Intrusion Detection Using Rough Sets based Parallel Genetic Algorithm Hybrid Model

Wa'el M. Mahmud, Hamdy N.Agiza, and Elsayed Radwan

***Abstract*— Recently machine learning-based Intrusion Detection systems (IDs) have been subjected to extensive researches because they can detect both misuse and anomaly. Most of existing IDs use all features in the network packet to look for known intrusive patterns. Some of these features are irrelevant or redundant. Rough Set Classification (RSC), a modern learning algorithm, is used to rank features extracted for detecting intrusions and generate intrusion detection models. In this paper a new hybrid model RSC-PGA (Rough Set Classification Parallel Genetic Algorithm) is presented to address the problem of identifying important features in building an intrusion detection system, increase the convergence speed and decrease the training time of RSC. Tests are done on KDD-99 dataset used for The Third International Knowledge Discovery and Data Mining Tools Competition. Results showed that the proposed model gives better and robust representation of rules as it was able to select features resulting in great data reduction, time reduction and error reduction in detecting new attacks**.

***Keywords*— Intrusion detection, Parallel genetic algorithm, Rough set classification.**

I. INTRODUCTION

Intrusion detection is one of core technologies of computer security. The goal of intrusion detection is identification of malicious activity in a stream of monitored data which can be network traffic, operating system events or log entries. An Intrusion Detection system (IDs) is a hardware or software system that monitoring event streams for evidence of attacks. A majority of current IDs follow a signature-based approach in which, similar to virus scanners, events are detected that match specific predefined patterns known as “signatures". The main limitation of these signature-based IDs is their failure to identify novel attacks, and sometimes even minor variations of known patterns. Machine learning is a valuable tool for intrusion detection that offers a major opportunity to improve quality of IDs.

As a broad subfield of artificial intelligence, machine learning is concerned with the design and development of algorithms and techniques that allow computers to "learn". At a general level, there are two types of learning: inductive, and deductive. Inductive machine learning methods extract rules and patterns out of massive datasets. The major focus of machine learning research is to extract information from data automatically, by computational and statistical methods. We can use supervised learning in IDS for automatic generation of detectors without a need to manually update signatures. Generally, there are two types of detecting an intrusion; misuse detection and anomaly detection.

Wa'el M. Mahmud is a master student, department of computer science, university of Mansoura, Egypt. (E-mail: wael_mohesn@yahoo.co.uk).

Hamdy N.Agiza is Prof. of Applied Mathematics, department of mathematics, university of Mansoura, Egypt. (E-mail: agizah@mans.edu.eg).

Elsayed Radwan is doctor of Rough Sets and Artificial Intelligence Techniques (e-mail: radwan@intlab.toin.ac.jp).

In misuse detection, an intrusion is detected when the behavior of a system matches with any of the intrusion signatures. In the anomaly based IDs, an intrusion is detected when the behavior of the system deviates from the normal behavior.
IDs can be treated as pattern recognition problem or rather classified as learning system. Thus, an appropriate representation space for learning by selecting relevant attributes to the problem domain is an important problem for learning systems.
Feature selection is useful to reduce dimensionality of training set; it also improves the speed of data manipulation and improves the classification rate by reducing the influence of noise. The goal of feature selection is to find a feature subset maximizing performance criterion, such as accuracy of classification. Not only that, selecting important features from input data lead to a simplification of the problem, faster and more accurate detection rates. Thus selecting important features is an important problem in intrusion detection.

Rough Set Classification (RSC) [15], a modern learning algorithm, is used to rank the features extracted for detecting intrusions and generate intrusion detection models. RSC creates the intrusion (decision) rules using the reducts as templates. After reduct generation, the detection rules are automatically computed subsequently. The rules generated have the intuitive “IF-THEN” format, which is explainable and very valuable for improving detector design. The main feature of Rough Set data analysis is noninvasive, and the ability to handle qualitative data. This fits into most real life problems nicely and to our problem too. There are many attribute reduction algorithms but the most effective algorithm for large decision system reduction computation in practice is genetic algorithm [8]. This paper proposes a hybrid Parallel Genetic Algorithm (PGA) [9] based on the attribute significance heuristic rule to find minimal reducts. Proposed model uses parallel computation of the optimal rough set decision reducts from data by adapting the island model for evolutionary computing. This hybrid genetic algorithm is the key subalgorithm in the RSC algorithm. In our reduction experiments, we used the dataset [6] used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. These data are considered a standard benchmark for intrusion detection evaluations.

The rest of this paper is structured as follows. In the second section, is describing KDD-99 intrusion detection benchmark data briefly. Rough Sets preliminaries and some important definitions are listed in third section. Fourth section briefly describes Parallel Genetic Algorithms. Fifth section presents proposed RSC-PGA (Rough Set Classification - Parallel Genetic Algorithm) system model for rough set classification algorithm. Final section analyzes results and draw conclusions.

II. KDD-99 DATASET

One of the most important datasets for testing IDs is the KDD 99 intrusion detection datasets. KDD-99 [6][14] provides designers of IDs with a benchmark on which to evaluate different methodologies. This dataset is created by MIT Lincoln Lab's DARPA in the framework of the 1998 Intrusion Detection Evaluation Program [5].

In this paper, we used the subset that was preprocessed by the Columbia University and distributed as part of the UCI KDD Archive [6][14].

The dataset can be classified into five main categories which are Normal, Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probing.

- Denial of Service (DoS): Attacker tries to prevent legitimate users from using a service.
- Remote to Local (R2L): Attacker does not have an account on the victim machine, hence tries to gain access.
- User to Root (U2R): Attacker has local access to the victim machine and tries to gain super user privileges.
- Probe: Attacker tries to gain information about the target host.

For each TCP/IP connection record, 41 various quantitative and qualitative features were extracted plus 1 class label .The labeling of data features as shown in (Table I) is adopted from Chebrolu [3][10][12].

III. ROUGH SET THEORY PRELIMINARY

Rough sets theory was developed by Zdzislaw Pawlak in the early 1980's (Pawlak, 1982) [15]. It is a mathematical tool for approximate reasoning for decision support and is particularly well suited for classification of objects. Rough sets can also be used for feature selection, feature extraction.
The main contribution of rough set theory is the concept or reducts. A reduct is a minimal subset of attributes with the same capability of objects classification as the whole set of attributes. Reduct computation of rough set corresponds to feature ranking for IDs. Below is the derivation of how reducts are obtained.

Definition 1 An information system is defined as a four-tuple as follows, $S=<U, Q, V, f>$, where $U=\{x_1, x_2, \ldots, x_n\}$ is a finite set of objects (n is the number of objects); Q is a finite set of attributes, $Q=\{q_1, q_2, \ldots, q_n\}$; $V=\bigcup_{q \in Q} Vq$ and is a domain of attribute q; $f: U \times V \rightarrow V$ is a total function such that $f(x, q) \in Vq$ for each $q \in Q$, $x \in U$. If the attributes in S can be divided into condition attribute set C and decision attribute set D, i.e. $Q=C \cup D$ and $C \cap D=\Phi$, the information system S is called a decision system or decision table.

Definition 2 Let $IND(P)$, $IND(Q)$ be indiscernible relations determined by attribute sets P, Q, the P positive region of Q, denoted $POS_{IND(P)}(IND(Q))$ is defined as follows:

$$\text{POS}_{\text{IND(P)}}(\text{IND}(\text{Q}))=\bigcup_{X \in U/\text{IND}(Q)} \underline{\text{IND}(\text{P})}(X)$$

Definition 3 Let P, Q, R be an attribute set, we say R is a reduct of P relative to Q if and only if the following conditions are satisfied:

(1) $\text{POS}_{\text{IND(R)}}(\text{IND}(\text{Q}))=\text{POS}_{\text{IND(P)}}(\text{IND}(\text{Q}))$

(2) $\forall r \in R$ follows that

$$\text{POS}_{\text{IND(R-\{r\})}}(\text{IND}(\text{Q})) \neq \text{POS}_{\text{IND(R)}}(\text{IND}(\text{Q}))$$

Table I: Network Data Feature Label

Label	Network Data Features	Label	Network Data Features	Label	Network Data Features
A	Duration	***O***	Su_attempted	***AC***	Same_srv_rate
B	Protocol_type	***P***	Num_root	***AD***	Diff_srv_rate
C	Service	***Q***	Num_file_creations	***AE***	Srv_diff_host_rate
D	Flag	***R***	Num_shells	***AF***	Dst_host_count
E	Sec_byte	***S***	Num_access_files	***AG***	Dst_host_srv_count
F	Dst_byte	***T***	Num_cutbounds_cmds	***AH***	Dst_host_same_srv_rate
G	Land	***U***	Is_host_login	***AI***	Dst_host_diff_srv_rate
H	Wrong_fragment	***V***	Is_guest_login	***AJ***	Dst_host_same_src_port_rate
I	Urgent	***W***	Count	***AK***	Dst_host_srv_diff_host_rate
J	Hot	***X***	Sev_count	***AL***	Dst_host_server_rate
K	Num_failed_login	***Y***	Serror_rate	***AM***	Dst_host_srv_serror_rate
L	Logged_in	***Z***	Sev_serror_rate	***AN***	Dst_host_rerror_rate
M	Num_comprised	***AA***	Rerror_rate	***AO***	Dst_host_srv_rerror_rate
N	Root_shell	***BB***	Srv_rerror_rate		

Definition 4 Let $L= (U, A\cup \{d\}, V, f)$ be a decision system, whose discernibility matrix $\mathbf{M}(U) = [M_A^d(i, j)]_{n\times n}$ is defined as:

$$M_A^d(i,j) = \begin{cases} \{a_k \mid a_k \in A \wedge a_k(x_i) \neq a_k(x_j)\}, & d(x_i) \neq d(x_j); \\ \Phi, & d(x_i) = d(x_j). \end{cases}$$

Where $a_k(x_j)$ is the value of objects x_j on attribute a_k , $d(x)$ is the value of object x on decision attribute d. Write

$$\mathbf{M}(U) = [M_A^d(i,j)]_{n\times n} \text{ as a list } \{p1,\ldots, pt\}.$$

Each p_i is called a discernibility entry, and is usually written as $p_i=a_{i1}, \ldots, a_{im}$, where each a_{ik} corresponds to a condition attribute of the information system, $k=q,\ldots,m$; $i=1,\ldots,t$.

Furthermore, the discernibility matrix can be represented by the discernibility function f, conjunction normal form (CNF), i.e., $f=p_1\wedge\ldots\wedge p_t$, where each $p_i=a_{i1}\vee\ldots\vee a_{im}$ is called a clause, and each a_{ik} is called an atom. Note that the discernibility function contains only atoms, but not negations of atoms.

Although the discernibility matrix and discernibility function have different styles of expression, they are actually the same in nature.

Definition 5 let h denote any Boolean CNF function of m Boolean variables $\{a1,\ldots , am\}$, composed of n Boolean sums $\{s_1,\ldots , s_n\}$. Furthermore, let $w_{ij} \in \{0, 1\}$ denote an indicator variable that states whether a_i occurs in $\sum_{j=1}^{m} w_{ij} \times a_j$, $h = \prod_{i=1}^{n} s_i$. We can interpret h as a bag or multiset $\boldsymbol{M}(h) = \{Si \mid Si = \{a \in A \mid a_j \text{ occurs in } s_i\}\}$. Because the discernibility function f is also a CNF Boolean function, so it has a multiset. Let M(f) denote the multiset of discernibility function f, $\boldsymbol{M}(f) = \{\{a_{11},\ldots,a_{1m}\},\ldots,\{a_{i1},\ldots,a_{im}\},\ldots, \{a_{t1},\ldots,a_{tm}\}\}$.

Definition 6 Hitting set of a given multiset M of elements from 2^A is a set $B \subseteq A$ such that the intersection between B and every set in M is nonempty. The set $B \in HS(S)$ is a minimal hitting set of M if B ceases to be a hitting set if any of its elements are removed.

Let $HS(M)$ and $MHS(M)$ denote the sets of hitting sets and minimal hitting sets, respectively,

$$HS(M) = \{B \subseteq A \mid B \cap Si \neq \emptyset \text{ for all Si in M}\}$$

Proposition 1 For decision system $L= (U, A\cup \{d\},V, f)$, g is its discernibility matrix, and $B\subseteq A$, $B\in RED(U, d)$ is equivalent to $B\in MHS(M(g))$. So the rough set reduct computation can be viewed as a minimal hitting set problem.

Definition 7 The significance of attribute is defined as: $SGF(a, R, D)=p(a)$, $p(a)$ is the number of appearing times of attribute a in the remain part of the discernibility matrix which removes all the elements that have nonempty intersection with R.

IV. GENETIC ALGORITHM

Genetic algorithm (GA) is an adaptive heuristic search method for solving optimization problems. It was formally introduced in the United States in the 1970s by John Holland at the University of Michigan (Goldberg, 1989). They have a solid basis in genetics and evolutionary biological systems. GAs are comprising a kind of effective searching and optimizing technique that outperforms most of traditional methods.

In particular, GAs work very well on combinatorial problems such as reduct finding in rough set theory. Furthermore, finding the minimal reducts is a NP-hard problem [11]. Hence, GA is a good candidate as a methodology for finding minimal reducts.

In classical GA, individuals are encoded as binary strings of the attributes ((e.g. 0100110100 $\equiv\{a_2; a_5; a_6; a_8\}$). Each individual represents a set of attributes generated by mutation, crossover and selection procedures using some fitness criteria. Individuals with maximal fitness are highly probable to be reducts but there is no full guarantee.

PGA was first attempted by Grefenstette. Parallelism refers to many processors, with distributed operational load. Each GA is a good candidate for parallelization. Processor may independently work with different parts of a search space and evolve new generations in parallel. This helps to find out the optimum solution for the complex problems by searching massive populations and increases quality of the solutions by overcoming premature convergence. There are many types of Parallel Genetic Algorithm taxonomies [9].

One of the most ingenious taxonomies is the Island Model (IM) [11], where processors are globally controlled by

message passing within Master-Slave architecture. Master processor sends "START" signal to the slave processors to start generations and continue sending "MIGRATION" message to partially exchange the best chromosomes between the processors. So the worst chromosomes are replaced by the best received ones. Time between two consecutive MIGRATION signals is called the migration step; percentage of the best chromosomes is called migration percentage. Migrations should occur after a time period long enough for allowing development of good characteristics in each subpopulation.

V. RSC-PGA SYSTEM MODEL

Feature extraction and detection rules generation are two key steps in any intrusion detection system based on learning algorithm. Feature extraction depends on data source and the category of attack be detected. For detection rules auto generation, our proposed model uses rough set classification for this task. It includes three phases:

A. Preprocessing

The raw data are first partitioned into three groups of attacks [5][6][10]:

1. DoS attack detection dataset,
2. Probe attack detection dataset,
3. U2R&R2L attack detection dataset.

For each dataset, a decision system is constructed. Each decision system is subsequently split into two parts:

1. The training decision system,
2. The testing decision system.

B. Training decision system

Rough set classifier is trained on each training dataset of the three constructed decision systems of attacks. Each training dataset uses the corresponding input features and fall into two classes: normal (+1) and attack (−1). So this step has following steps:-

1. Apply the discretization strategies [7] on real values attributes to obtain a higher quality of classification rules.

   Equal-Width-Interval is used. It is a generic method that simply divides the data into some number of intervals all with equal width. It divides the number line between *Vmin* and *Vmax* into k intervals of equal width. Thus the intervals have width $w = (Vmax - Vmin) / k$ and the cut points are at $Vmin + w; Vmin + 2w; .....; Vmin + (k - 1) w$. k is a user predefined parameter and is set as 10 in this model. Algorithm has a time complexity of $O(n \log n)$ where n is the number of in generated intervals.

2. The intrusion (decision) rules are created using the reducts computed by the attribute reduction algorithm as templates.

There are many attribute reduction algorithms. Since our decision systems are large, we need effective algorithm for reduction computation. This paper proposes a hybrid Parallel Island Model (PIM) [11] for attribute reduction based on the attribute significance heuristic rule to find minimal reducts.
The idea is to optimize reducts within separate populations (islands) [11] and enable the best reducts chromosomes to migrate among islands. This hybrid PGA decreases the training time and makes the generated classifier more effective and it is adjusted to fit the intrusion detection environment.

We are modified PIM to run on single PC instead of running on many PCs connected with a network and we called this Singleton Parallel Island Model (SPIM). New technique uses distributed evolutionary computing to exploit availability of computers with multicore processors, the robust threading pools provided and supported by the Operating Systems, and massive power of parallel computing.

SPIM optimizes reducts within separate populations (islands) and enable the best reducts chromosomes to migrate among islands. The total population is divided into sub-populations evolving in parallel, which increases performance of calculations, and it is exploiting migration technique to exchange genetic material between populations, which increases quality along with performance. In addition, it decreases the training time and makes the generated classifier more effective.

Following steps describes how to adjust this Parallel Genetic Algorithm taxonomy to fit the intrusion detection environment.

A. Frame of Hybrid Genetic Algorithm

Finding the rough set minimal reduct is viewed as minimal hitting set problem [2][8]. For the discretized decision system $L = (U, A \cup \{d\}, V, f)$, a multiset is constructed according to the previous Definition 5, Section 3.

Subsequently, the hitting set of this multiset is computed using hybrid genetic algorithm.

Also a hitting set of a given multiset M of elements from 2^A is calculated according to the previous Definition 6, Section 3.

The heuristic method used in this model will maintain the capability of optimizing globally and can converge faster. This heuristic method is based on the significance of attribute *SGF (a, R, D)* that is defined in Definition 7, Section 3.

B. Representation (Generation of the Initial Population)

For the minimal hitting set problem, straightforward choice of population is a set P of elements from 2^A, encoded as bit-vectors [8], where each bit indicates the presence of a particular element in the set. This is called binary GA [13] which works with bits. The variable x has a value represented by a string of bits that is N_{gene} long. For example, assume that we have 41 condition attributes like in our case $\{a_1, a_2, \ldots, a_{41}\}$ and we have a reduct candidate as $\{a_1, a_4, a_6, a_9, a_{11}, a_{14}, a_{16}, a_{19}, a_{21}, a_{24}, a_{26}, a_{29}, a_{31}, a_{34}, a_{36}, a_{39}\}$. Then the reduct candidate should be represented as:
10010100101001010010100101001010010100100.

C. Function of Fitness

According to the definition of relative reducts, we know that the fitness function depends on the assumption: the number of attributes (which we wish to keep as low as possible) and the decision ability (which we wish to keep as high as possible) [2][8][13]. Our fitness function for decision system $L= (U, A \cup$

$\{d\}$, V, f) is defined as follows: Let n denote the number of condition attributes, M the multiset of discernibility function of L and $B \subseteq A$,

$$f(B) = \frac{n-|B|}{n} + \min\{\varepsilon, \frac{|\{S \in M | S \cap B \neq \emptyset\}|}{|M|}\}.$$

The first term rewards the shorter elements and the second tries to ensure that we reward sets that are hitting sets to guarantee the decision ability. The parameter ε controls the degree of approximation decision ability.

D. Selection and Recombination Method

SPIM is used at this point. The selection and recombination operator occurs are implemented with two steps [11]:

Step 1: Master thread start the operation

```
Input: number of threads N
Output: reducts
1: for i = 0 to N do
2:         SendMessage (i, START)
3: end for

4: while short enough reducts not found do
5:         for i = 0 to N do
6:          SendMessage (i, NEXTGENERATION)
7:         end for

8:         CollectReducts()

9:      if current generation such that migration time
                    reached then

10:                 for i = 0 to N do
11:                 SendMessage (i, MIGRATION)
12:                 end for
13:        end if
14: end while
```

Fig 1: Master thread operations

Step 2: Each thread operation

```
Input: chromosomes set (a1; a2; a3; ...; an)
Output: reductsList

1:  CalculateInputFitness()
2:  chromosomesList← ChooseBestOffSprings()

3:  for j ←0 to length [chromosomesList] do
4:         R ← {aj1; ...; ajn}

5:         for i ← 0 to n do
6:            if POS R-{an-ij} (d) = POSR (d) then
7:                  R ← R – {an-ij}
8:            end if
9:         end for

10:  end for
11:  return chromosomesList
```

Fig 2: Thread operations

In Step 1 in the algorithm (Fig. 1):

- Calculate the fitness for each chromosome in the current generation.
- Use heuristic rule to make genetic algorithm converge faster [8]. This rule operator operates on the whole population.
    - Let R be the attribute set represented by current chromosome. If R is not a hitting set (It is judged in the fitness function computation),
    - Then find an attribute a in $C-R$ which has the maximal value $SGF(a, R, D)=p(a)$.
    - If there are several a_j, (j=1,2,…,m;) with the same maximal value, stochastically choose one attribute from them.
    - Set the bit corresponding with aj as "1".
- Then according to the fitness for each chromosome; we use stochastic sampling method to select;

In Step 2 in the algorithm (Fig. 2):

- Let *minsingle*(*Offspring*) be the worst individual in the new population, *minfit*(*Offspring*) be the corresponding fitness;
- Let *maxsingle*(*Parent*) be the best individual in the old population, *maxfit*(*Parent*) be the corresponding fitness.
- If *minfit*(*Offspring*) < *maxfit*(*Parent*), we replace *minsingel*(*Offspring*) with *maxsingle*(*Parent*).

E. Crossover, Mutation and Inversion

We use classical, one point crossover [8][13]. Crossing over process affects chromosome selected for reproduction with probability of Pc. In the mutation process, we first select a chromosome to be mutated with probability *Pm* and then choose a single gene of the chromosome randomly. Mutation of a single gene means replacement of "1" by "0" or "0" by "1". Suppose that chromosome $S1$= {$s11$, $s12$, …, $s1r$, $s1,r+l$, $s1,r+l+1$,…, $s1n$}, where r, l are random numbers. $S2$ is the inversion of $S1$:$S2$= {$s11$, $s12$, …, $s1r$, $s1,r+l$, $s1,r+l+1$,…, $s1n$};

F. Testing Decision System

In this step model measure the performance of generated rules on testing data. So this step has following steps:-

1. Discretization method is first used to discretizing the new object dataset.
2. Generated rules are used to match testing objects to compute the strength of the selected rule sets for any decision class.
3. The new object will be assigned to the decision class with maximal strength of the selected rule set.

VI. EXPERIMENT SETUP AND RESULTS

In order to compare RSC-PGA algorithm with other techniques we constructed our intrusion detection system (RSC-PGA) and tested their performance on the KDD-99 intrusion detection contest dataset.

As described previously, we are using dataset subset that was preprocessed by the Columbia University and distributed as part of the UCI KDD Archive [6][14]. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted plus 1 class label .The labeling of data features as shown in (Table I) is adopted from Chebrolu [3][10][12].

The dataset can be classified into five main categories which are Normal, Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probing. The original data contained 744 MB data with 4,940,000 records. In the International Knowledge Discovery and Data Mining Tools Competition, only "10% KDD-99" dataset is employed for the purpose of training. So, all other experiments performed their analysis on the "10% KDD-99" dataset. In our experiments, we will use this "10% KDD-99" to compare it with other approaches used these data.

First of all, in Zainal and Zhang [8] RSC reducts obtained using standard Genetic Algorithms were 26 and they were : C, D, E, F, G, J, M, N, P, W, X, Y, AA, AB, AC, AD, AE, AF, AG, AH, AI, AJ, AK, AL, AM and AN. They had ranked the six most significant features using Rough Set Concept as: C, D, E, X, AF and AO. In addition, there are three different techniques namely Support Vector Decision Function (SVDF), Linear Genetic Programming (LGP) and Multivariate Adaptive Regression Splines (MARS) used by Sung and Mukkamala 1[1] were used to filter out redundant, superfluous information exist in these features, and hence significantly reduce a number of computer resources, both memory and CPU time, required to detect an attack. SVDF's proposed features were; B, D, E, W, X and AG. Meanwhile LGP yielded features C, E, L, AA, AE and AI. MARS suggested features E, X, AA, AG, AH and AI. Fig. 3, illustrates the comparison between results obtained using these techniques. In addition research of Neveen [12] used rough set theory as a reduction tool and feed forward neural networks as a learning tool has ranked: E, F, W, X, AF, AG, and AJ as the most important features.

Our RSC-PGA model has 22 reduct features, and has ranked five most significant features as C, D, E, X, and AO as the core of these reduct features. This result is compatible with other approaches results and at the same time contains the "E" reduct feature which has been chosen by other approaches as the most important reduct feature.

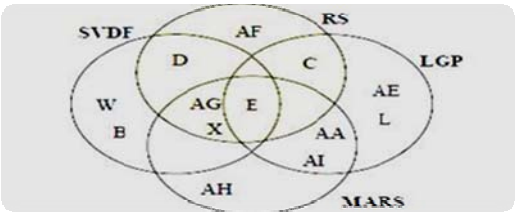


Fig 3: Comparison between different techniques

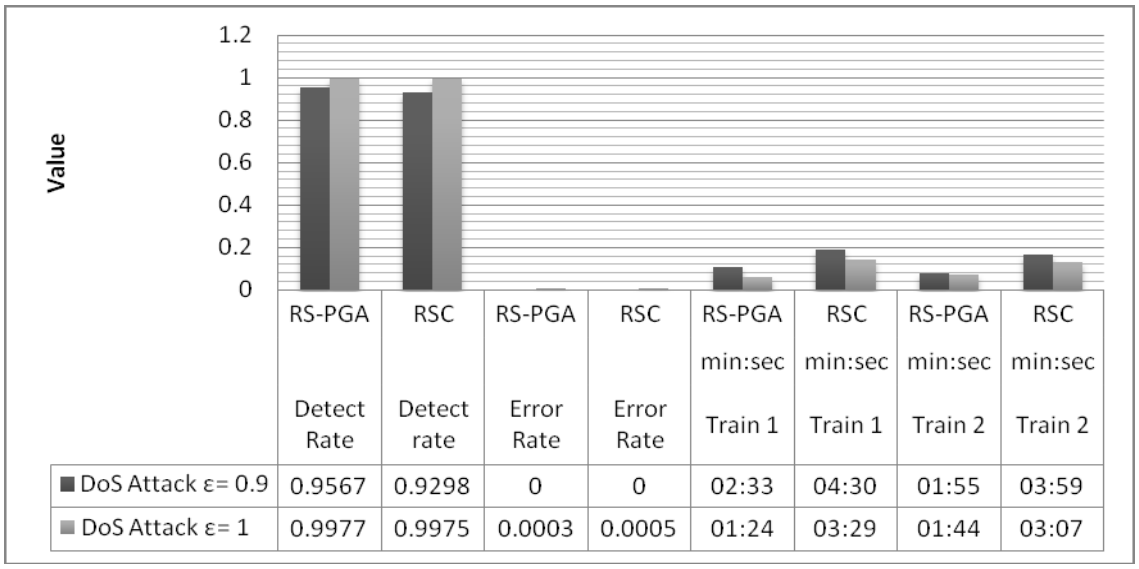
Besides this, we are compared the classification accuracy of each of these techniques against our RSC-PGA model. These results obtained by taking the average accuracy of applying our RSC-PGA model on the "10% KDD-99" dataset 20 times. Rows in (Table II) show the classification rate for each attack category and its classification accuracy. To simplify the analysis, we calculated the mean for all the four attack categories. Mean is important since it generalizes the overall performance of each feature subset when classifying the attack.

Another experiments result is shown in Fig.1~3 illustrates a comparison between RSC using standard Genetic Algorithms model introduced by Zhang [8], in against of our proposed model RSC-PGA that is use SPIM Genetic Algorithm over the different types of intrusions. The training time unit format is minutes: seconds. Training time 1 denotes the training time without using the heuristic rule in step 1 in algorithm 2 in used SPIM; training time 2 denotes the training time using this heuristic rule.

Table II: Comparison on the classification accuracy using RSC-PGA

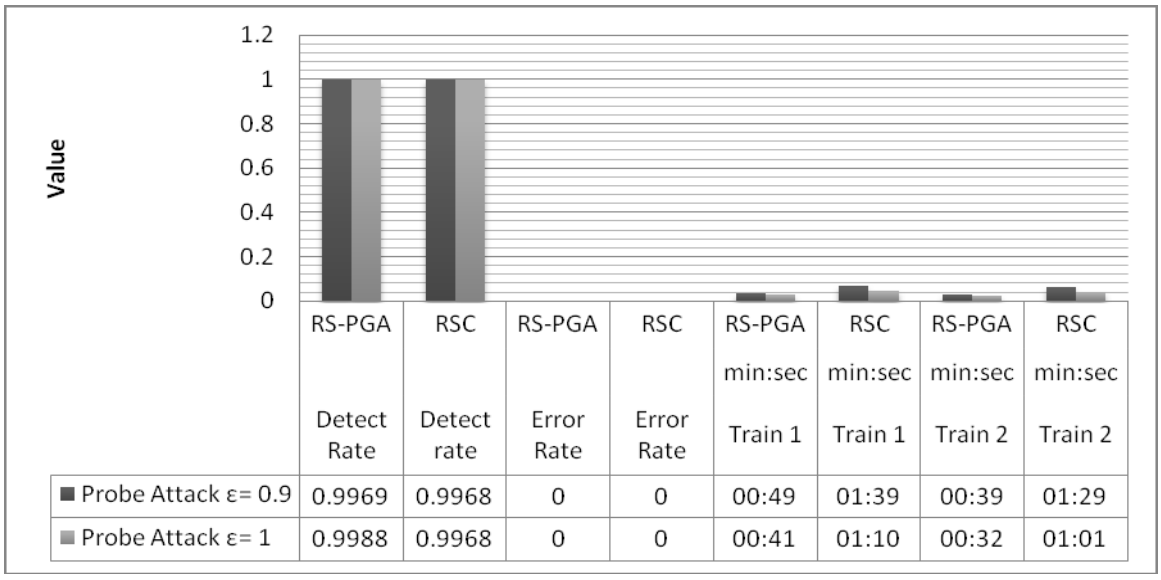
Type		MARS	SVDF	LGP	RSC	RSC-PGA
Normal		84.9	80.83	94.16	95.84	97.13
Attack	DoS	99.77	99.71	99.8	99.75	99.75
	Probe	100	100	100	99.68	99.98
	U2R	100	100	60	73.68	88.30
	R2L	100	100	100		
	Mean	96.934	96.108	90.792	92.23	96.29

ε value refers to the parameter ε used in the reduct computation. ε=1 means it is the accurately computed hitting set without approximation. From the above table, the heuristic rule decrease the training time in e. From these charts, we can conclude that RSC algorithm has detection performance level compatible with that of the other algorithms in terms of Probe and DoS attack detection (all above 99%). But for U2R&R2L attack detection, RSC algorithm is worse these algorithms.



	RS-PGA Detect Rate	RSC Detect rate	RS-PGA Error Rate	RSC Error Rate	RS-PGA min:sec Train 1	RSC min:sec Train 1	RS-PGA min:sec Train 2	RSC min:sec Train 2
DoS Attack ε= 0.9	0.9567	0.9298	0	0	02:33	04:30	01:55	03:59
DoS Attack ε= 1	0.9977	0.9975	0.0003	0.0005	01:24	03:29	01:44	03:07

Fig 4: DoS Attack Experiment Results for RSC-PGA



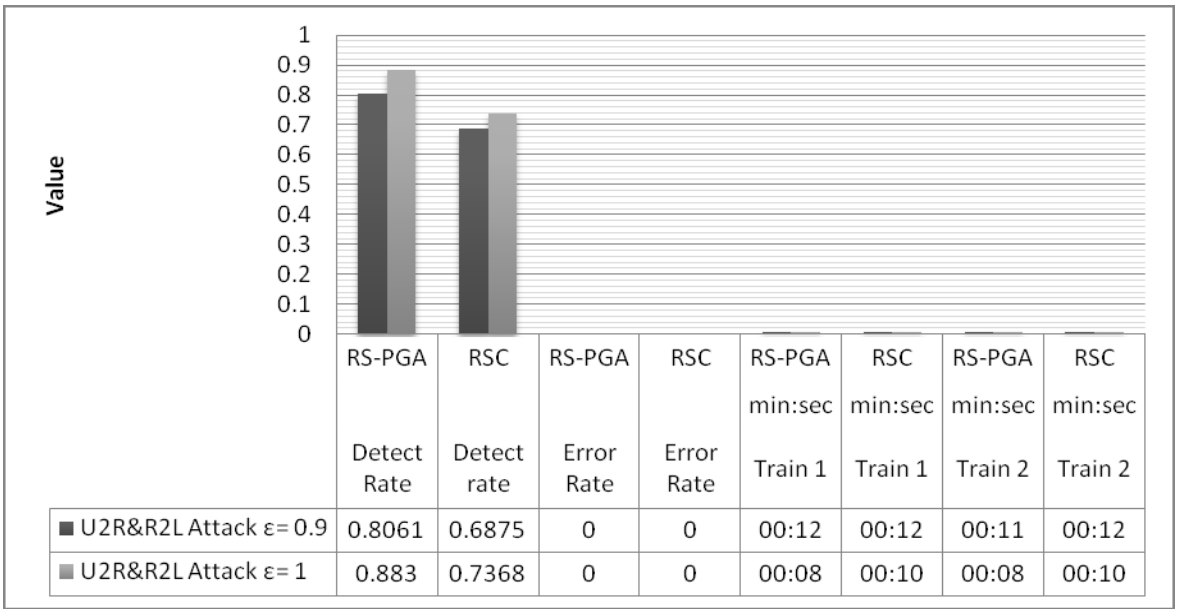
	RS-PGA Detect Rate	RSC Detect rate	RS-PGA Error Rate	RSC Error Rate	RS-PGA min:sec Train 1	RSC min:sec Train 1	RS-PGA min:sec Train 2	RSC min:sec Train 2
Probe Attack ε= 0.9	0.9969	0.9968	0	0	00:49	01:39	00:39	01:29
Probe Attack ε= 1	0.9988	0.9968	0	0	00:41	01:10	00:32	01:01

Fig 5: U2R & R2L Attack Experiment Results for RSC-PGA

The reason is that RSC algorithm can get good performance when the samples are enough while it performs a little worse for small attack sample case (In the DARPA dataset, U2R&R2L attack samples are low but DoS and Probe attack samples are enough). In contrast, SVM is a good tool that performs well for both small and enough sample attack cases. In addition, according to Kayacik and Heywood [3] it is not possible to achieve a high level of detection rate on attacks involving content (user to root and remote to local attacks).

Furthermore, the detection rules generated by the RSC-PGA algorithm, different from other techniques like MARS, SVDF, LGP, SVM [1][8], has the explainable "IF-THEN" format.

In addition, we can take advantage of information gain concept; that is underlying feature selection measure for constructing decision trees, to perform a feature relevance analysis to investigate the relevance of the 41 features with respect to dataset labels. Kayacik and Heywood [3] used this concept to determine the most relevant feature, which best discriminates the given class from the others. For given class, the feature with the highest information gain is considered the most discriminative feature.



	RS-PGA Detect Rate	RSC Detect rate	RS-PGA Error Rate	RSC Error Rate	RS-PGA min:sec Train 1	RSC min:sec Train 1	RS-PGA min:sec Train 2	RSC min:sec Train 2
U2R&R2L Attack ε= 0.9	0.8061	0.6875	0	0	00:12	00:12	00:11	00:12
U2R&R2L Attack ε= 1	0.883	0.7368	0	0	00:08	00:10	00:08	00:10

Fig 6: Probe Attack Experiment Results for RSC-PGA

VII. CONCLUSION

It is very valuable to get both high detection rate and explainable rules since this can improve our knowledge about the nature of the intrusion. In this paper we used Rough set Classification based on Parallel Genetic Algorithm (RSC-PGA) for IDs feature ranking and intrusion detection rules generation. Intrusion detection using RSC yields both explainable detection rules and high detection rate for attacks. Furthermore feature ranking using RSC for IDs is simple and fast.

We are modified Parallel Island Model (PIM) to run on single PC instead of running on many PCs connected with a network and we called this Singleton Parallel Island Model (SPIM). New technique uses distributed evolutionary computing to exploit availability of computers with multicore processors, the robust threading pools provided and supported by the Operating Systems, and massive power of parallel computing. SPIM Algorithm based on heuristic function increases performance of calculations, and its migration technique increases quality along with performance. In addition, it decreases the training time and makes the generated classifier more effective.

REFERENCES

[1] Anazida Zainal, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin, "*Feature selection using rough set in intrusion detection.*" IEEE TENCON 2006 Hongkong 14-17 November, 2006.

[2] Chinglai Hor, Peter A. Crossley, and Dean L. Millar, "*Application of genetic algorithm and rough set theory for knowledge extraction.*" IEEE Lausanne Volume, Issue 2007 Page(s):1117 – 1122.

[3] H. Güneş Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood, "*Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets.*" Third Annual Conference on Privacy, Security and Trust, October 2005.

[4] H. Sung, and S. Mukkamala, "*The feature selection and intrusion detection problems*". Springer Verlag Lecture Notes Computer Science 3321. 2004, Page(s): 468-482.

[5] Intrusion Detection Evaluation Program (http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html). June 2009

[6] Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, "*The 1999 DARPA off-line intrusion detection evaluation*" The International Journal of Computer and Telecommunications Networking ,Volume 34, Issue 4 (October 2000) Page(s): 579 – 595.

[7] Lixiang Shen, Francis E. H., "*A discretization method for rough sets theory*", Intelligent Data Analysis, Volume 5, Issue 5, October 2001, Pages: 431 - 438

[8] L. Zhang, G. Zhang, L. Yu, J. Zhang, and Y. Bai, "*Intrusion detection using rough set classification.*" Journal of Zheijiang University Science. 2004 5(9), pp. 1076-1086.

[9] Mariusz Nowostawski, Riccardo Poli, "*Parallel genetic algorithm taxonomy*" Knowledge-Based Intelligent Information Engineering Systems, 1999. Third International Conference Volume, Issue, Dec 1999 Page(s):88 – 92.

[10] Matthew V. Mahoney and Philip K. Chan, "*An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection.*" 6th International Symposium on Recent Advances in Intrusion Detection (September 2003).

[11] Mohammad M. Rahman1, Dominik Slezak, and Jakub Wroblewski, "*Parallel island model for attribute reduction.*" Lecture Notes in Computer Science. 2005.

[12] Neveen I. Ghali, "*Feature selection for effective anomaly-based intrusion detection.*" IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.3, March 2009.

[13] Randy L. Haupt, Sue Ellen Haupt, "*Practical genetic algorithms*" John Wiley & Sons, Inc, Second Edition, 2004.

[14] UCI KDD Archive (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html). June 2009

[15] Z. Pawlak, "*Rough sets: theoretical aspects of reasoning about data.*" Kluwer Academic Publishers, Netherlands. 1999.