# ***Intrusion Detection System Using Rough Set Classification Approach***

6th SEMESTER END-SEM PROJECT REPORT
FOR THE DEGREE OF

**BACHELOR OF TECHNOLOGY**
**IN**
**INFORMATION TECHNOLOGY**
**(B.Tech in IT)**

*Submitted by*


ChintadaMadhan Kumar (IIT2013106)
Tankala Harish (IIT2013118)
Ravula Keerthi Reddy (IIT2013171)
Pedakam Sneha (IIT2013176)
VangapalliTejasree (IIT2013203)


*Under the Guidance of :*
**Dr. ShekharVerma**
**(**Professor)
IIIT – Allahabad


Indian Institute of Information Technology
Allahabad

# CANDIDATES' DECLARATION

I hereby declare that the work presented in this project report entitled **"Intrusion Detection System Using Rough Set Classification"** submitted towards the completion of the 6th Semester of B.Tech(IT) at Indian Institute of Information Technology, Allahabad is an authenticated record of our original work carried out from January 2016 to May 2016 under the guidance of **Prof. Shekhar Verma.**

*(IIT2013106) CHINTADA MADHAN KUMAR*
*(IIT2013118) TANKALA HARISH*
*(IIT2013171) RAVULA KEERTHI REDDY*
*(IIT2013176) PEDAKAM SNEHA*
*(IIT2013203) VANGAPALLI TEJASREE*

# CERTIFICATE FROM SUPERVISOR

I do hereby declare that the mini project work prepared under my supervision by B.Tech group titled **"Intrusion Detection System Using Rough Set Classification"** be accepted in the fulfillment of the requirements of the mini project work of Bachelor of Technology in Information Technology, 6th semester. This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date : May 6, 2016
Place : Allahabad , IIITA

Dr.ShekarVerma
(Professor, IIITA)

_____

# ***ACKNOWLEDGEMENTS***

We would like to convey our deepest gratitude to Dr. Shekhar Verma, who guided us through this project. His keen awe-inspiring personality, superb guidance and constant encouragement are the motive force behind this projectwork.

We also owe a great deal of gratitude to the mini project evaluation committee members for their valuable insights that helps us to view the project in the new light.

Place : IIIT Allahabad
Date : May 6, 2016

Mini Project group
B.Tech IT
6th semester

*(IIT2013106) CHINTADA MADHAN KUMAR*
*(IIT2013118) TANKALA HARISH*
*(IIT2013171) RAVULA KEERTHI REDDY*
*(IIT2013176) PEDAKAM SNEHA*
*(IIT2013203) VANGAPALLI TEJASREE*

# ***ABSTRACT***

Intrusion detection is basically, the process of identifying and responding to suspicious activities targeted at computing and communication resources, and it has become the mainstream of information assurance as the dramatic increase in the number of attacks. Intrusion detection system (IDS) monitors and collects data from a target system that should be protected, processes and correlates the gathered information, and initiates responses when evidence of an intrusion is detected.

Intrusion Detection System (IDS) has been used as a vital instrument in defending the network from malicious or abnormal activities. It is still desirable to know what intrusions have happened or are happening, so that we can understand the security threats and risks and thus be better prepared for future attacks with the ability to analyze network traffic and recognize incoming and ongoing network attacks. Majority of network administrators has turned to IDS to help them in detecting anomalies in network traffic.

Machine learning based intrusion detection approaches are used to detect misuse and anomaly. In this project, Rough Set Classification (RSC), a modern machine learning algorithm is used to rank the features extracted for detecting intrusions and generate intrusion detection models. RSC performs feature ranking before generating rules and converts the feature ranking to minimal hitting setproblem addressed by using genetic algorithm (GA). In addition, a hybrid genetic algorithm is proposed to increase the convergence speed and decrease the training time of RSC. The models generated by RSC take the form of “IF-THEN” rules which have the advantage of explication.

# *TABLE OF CONTENTS*

# *1. INTRODUCTION*

**[1][2]**
Nowadays, there are lots and lots of attacks on the computer network system. As the Internet israpidly growing year by year, there are more vulnerabilities and threats for hacker and maliciouscracker to compromise network system especially through the Internet. So the best way to keep thecomputer network system secure is **implementing an Intrusion Detection System or [IDS]**.

We present the use of Rough Set Classification (RSC) for intrusion detection system (IDS) feature ranking and intrusion detection rules generation in this project. Intrusion detection using RSC can yield both explainable detection rules and high detection rate for some attacks namely (DoS and Probe Attacks) and feature ranking using RSC for IDS is simple and fast. RSC is one of the important contents of rough set theory. The main contribution of rough set to learning theory is the concept of reducts. A reduct is a minimal subset of attributes with the same capability of objects classification as the whole set of attributes. We use a fast hybrid genetic algorithm for the reduct computation of rough set. The reduct computation of rough set corresponds to feature ranking for IDS in RSC.

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 experiment data we will use isfrom MIT’s Lincoln Labs. It was developed for KDD (1999) competition by DARPA and is considered a standard benchmark for intrusion detection evaluations.

## *1.1 Problem Definition and Scope* :- [2]

Information systems and networks are subjected to electronic attacks. Attempts to breach information security are rising every day along with the availability of the vulnerabilities. Firewalls are put in place to prevent unauthorized access to the networks.

A common misunderstanding is that firewalls recognize attacks and block them. This is not true. Firewalls are simply a device or application that shuts off everything, then turns back on only a few well-chosen items.

The project focuses on extracting the features and generating the decision rules for intrusion detection system using Rough Set Classification approach.

In Intrusion Detection process, intrusion is happening or not will be decided by the connection record. A connection record is a sequence of TCP packets starting and ending at some well defined times between which data flows to and fro from a source IP address to a target IP address under some well defined protocols. Each connection sequence has 41 features and by using some of these features, we can decide whether intrusion is happening or not.

Using these important features, we generate explainable decision rules.Decisionruleswillexplain that on presence of whatkind of features and value is a reason for intrusion andthe intrusion detector design system will ensure that this kind of situations will not arise or minimize the intrusion happening probability.

## *1.2 Goal*

- To detect an intrusion as it happens and be able to respond to it.
- Designing and implementing a Rough Set Classification (RSC) and using RSC we are going to get intrusion detection system feature ranking and generate intrusion detection rules.

## *1.3 Motivation*

- Firewalls are mostly employed at the boundary of the network and yet a greater percentage of potential hackers may be from within the network.
- IDS's can monitor both internal and external attacks.
- IDS's can analyse the vulnerability of a network.

# 1.4 LITERATURE SURVEY

| S.NO | TITLE | YEAR | JOURNAL / CONFERENCE | OBJECTIVE | CHALLENGE(S) DEALT |
|---|---|---|---|---|---|
| 1. | Current studies on Intrusion Detection System, Genetic Algorithm and Fuzzy Logic**[1]** | 2013 | International Journal of Distributed and Parallel Systems (IJDPS) | This paper presents different approaches being employed in intrusion detection system and types of attacks being detected. | The analysis tells about the novel attacks by unauthorized users in network traffic |
| 2. | A Hybrid Intrusion Detection System design for computer network security**[2]** | 2009 | Research Fund of Istanbul University, Turkey | This paper lists out the limited functionality of the firewalls | After analysing this paper the need for IDS is well understood |
| 3. | First International Workshop on the Recent Advances in Intrusion Detection **[3]** | 1998 | Workshop report, Workshop held in Louvain-la-Neuve | This paper lists out two different categories of intrusion detection models | Through the analysis of this paper we understood the models i.e., misuse detection and anomaly detection |
| 4. | An Overview on Intrusion Detection System and Types of Attacks It Can Detect Considering Different Protocols**[4]** | 2012 | International Journal of Advanced Research in Computer Science and Software Engineering | Here, this paper tells about all the different types of protocol attacks and analysis | This deals in detail about the individual protocol attack and the analysis like traffic data and effectiveness of anomaly detection |
| 5. | How Re(Pro)active Should an IDS be?**[5]** | 2014 | International Centre for Security Analysis King's College London publications | This paper shows how the classical security paradigm of protect, detect and react has been applied to the field of Information Security with Firewalls. | After analysing this paper, we understood that a genuine intrusion incident has actually occurred can sometimes be extremely difficult. |
| 6. | Next Generation Intrusion Detection Systems (IDS)**[6]** | 2002 | McAfee Network Security Technologies Group publications | The paper aims at understanding IDS, the need for IDS and the challenges dealt today | Challenges to be faced while developing the IDS are well noted |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | through this analysis |
| 7. | Survey: Learning Techniques for Intrusion Detection System (IDS) **[7]** | 2014 | International Journal of Advance Foundation and Research in Computer (IJAFRC) | This research paper dealt with the drawbacks of the existing technologies and a better approach is proposed | The proposed new approach is dealt in this project |
| 8. | Reduct Generation from Binary Discernibility Matrix: An Hardware Approach**[8]** | 2012 | International Journal of Future Computer and Communication | The aim is to understand the Rough set theory which is a powerful mathematical tool used for extracting useful rules from a huge database | Analysis brings to an idea of implementing the basic concepts of Rough set theory |
| 9. | A comparative study on the currently existing intrusion detection systems | 2009 | International Association of Computer Science and Information Technology - Spring Conference | In this paper, almost all the noticeable IDS s proposed so far and focussed on each of them respectively. | Learnt some important factors that should come into issue while designing an efficient intrusion detection system. |
| 10. | An implementation of intrusion detection system using genetic algorithm | 2012 | International Journal of Network Security & Its Applications (IJNSA) | In this paper, the implementation of an Intrusion Detection System by applying genetic algorithm to efficiently detect various types of network intrusions is learnt. | We have learnt that if we can use a better equation or heuristic in this detection process we believe the detection rate and process will improve a great extent. |
| 11. | Network intrusion detection system on machine learning algorithms | 2010 | International Journal of Computer Science & Information Technology (IJCSIT) | An intrusion detection method using an SVM is proposed. | Performance analysis is observed. |

## *1.5 CURRENTLY EXISTING TECHNOLOGIES*

**1. Wavelet Clustering Based Intrusion Detection :-**

Wavelet clustering was first proposed by Gholamhosein which was a grid based algorithm and can be effectively used in the image recognition. Wu and Yao et al extended the application of

wavelet clustering into the information safety field and proposed an intrusion detection technology based on wavelet clustering to realize the two class intrusion detection.

**2. Decision Tree Techonology Based**:-

Decision tree technology is an intuitionistic and straightforward classification method. It has great advantage in extracting features and rules. Therefore applying decision tree technology into intrusion detection is of great significance.

**3. SVDD(supportvector data description) Based :-**

Application of SVDD to the idea of intrusion detection system was proposed to address the problem of ever-increasing amount of network data. Tax and others proposed SVDD is intended to address the issue of one- threshold classification method of support vector machines. One-threshold classification is a special kind of classification problem, the problem to be solved is distinguishing a target class from all other types of data which does not belong to the target class type (named as outlier type). SVDD algorithm optimizes the problem of the detection rate, false alarm rate and missing rate in intrusion detection system.

## *1.6 Analysis of previous research in this area*

### *1. Firewalls [1][2][5]*

A common misunderstanding is that firewalls recognize attacks and block them. This is not true. Firewalls are simply a device or application that shuts off everything, then turns back on only a few well-chosen items. In a perfect world, systems would already be "locked down" and secure, and firewalls will not be needed. The reason we have firewalls is precisely because security holes are left open accidentally.

Thus, while installing the firewall, the first thing it does is, stops ALL communication. The firewall administrator then carefully adds "rules" that allows only specific types of traffic to go through the firewall.

A firewall is simply a fence around the network with a couple of well-chosen gates. A fence has no capability of detecting somebody trying to break in (such as digging a hole underneath it) nor does a fence know if somebody coming through the gate is allowed in. It simply restricts access to the designated points.

In summary, a firewall is not the dynamic defensive system that users imagine it to be. In contrast, IDS is much more of that dynamic system. An IDS does recognize attacks against the network that firewalls are unable to see.

Another problem with firewalls is that they are only at the boundary of the network. Roughly 80% of all financial losses are due to the hacking that come from inside the network. A firewall at the perimeter of

the network sees nothing going on inside, it only sees the traffic which passes between the internal network and the Internet.

The Intrusion detection system in a similar way complements the firewall security. The firewall protects an organization from malicious attacks from the Internet and the intrusion detection system detects if someone tries to break in through the firewall or manages to break in the firewall security and tries to have access on any system in the trusted side and alerts the system administrator in case there is a breach in security.

## *2. Intrusion detection system*

In the research of feature extraction in intrusion detection,
Wenke (1999) used improved **Apriori algorithm** to acquire features of network connection level. This method is very effective. Apriori is a classic algorithm for frequent itemset mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

## *3.Machine Learning Approach for Intrusion Detection*

Intrusion Detection system could not distinguish between normal and abnormal behavior of system using the audit data due to the ineffective behavior model system. Thus has to rely on the human for detection of the behavior. Involvement of Human in the detection system reduces the performances of the IDS as it become the tedious job with increasing data and network traffic. Due to the above deficiencies of IDSs based on human experts, intrusion detection techniques using machine learning have attracted more and more interests in recent years. Machine learning is based heavily on statistical analysis of data and some algorithms can use patterns found in previous data to make decisions about new data.

## *4. Intrusion Detection System Using Neural Network*

Applying the Neural Network (NN) approach to Intrusion Detection, we first have to expose NN to normal data and to attacks to automatically adjust coefficients of the NN during the training phase. Performance tests are then conducted with real network traffic and attacks. In order to apply this approach to Intrusion Detection, we have to introduce data representing attacks and non-attacks to the Neural Network to adjust automatically coefficients of this Network during the training phase. In other words, it will be necessary to collect data representing normal and abnormal

behavior to train the Neural Network. After training is accomplished, a certain number of performance tests with real network traffic and attacks were to be conducted.

## *5. Intrusion Detection System Using Support Vector Machine*

Support Vector Machines or SVMs, are learning machines that plot the training vectors in high-dimensional feature space, labeling each vector by its class. SVMs view theclassification problem as a quadratic optimization problem. They combine generalization control with a technique to avoid the "curse of dimensionality" by placing an upper bound on the margin between the different classes, making it a practical tool for large and dynamic data sets. SVMs classify data by determining a set of support vectors which are members of the set of training inputs that outline a hyper plane in feature space. The primary advantage of SVMs is binary classification and regression that they provide to a classifier with a minimal dimension which implies low expected probability of generalization errors.

There are two main reasons that we experiment with SVMs for intrusion detection-

1)The first is speed as real-time performance is of primary importance to intrusion detection system. Any classifier that potentially outruns neural networks is worth considering.

2)The second reason is scalability:- SVMs are relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space , so they can potentially learn a larger set of patterns and be able to scale better than neural networks.

## *6. Machine Learning Based Anomaly Detection Techniques*

### *i. Bayesian Networks*

Bayesian methods provide a probabilistic approach to learning. They combine prior knowledge of probability distributions of the candidate hypotheses with the observed data to determine the posterior probability of target hypotheses. Thus they can be applied inherently to problems whose output requires probabilistic predictions. They also provide a framework for analyzing the bias of other algorithms that do not deal directly with probabilities. The naive Bayes classifier is an effective algorithm that uses Bayesian reasoning.

### *ii. Genetic Algorithm*

Genetic algorithm is a family of computational models based on principles of evolution and natural selection. These algorithms convert the problem in a specific domain into a model by using a chromosome-like data structure and evolve the chromosomes using selection, recombination, and mutation operators. The process of a genetic algorithm usually begins with a randomly selected

population of chromosomes. These chromosomes are representations of the problem to be solved. The set of chromosomes during a stage of evolution are called a *population*. An *evaluation function* is used to calculate the —goodnessof each chromosome. Applying genetic algorithm to intrusion detection seems to be a promising area. Genetic algorithms can be used to evolve simplerules for network traffic. These rules are used to differentiate normal network connections from anomalous connections.

## *1.7 Observations from the above analysis :-[7]*

In the research of detection model generation, it is desirable that the detection model be explainable and have high detection rate but the existing methods cannot achieve these two goals.
Neural Networks (James, 1998) could achieve high detection rate but the detection rules generated are not explainable. Decision trees (Wenke, 1999) could yield explainable rules but the detection rate is low. And SVM needs many iterations and is very time-consuming.

## *1.8 Our implementation to overcome the above mentioned drawbacks*

As we have seen above, there are some of the drawbacks in the exisiting systems. So we will try to overcome these drawbacks by using the Rough Set Classification approach which is explainable and also has high detection rate in intrusion detection system to generate decision rules to classify the attack data and the normal data.

## *Table 1:- Advantages Of RSC Intrusion Detection System*

| Solutions | Firewalls | Anti-Viruses | Intrusion detection System |
|---|---|---|---|
| **Ability to control internal attacks** | No | Limited way | Yes |
| **Ability to log intrusion attempts** | No | Limited way | Yes |
| **Ability to provide a history for attacks** | No | Yes | Yes |
| **Ability to send alerts** | No | No | Yes |
| **Ability to detect attacks** | No | Limited way | Yes |
| **Ability to reacting to attacks** | No | Limited way | Yes |

## *1.9 Formulation of the present problem :-*

The goal is that to design Rough Set Classifier for feature extracting and decision rules generation for intrusion detection system. Designing an intrusion detection system based on learning algorithm can be described in the following steps:

1. Capture network data.
2. Process these data into suitable input format.
3. Normalize the network flow and extract features of attack behavior or normal usage patterns from raw data.
4. Design and use learning algorithm to get detection rules.
5. Use these detection rules on test data.

We choose the 1999 KDD Intrusion Detection contest dataset to design our system.

For each TCP/IP connection, 41 various quantitative and qualitative features were extracted. The following three main feature sets can be used to classify each connection.

**1. Intrinsic features** i.e., general information related to the connection. They include the duration, type, protocol, flag, etc. of the connection.

**2. Traffic features** i.e., statistics related to past connections similar to the current one, e.g., number of connections with the same destination host or connections related to the same service in a given time window or within a predefined number of past connections.

**3. Content features** i.e., features containing information about the data content of packets that could be relevant to discover an intrusion, e.g., errors reported by the operating system, root access attempts, etc.

For detection rules auto-generation, we present the use of rough set classification for this task.
It includes three phases:**[7]**

1. **Preprocessing:-** The raw data is first partitioned into three groups: DoS attack detection dataset, Probe attack detection dataset, U2R&R2L attack detection dataset. For each dataset, a decision system is constructed. Each decision system is subsequently split into two parts: the training dataset and the testing dataset.
2. **Training:-** Rough set classifier is trained on each training dataset of three different types of attacks (DoS, Probe, U2R&R2L). Each training dataset uses the corresponding input features and fall into two classes: normal (0) and attack (1).

3. **Testing:-** Measure the performance on testing data.

## 2. Description of Hardware and Software :-

### 2.1 Hardware

- For this project we are using Window 8.1 operating system installed computer and it has 4GB RAM and Intel core i7 2.30 GHz processor.
- Monitor
- Keyboard

### 2.2 Software And Tools

- Code Block C++ IDE (Integrated Development Environment) or Devcpp
- C++ language
- Any text editor

## 3. Theoretical Tools – Analysis and Development :-

### 3.1 Definitions: - [8][9]

Rough sets theory was developed by ZdzislawPawlak in the early 1980's (Pawlak, 1982).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 of 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.

## Definition 1 :-

An information system is defined as a four-tuple as follows,
S=<U, Q, V, f>where
U={x1, x2, …,xn} is a finite set of objects
n is the number of objects.
Q is a finite set of attributes, Q={q1, q2, …, qn}.
$V = \bigcup_{q \in Q} Vq \; Uq \in QVq$ and Vq is a domain of attribute q.
f:U×V→V is a total function such that f(x, q) ∈Vq for each q∈Q, x∈U.

If the attributes in S can be divided into condition attribute set C and decision attribute set D then Q=C∪D and C∩D=Φ, 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:

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

## ***Definition 3 :-***

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

(1) POS IND(R) (IND( Q))= POS IND(P) (IND( Q))
(2) ∀r ∈ Rfollows that
POS IND(R-{r}) (IND( Q)) ≠ POS IND(R) (IND( Q))

## ***Definition 4:-***

Let L is a decision system - L = {U, AU{d} ,V,F}

Whose discernibility matrix is defined as –

$$\mathbf{M}(U) = [M^{d}_{A}(i , j)]n\times n$$

isdefined as

$$M(U) = \{ \begin{matrix} ak \mid ak\in A \cap ak(xi) \neq ak(xi)\} & d(xi) \neq d(xj) \\ \emptyset d(xi)= d(xj) \end{matrix} \}$$

whereak(xj) is the value of objects xjon attribute ak, d(x) is the value of object x on decision attribute d. Write **M**(U) = [MAd (i , j)]n×n as a list {p1,…, pt}.

Each $p_i$ is called a discernibility entry, and is usually written as pi=ai1, …,aim, where each aikcorresponds to a condition attribute of the information system, k=q,…,m; i=1,…,t.

Discernibility matrix can be represented by the discernibility function f, f=p1∩…∩pt, where each pi=$a_{i1}$∪…∪$a_{im}$is called a clause. 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 :-***

Lethdenote any Boolean CNF function of m Boolean variables {a1,… , am}, composed of n Boolean sums{s1,… , sn}. Furthermore, let wij ∈ $\{0,1\}$denote an indicator variable that states whether ai occurs in $\sum_{i=1}^{m} w_{ij} * a_i$,

$$h = \prod_{j=1}^{n} s_j.$$

We can interpret h as a bag or multiset

**M** (h) = {Si |Si = {a ∈A| ajoccurs in si}}

Because the discernibility function f is a CNF Boolean function, so it has a multiset. Let M(f) denote the multiset of discernibility function f,

**M**(f) = {{a11,…,a1m},…,{ai1,…,aim},…, {at1,…,atm}}.

## ***Definition 6:-***

Hitting set of a given multisetMof elements from 2A is a set$B \in A$such that the intersection between B and every set in M is nonempty. The set$B \in HS(S)$ is a minimalhitting 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 :-***

An approximate hitting set is a set that hits “enough” elements of the bag or multiset*M*. The approximate hitting set provides an approximate solution to the hitting set problem. The set of ∈-approximatehitting sets of the multiset *M* is denoted

$$AHS(M, \in) = \left\{ B \subseteq A \middle| \frac{|\{SinM | S \cap B \neq \emptyset\}}{|M|} \geq \in \right\}$$

where the parameter ∈controls the degree of approximation. The set is a minimal ∈-approximate hitting set if it ceases to be so if any of its elements are removed. The set of all minimal ∈-approximate hitting set is denoted MAHS(M, ∈). Computing all elements of MAHS(M, ∈) is computationally intractable, and heuristics are needed (Wroblewski, 1995). In our model a heuristic rule based on the significance of attribute is applied to search for solutions.

## ***Definition 8 :-***

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.

## *3.2 Genetic Algorithm [1]*

Genetic Algorithm (GA) is an effective searching and optimizing algorithm used to apply in various fields.GA works efficiently in combinational problems such as reduct finding in rough set theory and finding a minimal reduct is an NP- hard problem. In classical GA, individuals are encoded as binary strings of the attributes. 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. A GA starts by generating a large set of possible solutions to a given problem. It then evaluates each of those solutions and decides on a "fitness level". The "fitness level" is represented by fitness function in GA for each solution set. These solutions then generate new solutions. The parent solutions that have better "fitness level" are more likely to reproduce while those that have less "fitness level" are more unlikely to do so. In essence, solutions are evolved over time on generation. This way GA's evolve the search space scope to a point where the solution can be found.

### ***General Genetic Algorithm processing steps:-***

**Step 1 ) Initial Population Generation :-**

Initial population is generated using randomly selecting some section of solution.

**Step 2 ) Fitness Evolution :-**

Calculate fitness for each selected solution.This fitness tells us that how much this solution is near to the real solution.These solutions are not our problem solutions but they take us near to the real solution.

**Step 3) Reproduce,selection Mutation and Crossover :-**

From all the population we select the solution which has the maximum fitness value because these solutions are most likely to reproduce the offspring for new generation .So select two maximum fitness solutions as a parents. Offspring is generated by parent solution, so they have the properties of both the parents so we use crossover and mutation process to make it possible.Offsprings havinggeans from both parents is called as crossing over.

**Step 4) Control Next Generation:-**

If the solution generated by the iterations is more near to the desired answer then we stop reproducing the next generation.Otherwise we continuously repeat the above 3 steps until we getthe desired solution.

## *3.3 KDD99 Data Set [1]*

To implement the algorithm and to evaluate the performance of the system, we propose the standard datasets employed in KDD Cup 1999 "Computer Network Intrusion Detection" competition. The KDD 99 intrusion detection datasets depends on the 1998 DARPA proposal which offers designers of intrusion detection system with a standard on which to evaluate different methodologies. Hence a simulation is being prepared from a fabricated military network with three 'target' machines running various services and operating systems. They also applied three extra machines to spoof different IP addresses for generating network traffic. A connection is a series of TCP packets beginning and ending at some well defined periods between which data floods from a source IP address to a target IP address under some well defined protocol.Each network connection is labelled as normal (normal) or abnormal (attack). It results in 41 features for each connectionand one target value. Target value indicates the attack category name. Attacks are therefore categorized into on of four types :
Denial of Service attack, Remote to Local, User to Root and Probe. **[1][4]**

***DoS***

A DoS attack is a type of attack in which the unauthorized users build a computing or memory resources too busy or too full to provide reasonable networking requests and hence denying users access to a machine.

E.g ping of death, neptune, back, smurf, apache, UDP storm, mail bomb are all DoS attacks.

***U2R***

A remote to user (U2R) attack is an attack in which a user forwards networking packets to a machine through the internet, which he/she doesnot have right of access in order to expose the machine vulnerabilites and exploit privilages which a local user would have on the computer.

E.g guest, xlock, xnsnoop, sendmail dictionary, phf etc.

***R2L***

A R2L attacks are regarded as the exploitations in which the unauthorized users start off on the system with a normal user account and tries to misuse vulnerabilites in the system in order to achieve super user access rights.
E.gxterm, perl.

***Probe***

A probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to negotiate the system.
E.gportsweep, saint, mscan, nmap

KDD99 data set for each connection which describes the 41 features : -

2, tcp, smtp, SF, 1684, 363, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 104, 66, 0.63, 0.03, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, normal.

0, tcp, private, REJ, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 38, 1, 0.00, 0.00, 1.00, 1.00, 0.03, 0.55, 0.00, 208, 1, 0.00, 0.11, 0.18, 0.00, 0.01, 0.00, 0.42, 1.00, portsweep.

0, tcp, smtp, SF, 787, 329, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 76, 117, 0.49, 0.08, 0.01, 0.02, 0.00, 0.00, 0.00, 0.00, normal.

The above data set of three records is written in the CSV format, plus the final mark (label), a total of 42, first 41 of which were classified into four categories to explain the meaning of each characteristics in the following sequence : -

***1. TCP connection to the basic characteristics (nine kinds)***

Basic connectivity features include some of the basic connection properties, such as continuous - time, protocol type, the transmission of the number of bytes.

(1) **Duration**: Duration of connection, in seconds as a unit, a continuous type. The range is [0, 58329]. It is defined as the connection from the TCP three way handshakes to establish the date, to the FIN / ACK to connect up to the end of time; for the UDP protocol type, then each UDP packet as a connection. Data set the duration, = 0, because the duration of the connection of that section is less than one second.

(2) **Protocol Type:** Protocol type, discrete types, a total of three kinds: TCP, UDP, and the ICMP.

(3) **Service:** network service type of the target host, the discrete type, a total of 70 kinds.
'Aol', 'auth', 'bgp', 'courier', 'csnet_ns', 'ctf','daytime', 'discard','domain','domain_u','echo','eco_i',

‘ecr_i’,‘efs’, ‘exec’, ‘finger’, ‘ftp’, ‘ftp_data’, ‘gopher’, ‘harvest’,‘hostnames’, ‘http’, ‘http_2784’, ‘http_443’,‘http_8001’, ‘imap4’, ‘IRC’, ‘iso_tsap’, ‘klogin’, ‘kshell’, ‘ldap’, ‘link’, ‘login’, ‘mtp’, ‘name’, ‘netbios_dgm’, ‘netbios_ns’, ‘netbios_ssn’,‘netstat’, ‘nnsp’, ‘nntp’, ‘ntp_u’, ‘other’, ‘pm_dump’, ‘pop_2’, ‘pop_3’, ‘printer’, ‘private’, ‘red_i’, ‘remote_job’, ‘rje’, ‘shell’, ‘smtp’, ‘sql_net’, ‘ssh’, ‘sunrpc’, ‘supdup’, ‘systat’, ‘telnet’, ‘tftp_u’, ‘tim_i’, ‘time’, ‘'urh_i’, ‘urp_i’, ‘uucp’, ‘uucp_path’, ‘vmnet’, ‘whois’, ‘X11’, ‘Z39_50’.

(4) **The Flag:** connected properly or the wrong state, discrete types, a total of 11 kinds. ‘OTH’, ‘REJ’, ‘RSTO’, ‘RSTOS0’, ‘RSTR’, ‘S0’, ‘S1’, ‘S2’, ‘S3’, ‘SF’, ‘SH’. It indicates that the connection is in accordance with the requirements in the agreement to start orcomplete. Such as SF connected properly create and terminate; the S0 only received aSYN request packet, but not the back of the SYN / ACK Where SF is the normal.
The other 10 kinds of error are:

(5) **Src_Bytes:** the number of bytes of data from the source host to the target host, continuous type, range [0, 1379963888].

(6) **Dst_Bytes**: the number of bytes of data from the target host to the source host, continuous type, range [0.1309937401].

(7) **Land**: if the connection is from / served on the same host / port is 1, and 0 otherwise, discrete type, 0 or 1.

(8) **Wrong_Fragment:** number of error segments, continuous type, range [0, 3].

(9) **Urgent:** number of urgent packages, continuous type, range [0, 14].

***2. TCP Connection Characteristics (13 species)***

(10) **Hot:** the number of access to sensitive files and directories, continuous, range [0, 101]. Such as access to the system directory, the establishment or implementation of the program.

(11) **Num_Failed_Logins:** the number of failed login attempts, continuous, range [0, 5].

(12) **Logged_In:** successful login was 1, and 0 otherwise, discrete, 0 or 1.

(13) **Num_Compromised:** number of compromised conditions, continuous, range [0, 7479].

(14) **Root_Shell:** root_shell means to obtain super user privileges, if root shell is 1, and 0 otherwise,discrete, 0 or 1.

(15) **Su_Attempted:** if "su root" command is 1, and 0 otherwise, discrete, 0 or 1.

(16) **Num_Root:** root user visits, continuous, range [0, 7468].

(17) **Num_File_Creations**: the number of file creation operations, continuous, range [0, 100].

(18) **Num_Shells:** number of shell commands, continuous, range [0, 5].

(19) **Num_Access_Files:** the number of access control file, continuous, range [0, 9]. For example: the /etc/passwd.

(20) **Num_Outbound_Cmds:** an FTP session outbound connections, number of consecutive 0. The data set of this feature the number of occurrences of 0.

(21) **Is_Hot_Login:** 1 if the login belongs to the “hot” list, 0 otherwise, discrete, 0 or 1, such as the super user or administrator login.

(22) **Is_Guest_Login** if the guest login is 1, and 0 otherwise, discrete, 0 or 1.

***3. Time-based network traffic statistical characteristics (nine kinds)***

As network attacks have a strong correlation in time, the statistics of the currentconnection record connection between the records within the period before thecontact, can better reflect the connection between the relationship. These characteristics are divided into two collections: One is the "same host" features observed only in the last two seconds with the current connection to the same targethost connection, for example, the same number of connections, these same connectionwith the current connection the connection of services, etc. the other is the "same service" features only observed over the last two seconds with the current connectionwith the same service connection, such as the number of such connections, includingthe number of SYN errors or REJ errors.

(23) **The_Count:** number of connections to the same host as the current connection in the past two seconds, continuous, range [0, 511]. Note: The following features refer to these same - host connections.

(24) **Srv_Count:** number of connections to the same service as the current connection in the past two seconds, continuous, range [0, 511]. Note: The following features refer to these same - service connections.

(25) **Serror_Rate:** within the past two seconds, in connection with the current connection with the same target host, "SYN" errors the percentage of connected, continuous, range [0.00, 1.00].

(26) **Srv_Serror_Rate:** over the last two seconds, in connection with the current connection with the same service, "SYN" errors the percentage of connected, continuous, range [0.00, 1.00].

(27) **Rerror_Rate:** within the past two seconds, in connection with the current connection with the same target host, "REJ" errors the percentage of connected, continuous, range [0.00,1.00].

(28) **Srv_Rerror_Rate:** over the last two seconds, in connection with the current connection with the same service, "REJ" errors the percentage of connected, continuous, range [0.00, 1.00].

(29) **Same_Srv_Rate:** within the past two seconds, in connection with the current connection with the same target host, with the current connection with the same service connection percentage of continuous, range [0.00, 1.00].

(30) **Diff_Srv_Rate:** within the past two seconds, in connection with the current connection with the same target host, with the current connection with the different service connection percentage of continuous, range [0.00, 1.00].

(31) **Srv_Diff_Host_Rate:** within the past two seconds, in connection with the same service with the current connection, and is currently connected with a different target host to connect the percentage of continuous, range [0.00, 1.00].

***4. Host-based network traffic statistical characteristics (ten kinds)***

(32) **Dst_Host_Count:** 100 connections, with the current connection with the same target host connections, continuous in [0, 255].

(33) **Dst_Host_Srv_Count:** 100 connections, the number of connections with the same target host the same service with the current connection, continuous in [0, 255].

(34) **Dst_Host_Same_Srv_Rate:** 100 connections, with the current connection with the same target host the same service connection percentage, continuous [0.00, 1.00]

(35) **Dst_Host_Diff_Srv_Rate:** 100 connections, connected with the current connection to the percentage of different services with the same target host, continuous, [0.00, 1.00].

(36) **Dst_Host_Same_Src_Port_Rate:** 100 connections, with the current connection with the same source port for the same target host to connect the percentage of continuous in [0.00, 1.00].

(37) **Dst_Host_Srv_Diff_Host_Rate:** 100 connections, with the current connection with the same target host the same service to connect with current connection has a different source host to connect the percentage of, continuous in [0.00, 1.00].

(38) **Dst_Host_S_Error_Rate:** 100 connections, with the current connection with the same target host connection, the percentage of SYN errors of connection, continuous in [0.00, 1.00].

(39) **Dst_Host_Srv_S_Error_Rate:** 100 connections, with the current connection with the same connection to the service of the same target host, the percentage of SYN errors of connection, continuous in [0.00, 1.00].

(40) **Dst_Host_R_Error_Rate:** 100 connections, REJ error to connect the percentage of connection with the same target host is currently connected, continuous in [0.00, 1.00].

(41) **Dst_Host_Srv_R_Error_Rate:** 100 connections, with the current connection with the same target host the same service connection, REJ error connection to the percentage of, continuous in [0.00, 1.00].

## ***4. Development of Software :-***

For Feature extraction, we depend on the data source and the attacks to be detected. For Decision rule auto generation, we use rough set classification.

For this approach we proceed in the following three steps –

## ***4.1. Preprocessing :-***

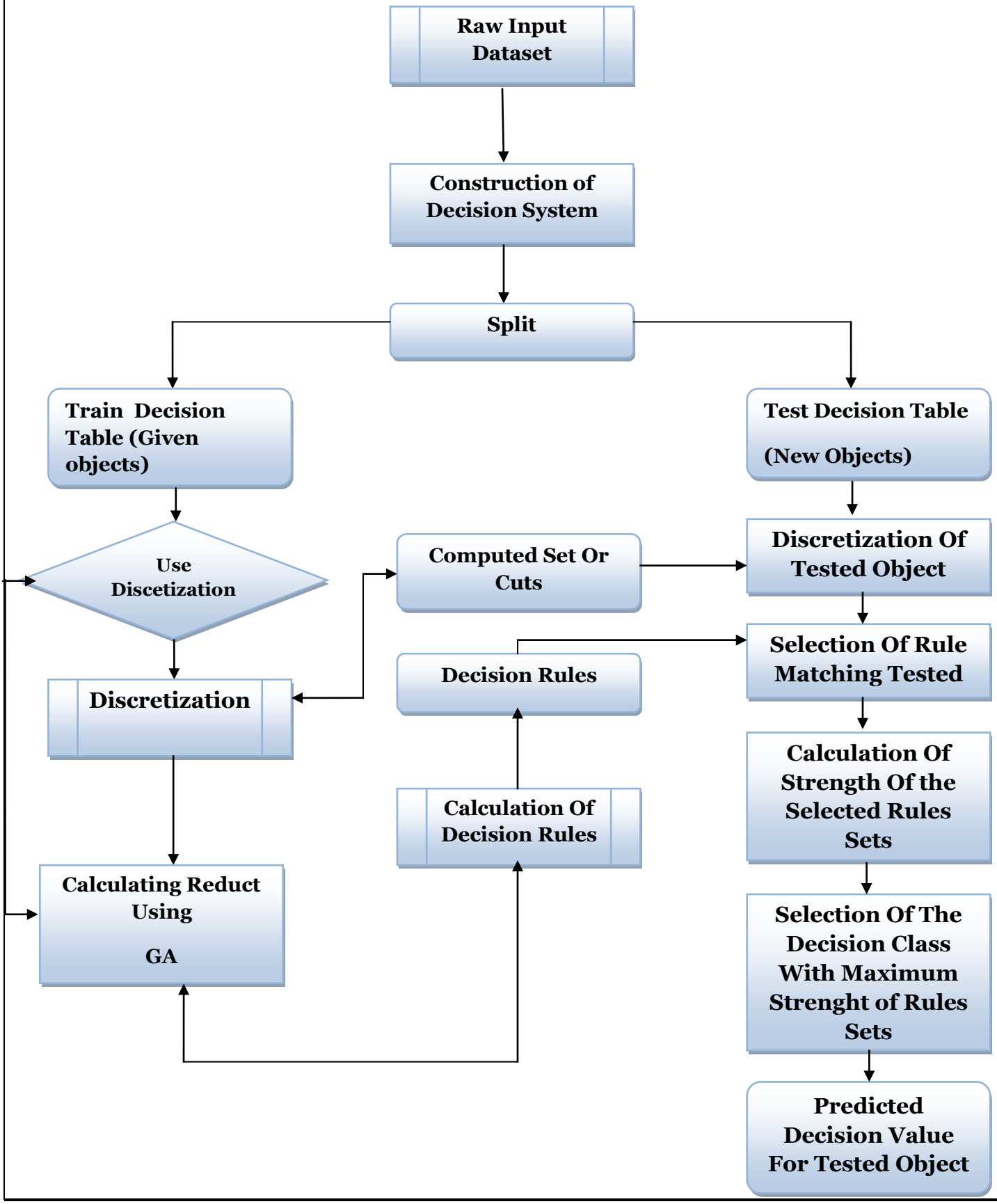
The raw data is first partitioned into three groups of attacks namely -

- DoS attack detection dataset,
- Probe attack detection dataset,
- U2R&R2L attack detection dataset.

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

- The training decision system,
- The testing decision system.

*Fig 1. General Scheme For Rough Set Classification Algorithm*



## *4.2 Training Decision System :-*

Each training dataset uses the corresponding input features and falls into two classes - normal (+1) and attack (−1) .We use these steps to develop Training Decision System for all the three training datasets -

## *1 ) Discretization :-*

On real value attribute to get the higher quality of classification rules we use this strategy.For this purpose we use Equal Width Discretization method.

**a)Equal Width Discretization Method** :-

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 Amin and Amax into k intervals of equal width. Thus the intervals have width -

$$w = (Amax - Amin) / k$$

Amax= Attribute A maximum value for all objects

Amin= Attribute A minimum value for all objects

K is user predefined parameter which is set 10 in this model

and the cut points are at Amin + w; Amin + 2w; .....; Amin + (k - 1) w..

Algorithm has a time complexity O(nlogn) where n is the number of generated intervals .

## *2 ) Decision Rule Generation :-*

For decision rule generation, we have used Reduct which is generated from attribute reduction Hybride Genetic algorithm which is based on attributes significance heuristic rules to find minimal reducts. This algorithm reduces the training time and make classifier more effective. We have extended this algorithm to find minimal reduct problem using rough set theory. GA approach is generally used with combinational problems to make problem solving more fast and accurate. We use approximate but faster heauristics such as SGF(significance of the attribute calculation) approach to find minimal reduct R.These approaches are easy to tune in GA and are more effective than general GA. Unfortunately, this approach is often sub-optimal and cannot avoid local optima. Moreover, the deterministic algorithm such as reduct computation based on SGF often spend more time on computations and be no hope for improvement. The hybrid genetic algorithms which use

non-deterministic, problem-oriented heuristics controlled by GAs can exploit the advantages of both genetic and heuristic algorithms. The heuristic method based on the significance of attribute *SGF*(*a*, *R*, *D*) is introduced in our rough set classification algorithm by a new Bit-adapt operator. This Bit-adapt operator is our main extension of reproduce selection step in the above general genetic algorithm.

These following steps describe how it fits in the intrusion detection environment –

## ***A ) Frame of Hybride Genetic Algorithm : -***

Finding the rough set minimal reduct is taken as hitting set problem . For discretized decision system

$$L = \{ U, AU\{d\} ,V,F\}$$

We are creating a multi set. For creating multi set we first create discernibility matrix for decision system L. Decision system L discernibility matrix is defined as –

$$\mathbf{M}(U) = [M^{d}{}_{A}(i , j)]n\times n$$

is defined as

$$M(U) = \{ \begin{matrix} ak \mid ak \in A \cap ak(xi) \neq ak(xi)\} & d(xi) \neq d(xj) \\ \emptyset & d(xi) = d(xj) \end{matrix}$$

where ak(xj) is the value of objects xj on attribute ak,d(x) is the value of object x on decision attribute d. Write **M**(U) = [MAd (i , j)]n×n as a list {p1,…, pt}.
Each $p_i$ is called a discernibility entry, and is usually written as $pi=ai_1, \ldots, ai_m$, where each $ai_k$ corresponds to a condition attribute of the information system k=q,…,m; i=1,…,t.

Discernibility matrix can be represented by the discernibility function f, f=p1∩…∩pt, where each $pi=a_{i1} \cup \ldots \cup a_{im}$ is called a clause.

f is a CNF Boolean function, so it has a multiset. Let M(f) denote the multiset of discernibility function f,

M(f) = {{a11,…,a1m},…,{ai1,…,aim},…, {at1,…,atm}}.

Subsequently the hiting set of multiset is constructed by using hybrid genetic algorithm. 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.

HS (M) = {B ⊆ A | B ∩ Si ≠ Ø for all Si in M}

## ***B ) Representation (Generation of the Initial Population) :-***

Initial population is a set *P* of elements from $2^A$, encoded as bit vector , where each bit indicates the presence of a particular element in the set. The variable *x* has a value represented by a string of bits that is *Ngene* long. For example, assume that we have 41 condition attributes like in our case {*a1, a2, …, a41*} 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). Our fitness function for decision system *L* = (*U*, *A* ∪ {*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 is a subset or equal to A.*

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

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 : -***

This step is implimented in two steps –

**STEP 1**:

- Calculate the fitness for each chromosome in the current generation

- Use heuristic rule to make genetic algorithm converge faster.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 C-R which has maximum SGF(a,R,D) = P(a) value . P(a) is the number of time attribute a present in remaining discernibility matrix which removes non-empty interaction with R.

- If there are more then one $a_j$ {j = 1,2 ….m} for maximal value then stocastically choose one attribute from them.
- Set the corresponding to $a_j$ as '1'.

❖ Then according to fitness value we select the parent.

**STEP 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 minsingle(Offspring) with maxsingle(Parent).

## ***E ) Crossover and Mutation :-***

**Crossover :-**

We use classical, one-point crossover. Crossing-over process affects chromosome selected for reproduction with probability of $P_c$.Crossover probability ($P_c$) is the probability that crossover will occur at a particular mating. All mating are not reproduced by the crossover. So use crossover probability to decide which mating is reproduced by crossover. For this purpose, we choose a crossover probability $P_c$ and for each mating, a random number is generate between 0 to 1 and if this newly generated random number is greater than the $P_c$ then we are going to use crossover over that mating and reproduce new mating. In this model we use $P_c = 1$ .

**Mutation: -**

In the mutation process, we first select a chromosome to be mutated with probability $P_m$ 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". Here $P_m$ mutation probability is the probability which is basically a measure of the likeness that a random element of our chromosome will be flipped into something else . Suppose if we take the mutation probability 0.001. This means that in our chromosome, we chose a bit randomly from 1000 bit flip value of that bit. In our model we use $P_m = 0.01$

## *F ) Decision Rule Generation:-*

In this step we generate the rule using reduct. Reduct is the reduced set of relation that conserve the same classification ability of relation. We further by using the confidence or strength (α) of the attribute and find another indispensable attribute of the table. The confidence or strength for an association rule x -> D is :-

$$\alpha = \frac{\text{No of example that contain x } \cup D}{No\ of\ example\ that\ contain\ x}$$

Suppose our reduct set is r = {$x_1$, ….,$x_n$} . Then we calculate the strength of all the attributes for association rule $x_i$->D {i = 1, 2,…n}. From all the reduct attributes, the attribute which have maximum strength is indispensible among other attributes. Suppose from reduct set, x2 and x3 have maximum strength then new reduct R' = {x2,x3}. Using this attribute we reduce the table by merging duplicate rows .Now we again eliminate the identical rows. Now if no further reduction is possible then we give final decision rules in the following ways :-

1. IF X2 -> P AND X3 -> A THEN D -> 1
2. IF X2 -> Q AND X3 -> B THEN D -> 0

The above two stamements defines that if attribute X2 is P and attribute X3 is A then decision attribute D is 1.

## *4.3 Testing Decision System :-*

In this step, we test the perpormance of the model on a testing data. The following are the steps to be followed –

- Discretization method : Used to dicretize the row testing data object.
- Generated rules are used to check the strength of the rule for matching object to its decision class.
- The object is assigned to the class which has maximum strength for decision rule set.

## *4.4 Pseudo-code for the hybrid genetic algorithm based SGF :-*

P←initializePopulation();

evaluate(P);

```
while(not terminate(P)) do
        Parents[1..2]←selectParents(P);
        Offspring[1]←CrossoverParents(Parents[1]);
        Offspring[2]←Mutation(Parents[2]);
        P←recombine (P, Offpring[1..2], Parents[1..2]);
        Bit-Adapt(P);   //This operator implements an adaptation strategy, the attribute subset
                        (represented by each chromosome) has the approximate classification ability
                        of the whole condition attribute set;
        evaluate(P);
done
```

## ***5. Testing and Analysis :-***

## ***5.1 Testing :-***

The raw data from the KDD99 contest is first partitioned into three groups (input dataset):

- **DoS attack detection dataset** – This dataset consists 250 datapoint : 150 for training and 100 for testing purpose. From these 200 are attacked dataset and 50 are normal data point. Data points are used for training using the RSC algorithm. The generated rules are used to predict the tested objects.

- **Probe attack detection dataset** - This dataset consists 250 datapoint : 150 for training and 100 for testing purpose. From these 180 are attacked dataset and 70 are normal data point. Data points are used for training using the RSC algorithm. The generated rules are used to predict the tested objects.

- **U2R&R2L attack detection dataset** - This dataset consists 250 datapoint : 150 for training and 100 for testing purpose. From these 160 are attacked dataset and 90 are normal data point. Data points are used for training using the RSC algorithm. The generated rules are used to predict the tested objects.

For each input dataset, we construct a decision system using the following method:

- For each attack detection dataset, different connection record feature set are selected as the condition attributes of the decision system. For Probe and DoS attack, intrinsic and traffic features are used; for U2R&R2L attack, intrinsic and content features used.

- The label (normal ‘0’, attack ‘1’) variable of each record is used as the decision attribute of the decision system.

- A connection record data point is used as an object in the decision system. Each of the constructed decision systems will be processed by RSC subsequently.

Data points are used for training using the RSC algorithm. The generated rules are used to predict the tested objects.

**********

We can explain the tested results by the following example :-

## ***1 )Result For DoS Class :-***

Training data of DoS attack detection dataset i gave as input in DoS Attack training system and it generate the decision rule in given table 3 formate -

***Table 1 : Generated Rule For DoS Attack Data Set Class -***

| Rule No | Attribute 4 | Attribute 12 | Attribute 23 | Attribute 33 | Decision Class |
|---|---|---|---|---|---|
| **1** | 1 | 3 | 1 | 4 | 0 |
| **2** | 1 | 1 | 1 | 1 | 1 |
| **3** | 1 | 3 | 1 | 2 | 0 |
| **4** | 1 | 1 | 4 | 1 | 1 |
| **5** | 1 | 3 | 1 | 3 | 0 |
| **6** | 2 | 1 | 1 | 1 | 1 |
| **7** | 2 | 1 | 2 | 1 | 1 |
| **8** | 1 | 3 | 1 | 1 | 1 |

After geting these decision rules, we give Testing data of the DoS Attack dataset as input in DoS Attack testing system and testing system use these generated decision rules and classify the newly object in class ‘1’(for intrusion) and ‘0’ for normal object.

***Table 2 :Testing Result For DoS Attack Data Set Class-***

```
class of object 0 is calculated by our model is -- 1 and actual by dataset is 1
class of object 1 is calculated by our model is -- 1 and actual by dataset is 1
class of object 2 is calculated by our model is -- 1 and actual by dataset is 1
class of object 3 is calculated by our model is -- 1 and actual by dataset is 1
class of object 4 is calculated by our model is -- 1 and actual by dataset is 1
class of object 5 is calculated by our model is -- 1 and actual by dataset is 1
class of object 6 is calculated by our model is -- 1 and actual by dataset is 1
class of object 7 is calculated by our model is -- 1 and actual by dataset is 1
class of object 8 is calculated by our model is -- 1 and actual by dataset is 1
class of object 9 is calculated by our model is -- 1 and actual by dataset is 1
class of object 10 is calculated by our model is -- 1 and actual by dataset is 1
class of object 11 is calculated by our model is -- 1 and actual by dataset is 1
class of object 12 is calculated by our model is -- 1 and actual by dataset is 1
class of object 13 is calculated by our model is -- 1 and actual by dataset is 1
class of object 14 is calculated by our model is -- 1 and actual by dataset is 1
class of object 15 is calculated by our model is -- 1 and actual by dataset is 1
class of object 16 is calculated by our model is -- 1 and actual by dataset is 1
class of object 17 is calculated by our model is -- 1 and actual by dataset is 1
class of object 18 is calculated by our model is -- 1 and actual by dataset is 1
class of object 19 is calculated by our model is -- 1 and actual by dataset is 1
class of object 20 is calculated by our model is -- 1 and actual by dataset is 1
class of object 21 is calculated by our model is -- 1 and actual by dataset is 1
class of object 22 is calculated by our model is -- 1 and actual by dataset is 1
class of object 23 is calculated by our model is -- 1 and actual by dataset is 1
class of object 24 is calculated by our model is -- 1 and actual by dataset is 1
```

***2) Result For U2R & R2l Class :-***

Training data of U2R & R2L attack detection dataset i gave as input in U2R & R2L Attack training system and it generate the decision rule in given table 5 formate -

***Table 3 : Generated Rule For U2R & R2L Attack Data Set Class –***

| Rule No | Attribute 4 | Attribute 25 | Attribute 26 | Attribute 27 | Decision Class |
|---|---|---|---|---|---|
| **1** | 4 | 1 | 1 | 3 | 1 |
| **2** | 3 | 3 | 2 | 1 | 1 |
| **3** | 1 | 2 | 2 | 1 | 1 |
| **4** | 1 | 1 | 1 | 1 | 0 |
| **5** | 2 | 3 | 3 | 1 | 1 |
| **6** | 3 | 3 | 2 | 1 | 1 |
| **7** | 1 | 2 | 2 | 1 | 1 |
| **8** | 3 | 3 | 3 | 1 | 1 |
| **9** | 4 | 2 | 2 | 2 | 1 |

After getting these decision rule we give Testing data of the U2R & R2L Attack dataset as input in U2R & R2L Attack testing system and testing system use these generated decision rule and classify the newly object in class '1'(for intrusion) and '0' for normal object .

***Table 4:Testing Result For U2R & R2L Attack Data Set Class-***

```
class of object 0 is calculated by our model is -- 1 and actual by dataset is 1
class of object 1 is calculated by our model is -- 1 and actual by dataset is 1
class of object 2 is calculated by our model is -- 1 and actual by dataset is 1
class of object 3 is calculated by our model is -- 1 and actual by dataset is 1
class of object 4 is calculated by our model is -- 1 and actual by dataset is 1
class of object 5 is calculated by our model is -- 1 and actual by dataset is 1
class of object 6 is calculated by our model is -- 1 and actual by dataset is 1
class of object 7 is calculated by our model is -- 1 and actual by dataset is 1
class of object 8 is calculated by our model is -- 1 and actual by dataset is 1
class of object 9 is calculated by our model is -- 1 and actual by dataset is 1
class of object 10 is calculated by our model is -- 1 and actual by dataset is 1
class of object 11 is calculated by our model is -- 1 and actual by dataset is 1
class of object 12 is calculated by our model is -- 1 and actual by dataset is 1
class of object 13 is calculated by our model is -- 1 and actual by dataset is 1
class of object 14 is calculated by our model is -- 1 and actual by dataset is 1
class of object 15 is calculated by our model is -- 1 and actual by dataset is 1
class of object 16 is calculated by our model is -- 1 and actual by dataset is 1
class of object 17 is calculated by our model is -- 1 and actual by dataset is 1
class of object 18 is calculated by our model is -- 1 and actual by dataset is 1
class of object 19 is calculated by our model is -- 1 and actual by dataset is 1
class of object 20 is calculated by our model is -- 1 and actual by dataset is 1
class of object 21 is calculated by our model is -- 1 and actual by dataset is 1
class of object 22 is calculated by our model is -- 1 and actual by dataset is 1
class of object 23 is calculated by our model is -- 1 and actual by dataset is 1
class of object 24 is calculated by our model is -- 1 and actual by dataset is 1
```

***3) Result For Probe Class :-***

Training data of Probe attack detection dataset i gave as input in Probe Attack training system and it generate the decision rule in given table 7 formate -

***Table 5 : Generated Rule For Probe Attack Data Set Class –***

| Rule No | Attribute 4 | Attribute 25 | Attribute 33 | Attribute 34 | Decision Class |
|---|---|---|---|---|---|
| **1** | 1 | 1 | 2 | 1 | 0 |
| **2** | 1 | 1 | 2 | 2 | 0 |
| **3** | 1 | 1 | 3 | 2 | 0 |
| **4** | 1 | 1 | 1 | 1 | 1 |
| **5** | 3 | 3 | 1 | 1 | 1 |
| **6** | 3 | 3 | 1 | 1 | 1 |
| **7** | 1 | 1 | 4 | 2 | 0 |

After geting these decision rule we give Testing data of the Probe Attack dataset as input in Probe Attack testing system and testing system use these generated decision rule and classify the newly object in class '1'(for intrusion) and '0' for normal object .

***Table 6:Testing Result For Probe Attack Data Set Class-***

```
class of object 0 is calculated by our model is -- 1 and actual by dataset is 1
class of object 1 is calculated by our model is -- 1 and actual by dataset is 1
class of object 2 is calculated by our model is -- 1 and actual by dataset is 1
class of object 3 is calculated by our model is -- 1 and actual by dataset is 1
class of object 4 is calculated by our model is -- 1 and actual by dataset is 1
class of object 5 is calculated by our model is -- 1 and actual by dataset is 1
class of object 6 is calculated by our model is -- 1 and actual by dataset is 1
class of object 7 is calculated by our model is -- 1 and actual by dataset is 1
class of object 8 is calculated by our model is -- 1 and actual by dataset is 1
class of object 9 is calculated by our model is -- 1 and actual by dataset is 1
class of object 10 is calculated by our model is -- 1 and actual by dataset is 1
class of object 11 is calculated by our model is -- 1 and actual by dataset is 1
class of object 12 is calculated by our model is -- 1 and actual by dataset is 1
class of object 13 is calculated by our model is -- 1 and actual by dataset is 1
class of object 14 is calculated by our model is -- 1 and actual by dataset is 1
class of object 15 is calculated by our model is -- 1 and actual by dataset is 1
class of object 16 is calculated by our model is -- 1 and actual by dataset is 1
class of object 17 is calculated by our model is -- 1 and actual by dataset is 1
class of object 18 is calculated by our model is -- 1 and actual by dataset is 1
class of object 19 is calculated by our model is -- 1 and actual by dataset is 1
class of object 20 is calculated by our model is -- 1 and actual by dataset is 1
class of object 21 is calculated by our model is -- 1 and actual by dataset is 1
class of object 22 is calculated by our model is -- 1 and actual by dataset is 1
class of object 23 is calculated by our model is -- 1 and actual by dataset is 1
class of object 24 is calculated by our model is -- 1 and actual by dataset is 1
```

The experiments results are shown in Table 9, where the training time units format is minutes: seconds. Training time 1 denotes the training time without "Bit-Adaptation strategy" in the genetic algorithm; training time 2 denotes the training time with "Bit-Adaptation strategy". RSC- ∈ refers to the parameter ∈ used in the reduct computation. ∈ = 1 means it is the accurately computed hitting set without approximation. From the above table, our proposed "Bit- Adaptation strategy" can decrease the training time.

***Table 9. Experiment Based Result For RSC***

| Category | RSC- ∈ | Detection Rate% | Training Time 1 ( min : sec ) | Training Time 2 ( min : sec ) |
|---|---|---|---|---|
| **Probe Attack** | ∈ = 0.9 | 95.1614 | 0.89 | 0.88 |

| | | | | |
|---|---|---|---|---|
| | ∈ = 1.0 | 96.1718 | 0.88 | 0.86 |
| **DOS Attack** | ∈ = 0.9 | 94.9178 | 0.46 | 0.35 |
| | ∈ = 1.0 | 95.7535 | 0.44 | 0.33 |
| **U2R & R2L Attack** | ∈ = 0.9 | 86.5331 | 1.46 | 1.44 |
| | ∈ = 1.0 | 87.9686 | 1.40 | 1.33 |

## *5.2 Analysis :-*

For detection rule generation for all three datasets we have taken the discretization coefficient k as 4. This value is taken after the experimental analysis. And for Equal width discretization method we have predefined min and max value for the attributes according to their range.For k=4 we can get result more accurate .If we increase k value then our result will not be efficient and it takes more time to generate the rules.

For fitness function we take ∈ = 1 or ∈= 0.9 both values as decision coefficient.And after the analysis all the three data sets generated the decision rules. We can find 6 most significant feature attribute namely 3, 24 , 25, 28, 31, 32. These attribute sets are minimal reduct for classification .

Using reduct, we can generate rule for each classifier and get result such that which decision class a new object will get. RSC generate decision rule in explainable "IF-THEN" format.
For instance, one of the Probe attack detection rules is:

"IF ATTRIBUTE 4 = 1 AND ATTRIBUTE 24 = 1 AND ATTRIBUTE 26 = 1 AND ATTRIBUTE 32 = 4 THEN RESULT IS 1 "

In simple word "if attribute "**Flag**" = (1,2.75] and attribute "**Sev_count**" =(0,125.5] and attribute "**Srv_rerror_rate**" = (0.0.25] and attribute "**Dst_host_count**" =(0,63.75] then decision class for that new object will be 1 means intrusion is happening" .

We can improve the design of the detector by modify detector so that this situation is not happen in future. Using above decision rule if feature attribute "Flag " have value between 0 to 2.75 and attribute "Sev_count" =(0,125.5] and "Srv_rerror_rate =(0,0.25 ]" and attribute "Dst_host_count = (0.63.75]" that means intrusion condition so our detector will not allow that data point to enter in network and due to that action we can prevent intrusion before happening.

This advantage together with its high detection performance for some attacks, makes the RSC algorithm very valuable in practical intrusion detector design. We use the heuristic rule to

accelerate the convergence speed of the reduct computation and decrease the training time of RSC, the training time of RSC is still long and needs further improvement. However, the running time of RSC testing is notably short since it just needs judgment of some conditions.

## *Comparision Of RSC IDs and Back Propagation Neural Network Approach IDs*

***Table 10 : Comparison Between RSC IDS and BPNN IDS –***

| Category | RSC IDs (∈ = 1) | BPNN Ids |
|---|---|---|
| **DoS Attack** | 95.7535 % | 90.6% |
| **Probe Attack** | 96.1718% | 90.6% |
| **R2L & U2R** | 87.9686% | 90.6% |

For DoS attack and Probe attack the detection rate is high (above 99 %) But for U2R&R2L attack detection, RSC IDs is worse than BPNN IDs. 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).But BPNN algorithm IDS work well with both small and big dataset .So we can say that for DoS and Probe attacks our RSC Ids working very well but for R2L & U2R attack BPNN IDs is working well then RSC Ids.

## *6. 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. I use rough set classification (RSC) for intrusion detection system (IDS) feature ranking and intrusion detection rules generation. Intrusion detection using RSC can yield both explainable detection rules and high detection rate for some attacks (DoS and Probe Attack). And feature ranking using RSC for IDS is simple and fast. In addition, we proposed a hybrid genetic algorithm based on the attribute significance to compute the rough set reduct and accelerate the convergence speed and decrease the training time of RSC. But for the real-time IDS, the training time of RSC is still long and needs further improvement.

## *7 . Future Work : -*

- This project classifies Dos and Probes Attack accurately (above 99% accuracy). But for U2R&R2L attack detection, RSC algorithm is worse than any other classifying algorithm. 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). So we can improve

this classification for U2R& R2l attack also by increasing U2R& R2l attack entry in the dataset in significant amount.

- Using RSC we can make antivirus in very efficient way. The reason behind is that it will give us the rules for intrusion happening or using this rule we can prevent intrusion.

- Using RSC we can also store history of attacks so that we can identify a user chances of attacking the network .

- Using Rough Set Classifier we can improve the efficiency of detecting intrusion and decrease attack probability on network.

# ***References***

[1] Mostaque Md. Morshedur Hassan, International Journal of Distributed and Parallel Systems (IJDPS) Vol.4, No.2, on "**CURRENT STUDIES ON INTRUSION DETECTION SYSTEM, GENETIC ALGORITHM AND FUZZY LOGIC",** March 2013.

[2] M. Ali Aydın*, A. HalimZaim, K. GokhanCeylan, "**A hybrid intrusion detection system design for computer network security**", Research Fund of Istanbul University, Turkey, February 2009

[3] Marc Wilikens, Workshop report on "**First International Workshop on the Recent Advances in Intrusion Detection**", Workshop held in Louvain-la-Neuve, Belgium on 14-16 September 1998

[4] Amrita Anand *, Brajesh Patel, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 8 on **"An Overview on Intrusion Detection System and Types of Attacks It Can Detect Considering Different Protocols"**, August 2012

[5] Richard E Overill, International Centre for Security Analysis King's College London publications on "**How Re(Pro)active Should an IDS be?",** Dec 01, 2014

[6] Dr.Fengmin Gong, McAfee Network Security Technologies Group publications on **"Next Generation Intrusion Detection Systems (IDS)"**, March 2002

[7] RoshaniGaidhane, Student*,* Prof. C. Vaidya, Dr. M. Raghuwanshi, International Journal of Advance Foundation and Research in Computer (IJAFRC), Volume 1, Issue 2 on "**Survey: Learning Techniques for Intrusion DetectionSystem (IDS**)", Feb 2014.

[8] Kanchan S. Tiwari, Ashwin G. Kothari, and Avinash G. Keskar, International Journal of Future Computer and Communication, Vol. 1, No. 3 on "**Reduct Generation from Binary Discernibility Matrix: A Hardware Approach**", October 2012

[9] ZHANG Lian-hua,ZHANG Guan-hua, YU Lang ZHANG Jie,BAI Ying-cai , Journal of Zhejiang University SCIENCE on "**Intrusion detection using rough set classification**, July 21, 2003

[10] Martuza Ahmed, Rima Pal, International Association of Computer Science and Information Technology - Spring Conference on "**A comparative study on the currently existing intrusion detection systems**", 2009

[11] Mohammad Sazzadul Hoque1, Md. Abdul Mukit2 and Md. Abu Naser Bikas, International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2 on **"AN IMPLEMENTATION OF INTRUSION DETECTION SYSTEM USING GENETIC ALGORITHM",** March 2012.

## ***Suggestions of board members***