal reduct is the reduct of least cardinality for the equivalence relation denoted by *C* and *D*.

Definition 10 (core)
Attribute $c \in C$ is a core feature with respect to *D*, if and only if it belongs to all the reducts. We denote the set of all core features by core(*C*). If we denote by *R*(*C*) the set of all reducts, we can put

$$\text{Core}(C) = \cap_{R \in R(C)} R \quad (5)$$

The reducts computation and core of condition features from a decision table is actually the selection of significant features. The reducts produced represent the minimal set of features necessary to maintain classification capability given by a complete feature set. Rules are generated based on these reducts. These rules are the building blocks of the classifier model.

This study exploited the capability of reducts to size down the number of features to 15 (from 41 features).

3.1.2 Binary 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 behavior of each particle and its neighbours while they fly through search space [18]. Thus, particles have a tendency to fly towards the better and better search area over the course of search process [19]. The calculation of velocity and position are described as below:

$$V_{id} = wV_{id} + C1rand(\,)(P_{id} - X_{id}) + C2Rand(\,)(P_{gd} - X_{id}) \quad (6)$$
$$X_{id} = X_{id} + V_{id} \ (7)$$

C1 and *C2* 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 fly far from target regions before being tugged back, while high values result in abrupt movement toward or past target regions. Meanwhile, *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. $Xi = (x_{i1}, x_{i2}, \dots, x_{iD})$ represents the i^{th} particle and $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ represents the best previous position of the i^{th} particle.

This study employed the binary version of PSO to determine the need whether to include a feature or otherwise. Modification on velocity calculation and position were done to suit the binary nature of the feature selection domain. Apart from feature representation, [20] 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 Wang et al. [20].

3.2 Ensemble Intelligence for Classification

The effectiveness of a ensemble or multiple classifier approach also depends on the choice of the decision fusion function. To determine the decision function, the expected degree of diversity among classifiers should be taken into account [21]. Here, ensemble machine learning techniques with different learning paradigms were used to classify the network connection. Decision function was determined based on the individual performances on overall accuracy and true positive rates.

3.2.1 Linear Genetic Programming (LGP)

The 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 [22].

Genetic programming is a technique to automatically discover computer programs using the principles of Darwinian evolution [23]. 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 [23]. 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 [11]. GP algorithm can be found in [24].

Linear genetic programming is a variant of the GP technique which uses a specific linear representation of computer programs. The main difference in comparison to tree-based GP is the evolvable units are not the expressions of a functional programming language (like LISP), but the programs of an imperative language (like c/c++) [11]. Abraham et al. [11] further 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. [18] 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 [9].

3.2.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

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.

The fuzzy inference system refers to a process that maps the input characteristics to the input membership functions. There are two basic types of fuzzy inference system and they are Mamdani and Sugeno Fuzzy Models. The difference lies in how the output is determined. Mamdani Fuzzy Model was proposed as the very first attempt to map an input space to an output space based on experience of a human expert. An example of a Mamdani fuzzy rule is,

if (x is high) *then* (y is small)

and it is of a linguistic form. Similar to Mamdani model, Takagi Sugeno *if-then* rule's premise part is of linguistic form characterized by a membership function. Meanwhile, the consequent part is described by non-fuzzy equation of a fuzzy input variable. A fuzzy rule in a Sugeno fuzzy model has the form of,

*if (*x is high) *then* y=f(x)

ANFIS adopts the Takagi-Sugeno model. Similar to the work by Toosi and Kahani [25], 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 [26].

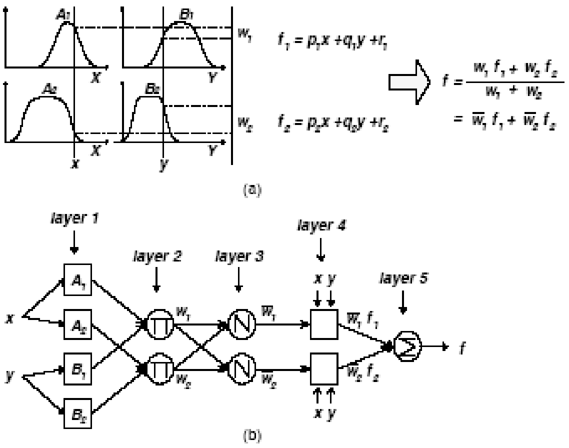

Figure 1. (a) Sugeno Fuzzy Reasoning; (b) equivalent ANFIS structure [26]

Toosi and Kahani [25] 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. [27] where GA based learning was adopted and their experiment was to discriminate between normal and attack.

3.2.3 Random Forest (RF)

The random forests [28] 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 [29]. By injecting randomness at each node of the grown tree, it has improved accuracy. RF algorithm is given below [30]:

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, … , 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 modeling [31,32], prediction [33] and intrusion detection system [29,34]. Zhang and Zulkernine [21] 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. [34] 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 and they are termed as f1, f2, f3,… and f41. The type of attacks belongs to four main categories, namely, Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probing.

(i) **Probe (Probing and Surveillance)**

It is characterized by the scanning activity done by attackers looking for opened and vulnerable ports at the victim machine. Understanding which ports are opened and what version of services they are will enable attacker to study further on the known vulnerabilities of those services. Thus, an attack can easily be made.

(ii) **DoS (Denial of Service)**

This attack is characterized by an explicit attempt to deny or prevent legitimate users from using the resources. For instance, attackers can flood the network thus preventing legitimate network traffic and disrupt a connection that will deny a legitimate user from accessing certain services. Packet filtering and disable the unnecessary ports may lessen the risks from this attack.

(iii) **U2R (User to Root)**

This attack normally start with accessing a normal local user account and later the attacker exploit the vulnerabilities to gain access to root and able to work with superuser privilege.

(iv) **R2L (Remote to Local)**

Attacker will normally send codes to a machine over a network and later he will exploit the victim's vulnerability and gain access to normal local user on the host.

Since the described machine learning approaches in the earlier sections are of supervised learning, the experiment was done on both training and testing phases. The training and testing data used in this study comprises of 5,092 and 6,890 records respectively as shown in Table 1 and all the data were scaled to [0,1][35]. The composition of these sample data maintains the actual distribution of KDD Cup 1999 data.

Table 1. Training and testing data.

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

The flow of the experiments presented in this paper is depicted in Figure 2. The process to obtain important features was done offline. Each of the classifiers (LGP, ANFIS and RF) was trained using the same training data.

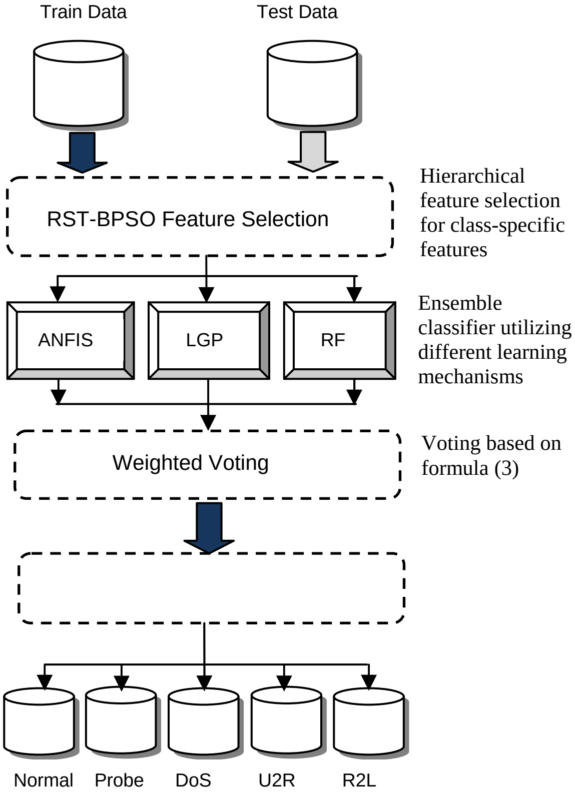
Train Data
Test Data
RST-BPSO Feature Selection
Hierarchical feature selection for class-specific features
ANFIS
LGP
RF
Ensemble classifier utilizing different learning mechanisms
Weighted Voting
Voting based on formula (3)
Normal
Probe
DoS
U2R
R2L

Figure 2. Experimental Flow

Rough-Discrete Particle Swarm Optimization (Rough-BPSO) was used to selectively choose significant features. Initial 41 features were reduced to 15 for all classes and they varies from one class to another. These reducts were then refined using Binary PSO. Details on feature selection procedure can be found in [36]. The obtained class-specific features are shown in Table 2. The number of reduced features ranges from 6 to 8, which roughly about 80% reduction.

Table 2. Reduced features

Class	Best result
Normal	f12, f31, f32, f33, f35, f36, f37 and f41
Probe	f2, f3, f23, f34, f36 and f40
DoS	f5, f10, f24, f29, f33, f34, f38 and f40
U2R	f3, f4, f6, f14, f17 and f22
R2L	f3, f4, f10, f23, f33 and f36

Meanwhile, Table 3 shows the error produced by the ANFIS when trained with few epoch numbers, 100, 300 and 500. The *neuro-fuzzy* (ANFIS) classifier was best trained at 300 epoch. No improvement on the error found beyond 300 epochs. Two membership functions (MF) in the form of Bell-shape were used for the input and output fuzzy sets. We have also experimented with two other MFs which were *trapezoidal* and *gaussian* forms on DoS with 300 epochs and they were compared with the performance of the Bell-shape MF. Each gave an error of 0.42137 and 0.31536 respectively. Therefore, it can be concluded that bell-shaped MFs are more suitable for the data used in this study.

Table 3. Number of epoch and errors

Class	Epoch	Error
Normal	100	0.300930
	300	**0.297297**
	500	0.297297
Probe	100	0.102732
	300	**0.095630**
	500	0.095630
DoS	100	0.431270
	300	**0.314400**
	500	0.314400
U2R	100	0.072144
	300	**0.071277**
	500	0.071277
R2L	100	0.272360
	300	**0.261350**
	500	0.261350

Five ANFIS were produced to individually represent the five classes of the network traffic. Maximum number of rules generated was 2^8 for both Normal and DoS. Minimum rules were 2^6 for Probe, U2R and R2L. Apparently, the compact size of rules generated in this study is far less than of Toosi and Kahani [25] which is 2^{41}. The number of rules can greatly affect the performance in terms of classification time.

As for LGP classifier, we used the following parameter settings as shown in Table 4.

Table 4. Number of epoch and errors

Parameter	Normal	Probe	DoS	U2R	R2L
Population size	2048	2048	2048	2048	2048
Instruction sets	addition, arithmetic, comparison, data transfer, multiplication, subtraction and trigonometric				
Mutation frequency (%)	97	95	78	95	95
Crossover frequency (%)	50	50	30	72	50
Number of demes	10	10	10	10	10
Maximum program size	512	512	512	512	512

We limit 1000 generations of classifier codes to evolve with average of 20 runs per generation. In the example of U2R classifier, it took 90 generations to be stabilized. Subsequent generations showed no improvement in terms of accuracy as shown in Figure 3.

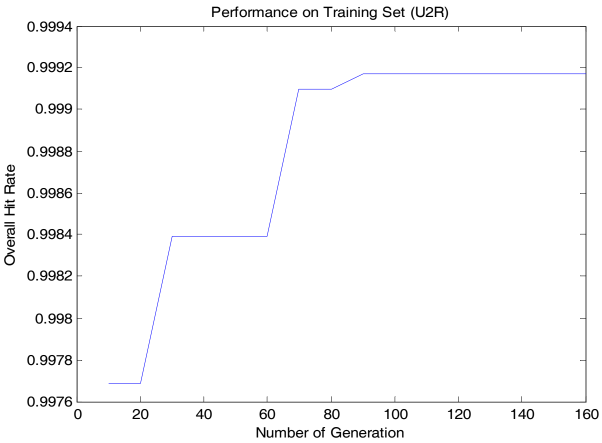

Figure 3. Relation between accuracy and number of generations for U2R

Meanwhile Figure 4 shows an example of C code excerpt for U2R classification program with f[0] (equivalent to f3) having the highest input impact.

```
L0: f[0]-=v[2];
L1: f[0]/=v[0];
L2: f[0]+=0.003250631503760815f;
L3: cflag=(f[0] < f[2]);
L4: f[0]+=v[2];
L5: f[0]=sqrt(f[0]);
L6: f[0]+=0.2609443664550781f;
L7: if (!cflag) f[0] = f[1];
L8: f[0]+=v[4];
L9: f[0]=-f[0];
L10:   f[0]+=v[3];
L11:   f[0]=fabs(f[0]);
L12:   f[0]-=v[5];
L13:   f[0]+=f[0];
L14:   f[0]=sqrt(f[0]);
```

Figure 4. An excerpt of C code for U2R 1-vs-rest classification program evolved using LGP

As in RF experiment, we used three features as a node split factor in building the trees. 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 5 and 6. 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:

$$D_{prob} = (0.5\text{x}LGP_{prob}) + (0.1\text{x}ANFIS_{prob}) + (0.4\text{x}RF_{prob}) \quad (8)$$

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 5. The accuracy, False Positive and True Positive are calculated based on the following equations.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

$$\text{False Positive} = \frac{FP}{FP+TN} \quad (5)$$

$$\text{True Positive} = \frac{TP}{Total\_class\_samples} \quad (10)$$

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 5.

Table 5. Individual and ensemble performances

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

We further analyzed the results to explore the discriminative powers of each technique. Figure 5 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.

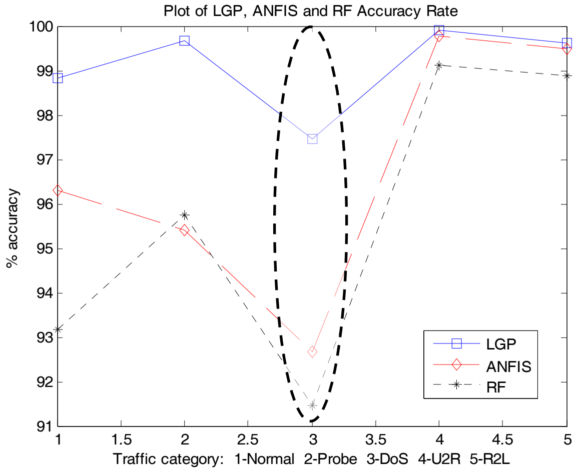

Figure 5. Individual performance based on accuracy rate

Figure 6 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.

According to [37] 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.

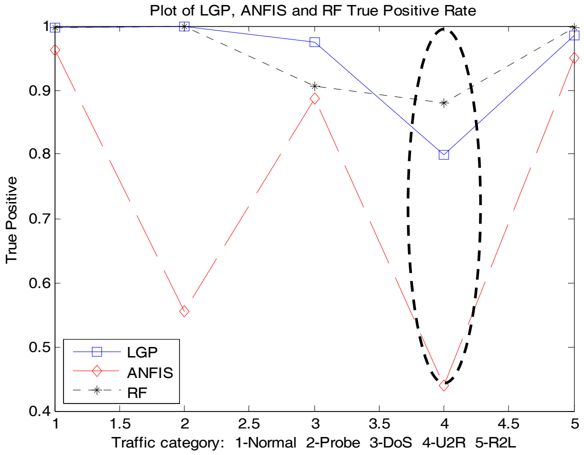

Figure 6. Individual performance based on true positive rate

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 Khoshgoftaar et al. [30] in which they conclude that RF is robust and it can handle imbalanced data problem. They 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 7 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.

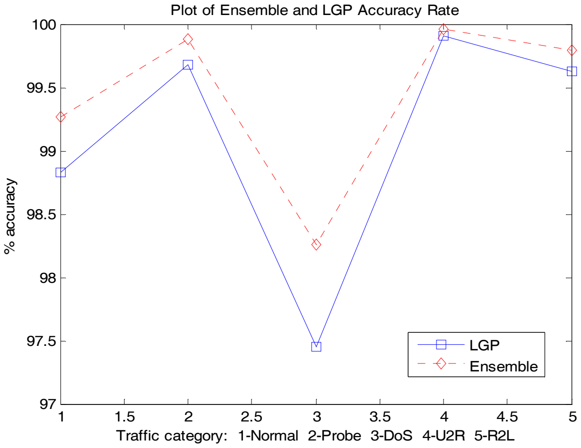

Figure 7. Accuracy rate of ensemble vs. LGP

6. Conclusion

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.

Acknowledgment

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

References

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

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

[3] E. Nikolova and V. Jecheva. "Anomaly Based Intrusion Detection Based on the Junction Tree Algorithm." *Journal of Information Assurance and Security*, 2, pp. 184-188, 2007.

[4] S. Peddabachigari, A. Abraham, C. Grosan and J. Thomas. "Modeling intrusion detection system using hybrid intelligent systems." *Journal of Network and Computer Applications*, 30, pp. 114-132, 2007.

[5] G. Giorgio, P. Roberto, R.D. Mauro and R. Fabio. "Intrusion detection in computer networks by a modular ensemble of one-class classifiers." *Journal of Information Fusion*, 9, pp. 69-82, 2008.

[6] 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(4), pp. 295-307, 2005.

[7] 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.rith: Select real-world application." *Journal of Information Fusion*, 9, pp. 4-20, 2008.

[8] N.K. Oza and K. Tumer, "Classifier ensembles: Select real-world applications." *Journal of Information Fusion*, 9, pp. 4-20, 2008.

[9] 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.

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

[11] A. Abraham, C. Grosan, and C.M. Vide, "Evolutionary Design of Intrusion Detection Programs." *International Journal of Network Security*, Vol. 4(3), pp. 328-339, 2007.

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

[13] Z. Pawlak. "Rough Sets." *International Journal of Computer and Information Science*, 11, pp. 341-356, 1982.

[14] P.F. Pai and T.C. Chen. "Rough Set Tehory with Discriminant Analysis in analyzing Electricity Loads." *Expert Systems with Applications*, 36, pp. 8799-8806, 2009.

[15] L. Shen, E.H. Tay, Q. Liangsheng and Y. Shen. "Fault Diagnosis using Rough Sets Theory." *Journal of Computer in Industry*, 43, pp. 61-72, 2000.

[16] L.Y. Zhai, L.P. Khoo and S. C. Fok. "Feature Extraction using Rough Set Theory and Genetic Algorithms – An Application for the Simplification of Product Quality Evaluation." *Journal of Computers & Industrial Engineering*, 43, pp. 661-676, 2002.

[17] A.E. Hassanien, M. E. Abdelhafez and H.S. Own. "Rough Sets Data Analysis in Knowledge Discovery: A Case of Kuwaiti Diabetic Children Patients." *Advances in Fuzzy Systems*, 13pp, 2008.

[18] S. Monteiro, T.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* 7(1), pp. 232–235, 2005.

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

[20] X. Wang, J. Yang, X. Teng, W. Xia, R. Jensen, "Feature Selection based on Rough Sets and Particle Swarm Optimization." *Pattern Recognition Letters* 28(4), pp. 459–471, 2007.

[21] F. Roli, J. Kittler, (Eds.). Multiple Classifier Systems. Springer-Verlag, Lecture Notes in Computer Science, vol. 2364, 2002.

[22] 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, pp. 1362-1374, 2007.

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

[24] 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(1), pp. 79-90, 2006.

[25] 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, pp. 2201-2212, 2007.

[26] J.R. Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System.” *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23(3), pp. 665-685, 1993

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

[28] L. Breimann, 2001, “Random Forests.” *Journal of Machine Learning*, Kluwer Academic, Netherland, Vol.45, pp. 5-32, 2001.

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

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

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

[32] 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*, 207(2-4), pp. 304-318, 2007.

[33] 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*, 29(2), pp. 472-482, 2005.

[34] 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.

[35] A. Abraham and C. Grosan. “Evolving Intrusion Detection Systems”, *in Genetic Systems Programming*, N. Nedjah, A. Abraham, L. M. Mourelle (eds), Springer Berlin, pp. 57-79, 2006.

[36] 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.

[37] 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

Author Biographies

Anazida Zainal is a PhD candidate at Universiti Teknologi Malaysia (UTM), Skudai, Johor, Malaysia. She is a member to Information Assurance and Security Research Group, UTM. She holds a MSc in Computer Science from Universiti Teknologi Malaysia and BSc in Computer Science from Rutgers University, New Jersey, United States. Her research interest focuses on the application of machine learning in computer security.

Mohd Aizaini Maarof is a Professor at Faculty of Computer Science and Information System, Universiti Teknologi Malaysia (UTM). He obtained his B.Sc (Computer Science) from Western Michigan University, M.Sc (Computer Science), from Central Michigan University, U.S.A and his Ph.D degree from Aston University, Birmingham, United Kingdom in the area of Information Technology (IT) Security. He is currently leading the Information and Assurance Research Group (IASRG). His research areas are Network Security, Web Content Filtering, MANET Security and Cryptography

Siti Mariyam Shamsuddin received her Bachelor and Master degree in Mathematics from New Jersey USA, and Phd in Pattern Recognition & Artificial Intelligence from Universiti Putra Malaysia (UPM), MALAYSIA. Currently, she is a Head of Soft Computing Research Group, k-Economy Research Alliance, Universiti Teknologi Malaysia (UTM), Johor MALAYSIA. Her research interests include the Fundamental Aspects of Soft Computing and its Application, Pattern Recognition, Forensic Document Analysis, and Geometric Modeling.