Hybrid Genetic Fuzzy Rule Based Inference Engine to Detect Intrusion in Networks

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Kriti Chadha
(Roll No. 801232010)

Under the supervision of:
Dr. Sushma Jain
Asstt. Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "***Hybrid Genetic Fuzzy Rule Based Inference Engine to Detect Intrusion in Networks***", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Sushma Jain* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Kriti Chadha
Signature:
(Kriti Chadha)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Sushma Jain)
Assistant Professor,
CSED

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

No volume of words is enough to express my gratitude towards my guide **Dr. Sushma Jain**, Computer Science & Engineering Department, Thapar University, Patiala, who has been very concerned and has aided for all the materials essentials for the preparation of this thesis report. She has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am also thankful to **Dr. S.K Mohapatra**, Dean of Academic Affairs, **Dr. Deepak Garg**, Head of Computer Science & Engineering Department and **Mr. Ashutosh Mishra**, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there at the need of hour and provided with all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

ABSTRACT

With the drastic increase in internet usage, various categories of attacks have also evolved. These attacks exploit system vulnerabilities thus posing danger to the sensitive or private information stored in the system thus violating confidentiality, integrity and availability of the resources. Conventional intrusion detection techniques like firewall have been implemented to counter these attacks but they too have failed due to the increased potential of the attackers or hackers as they use innovative ideas to attack the system. Thus substantial systems are needed to eliminate these attacks before they inflict huge damage to the organization by retrieving all the sensitive information. Computational techniques are currently being considered as a novel field to detect intrusions due to their characteristic properties such as adaptability, fault tolerance and higher accuracy. Genetic fuzzy system is considered to be the most suitable approach for constructing the intrusion detection system. The proposed approach is the hybrid of fuzzy logic with genetic algorithm and then implying mathematical model to cover the whole dataset, thus making the approach suitable for high dimensional problems. The fuzzy logic constructs precise and flexible patterns while the genetic algorithm based on evolutionary computation helps in attaining an optimal solution due to its learning capability. The proposed approach has been applied on KDD-99 dataset, which is the most widely used network dataset and is able to classify normal connections as well as anomalies with greater precision. The intended approach has been compared with the existing genetic fuzzy systems used in intrusion detection and other approaches on basis of various metrics to check its performance and conciseness in classification.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Intrusion

Internet is a global public network. There has been a tremendous growth in the networks which has shifted the era from centralized computing to distributed computing. Hence various heterogeneous computers communicate and exchange valuable information. As the complexity and usage of internet has invariably grown, the responsibility to ensure continuous operation and maintaining security has grown. On one side internet has an enormous potential to provide open and easy communication and reach the end users, while on the other hand it brings a lot of risks with serious threats, intrusions and vulnerabilities to the systems.

Set of operations that endeavor to subvert the integrity, confidentiality and availability of a resource is defined as intrusion [1].The malicious user applies different methods or skills such as sniffing or cracking password to exploit system vulnerabilities, thus posing a threat to individual privacy or organizational or even sensitive national information, thus making the system unreliable to use.

There are different categories of attacks such as probe, viruses or denial of services which are launched by the attackers in various ways. The hacker gains access to the system or the services either through social engineering by deceiving a user or launching dos attacks by exploiting implementation bugs, or system misconfiguration or masquerading where the attackers beguile the system by misrepresenting themselves as the authorized users.

Kendall *et al.* [14] in 1999 provided the detailed taxonomy of the attacks on DARPA 98 dataset which was used to evaluate the performance and accuracy of the intrusion detection systems designed. Thus there are mainly four categories of attacks:-

1. ***Denial of Service (DoS) Attacks****:*-It is an explicit attempt by attackers to make a machine or network resource inaccessible by flooding useless traffic, thus preventing legitimate users to use the machine. DoS attacks such as mailbomb and neptune are considered as abuses while teardrop and ping of death are bugs which create abnormal packets to create skepticism in TCP/IP stack.
2. ***User to Root (U2R) Attacks****:*-In this category of attack, the local user is able to exploit the vulnerability and thus obtain privileges which have been reserved

for the administrator. The root access can be obtained by either cracking passwords or buffer overflow attack where more amount of data is stored in the static buffer beyond its capacity. The U2R attacks exploit the programs which are unable to manage temporary files or the domain in which they are being managed.

3. ***Remote to User (R2L) Attacks:***-In this class of attack, the attacker who does not have an account on the local machine but has the ability to send packets, exploit the vulnerability and gain local access to the file and can tamper the data. Some of the examples of R2L attacks are imap, phf, sendmail etc. Xlock is a social engineering attack while ftp-write, xsnoop and guest try to maneuver security policies of the system.
4. ***Probe Attacks:***-It is an attempt to gather information or surveillance about network of computers to find valid IP addresses, OS types, active ports and vulnerability. The attackers use scanning tools such as satan, mscan to retrieve information about the systems and the services running on them and obtain vulnerable points which can be used to launch an attack.

1.2 Intrusion Detection Systems

These intrusions discussed above are difficult to be detected by firewalls or basic security mechanisms as the attackers use novel ideas to commence an attack. Hence impregnable intrusion detection systems are needed to counteract these vulnerabilities and maintain the CIA of a resource.

CIA triad (confidentiality, integrity and availability) is the most widely used security model and one of the essential principles of information security. They are explained as follows:

1. ***Confidentiality:***-It refers to hiding information or preventing disclosure of important information to unauthorized people and allowing information to only authenticated users. Various encryption techniques are applied to maintain confidentiality of data.
2. ***Integrity:***-It ensures that the data is accurate and the original information has not been modified or corrupted.
3. ***Availability:***-The information should be accessible to authorized users at all the times. Ensuring availability, denial of service attacks (flooding of

incoming packets to the target system thus forcing it to shut down or unusable) can be prevented.

Breaching of any one of the principles can have serious consequences to the end users/parties.

Therefore, intrusion detection system is an automated system which can detect these threats in the network and maintain confidentiality, integrity and availability of a resource. It is a security system that acts as a watchdog for computer systems and network traffic and determines the entities that endeavor to disrupt the security mechanism [2]. Figure1.1 depicts the block diagram of intrusion detection system.

The intrusion detection systems (IDS) have been classified into mainly two types:-

1. ***Network Based IDS (NIDS)****:*-NIDS are placed at strategic points within the network to monitor traffic moving to and from all devices on the network. They work in promiscuous mode and collects packets using either a network tap or hub. The system then processes these packets and flags/triggers alarms. Some of the advantages of NIDS are:-
    - NIDS requires less cost of ownership and system overhead as it evaluates all the systems which are in the network.
    - NIDS does not influence the existing infrastructure. It would just detect the attacks in its network area and is therefore easier to deploy.
    - It evaluates the network traffic in real time. Thus it can detect the vulnerabilities as they occur and it holds the evidences for each activity.
2. ***Host Based IDS (HIDS)****:*-HIDS attempts to identify anomalous or unauthorized behavior on a specific device on the network. Here an agent is installed on each system which monitors operating system, write data to log files or trigger alarms. The HIDS monitors the incoming and outgoing messages from the host or device and alerts the administrator about the malicious activities. The benefits of HIDS are:-
    - HIDS monitors the malicious activities with more accuracy and generates fewer false alarms than NIDS.
    - No subsidiary hardware is required by HIDS as they are deployed on host machine only.
    - They are more economical than NIDS.

- Since the HIDS monitors the malicious activities like file access, altered file permissions etc, so it detects the malicious activities which NIDS cannot fulfill.

The intrusion detection system uses one of the two detection methods to detect and identify anomalous behavior:

1. ***Misuse detection:***-It is also referred to as signature based detection. A database is maintained which contains information about system vulnerabilities and previous attack profiles. When a malicious user tries to intrude, the intrusion is detected by looking into the database and an alarm is triggered. Therefore they have a potential of very low false rates and are more accurate in detecting attacks. But its drawback lies in the difficulty to maintain and update the database with the latest system vulnerabilities.
2. ***Anomaly Detection:***-A baseline is maintained which contains information about valid or normal behavior retrieved from reference information. This methodology refers a baseline to detect deviation from normal or expected behavior of systems or users. Therefore, they can attempt to detect new or unforeseen vulnerabilities. But, since the entire behavior of the system may not be considered, high false rate is considered as the drawback of anomaly detection.

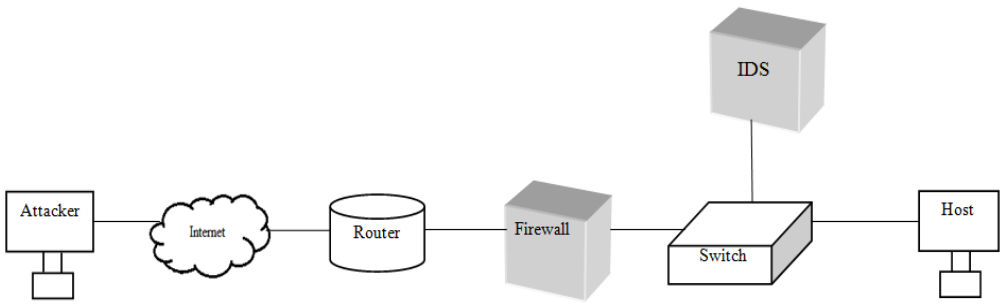

Figure1.1. Block Diagram of Intrusion Detection System

With the beneficiaries of the Intrusion Detection Systems, there are some challenges posed by them. Some of them are:-

- ***Human Intervention:***-Today the IDS is mostly automated with facilities like triggering an alarm for administrator when a malicious activity is detected, ostracizing the malicious connection, ceasing a vicious connection by automatically modifying router's access control list. But still, a skilled security

professional is required who can regularly monitor the logs and examine the malicious activities discovered by IDS over a particular time period as IDS has not reached a standard where it can give periodical analysis of intrusions detected.

- ***False Positives:***-While IDS has the capacity to detect malicious activities and generate alarms, but it also creates lot of false alarms. A lot of normal traffic is considered as false alerts and therefore the IDS may not be considered as properly configured.
- ***Deployment:***-Planning on how to deploy the IDS is very important in design and implementation phase of the IDS. Many organizations adopt a collaborative approach of NIDS and HIDS. So it is necessary that the organization possess sufficient resources for the success of IDS.
- ***Encrypted Data:***-IDS does not evaluate encrypted data. Therefore, once the data is settled in the system, it may release a virus or vulnerability which could severely affect the system.
- ***Reactive in nature:***-The IDS still possess reactive behavior rather than reactive. With the growth of internet at faster pace, the hackers use innovative ideas to launch attacks, therefore making the system more vulnerable to malicious activities. Thus the database consisting of signatures is required to be updated whenever a new attack or vulnerability is detected.

1.3 Fuzzy Logic

Fuzzy Logic was introduced by Zadeh in 1965.Since then; it has been used in various classification fields and control applications.

The fuzzy system consists of following components and processes:-

- ***Crisp set***- Crisp sets are those sets that represent bivalent condition i.e. either 0(false) or 1(true) meaning, that an element either belongs to a set or not. It is taken as an input variable for fuzzification process.
- ***Fuzzy Sets***- Fuzzy means ‘vagueness’ or ‘uncertainty’. It is a multi-valued logic that allows values to exist between 0(false) and 1(true) which represents human reasoning. The crisp sets are converted to fuzzy sets after the fuzzification process.
- ***Membership function***- It provides a measure of the degree of similarity of elements in the universe of discourse to the fuzzy set. It is not same as

probability but represents membership in vaguely defined sets. It is represented by $\mu_A(x)$.

For fuzzy sets, the membership function lies in the range of 0-1.

$$\mu_A(x) \in [0,1]$$

For crisp sets,

$$\mu_A(x) = \begin{cases} 0 & if\ x \in X \\ 1, & otherwise \end{cases}$$

- ***Fuzzification***- The process of converting crisp sets into fuzzy sets with membership function lying between 0 and 1 and represented by linguistic labels.
- ***Defuzzification***- The process of converting fuzzy sets back to crisp values is called defuzzification.
- ***Fuzzy Rule Based System***- A fuzzy rule based system is a collection of propositions containing linguistic variables and the rules are expressed in the form of: If X_1 is A_1 and X_2 is A_2 and ……..and X_n is A_n, then Y is B.

Figure1.2 represents all the processes undergoing in the fuzzy system. The crisp set is taken as input where the fuzzification process takes place and a fuzzy set is received as an output. This fuzzy set act as an input for the inference engine where a fuzzy reasoning process is conducted and a fuzzy output is obtained. Now the fuzzy output undergoes a defuzzification process to obtain a crisp output as a resultant.

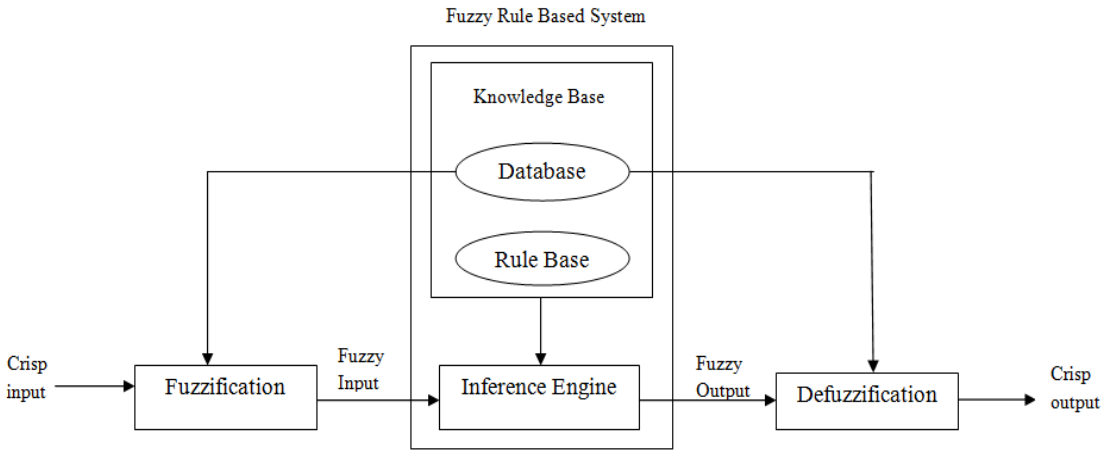

Figure1.2. Fuzzy Systems

1.3.1 Fuzzy Rule Based System

Fuzzy Rule Based systems consists of fuzzy if then rules and have been used in various pattern classification problems[5]. It consists of following components:-

- ***Knowledge Base***- The Knowledge Base is further divided into two types-
  1. ***Rule Base***- The rule base consist of set of rules. The rules can be represented in different ways. Here the rule is of the following structure-

     $$R^k - If\ X_1\ is\ A_1^k\ and\ \ldots\ldots\ldots X_n is\ A_n^k, then\ class\ is\ C^k with\ CD^k.$$

     Where each input variable X_i has a value from a set of linguistic labels $A_i^k [L_i^1\ or\ L_i^2\ \ldots. or\ L_i^i]$, C^k represents the consequent class and CD^k is the certainty degree or confidence($CD^k \in [0,1]$) . Confidence or certainty degree represents how effectively a fuzzy rule identifier assigns an input pattern to a class label [8].

     $$CD_r = \frac{\beta_{Class\ h_r}(R_j) - \overline{\beta}}{\sum_{h=1}^{c} \beta_{Class\ h_r}(R_j)} \quad (1)$$

     Where $\beta_{Class\ h_r}(R_j)$ represents the sum of membership function of training pattern in class h and $\bar{\beta}$ represents the sum of membership function of training pattern that do not belong to class h.
  2. ***Database***-The fuzzy set related to linguistic terms which are used in Rule Base is defined in Database (DB). Thus the number of linguistic labels (l_i) for each variable X_i and the membership function of the fuzzy set concerned with the linguistic terms are specified. Here each variable has been assigned five linguistic labels (i.e. very low, low, medium, high, very high) and each variable definition interval has been uniformly partitioned under triangular fuzzy sets as shown in figure 1.3.

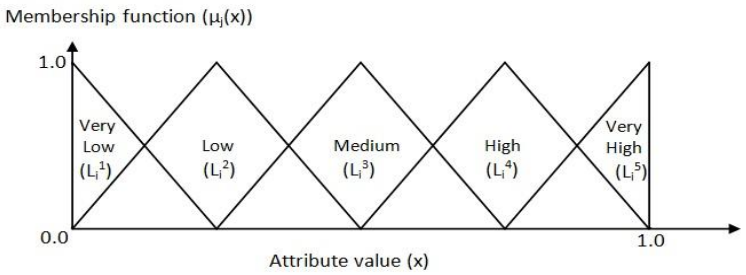

Figure1.3. Triangular membership functions of fuzzy sets

- ***Fuzzy Inference Engine***- It is a reasoning procedure that derives inference from fuzzy if-then rules. Here the incoming pattern is matched with the

antecedents of rules and the pattern is classified according to the rule consequent using the property of maximum matching.

So to classify an unlabelled pattern, $x_p = \{x_{p1}, x_{p2}, \ldots, x_{pn}\}$ from a given rule set, the input pattern with the rule set is matched and the following approach is applied.

The class of the unknown pattern is determined by the following equations:-

$$C = \max_{h=1,2..c}(\tau_h) \tag{2}$$

where

$$\tau_h = \max_{\substack{r=1..N \\ C_r \epsilon h}} \left\{ \mu_{A_r(x_p)} . CD_r \right\} \tag{3}$$

Thus the successful rule is the one which has highest τ_h but if more than one class has highest τ_h or if $\tau_h = 0$ then the rule is rejected.

Therefore in this manner, a rule set is built and the unseen patterns are classified into their specific classes based on their certainty degree.

Hence fuzzy rule based systems provide a good platform to deal with noisy or imprecise information and thus achieve two goals: accuracy and interpretability.

1.4 Genetic Algorithms

Genetic algorithms are inspired from Darwin's theory of evolution-survival of fittest. It is an adaptive heuristic search algorithm based on evolutionary ideas of natural selection and genetics and is very successful in search and optimization problems. Genetic algorithms exploit historical information to direct the search into the region of better performance. Some of the terminologies which are used in genetic algorithm are:-

- ***Chromosome***- A chromosome is a set of genes and represents the solution.
- ***Genes***- Each gene represents a parameter of the whole problem space and has a specific meaning.
- ***Population***- It represents the number of individuals or chromosome participating to find the optimized solution.
- ***Fitness***- It is the value assigned to each individual to determine how close the individual is from the solution and is calculated with the help of fitness function.

- ***Selection***- A proportion of individuals is selected from existing population during each successive generation.
- ***Crossover***- It is a genetic operator in which two individuals are selected and they perform intermingling to exchange information and generate two new individuals.
- ***Mutation***- It is also a genetic operator which alters a random gene or more number of genes in a chromosome from its initial state.

The basic steps of a genetic algorithm are as follows:-

- Before a genetic algorithm is applied, a method to encode potential solution to that problem is required. Binary encoding is one of the most common techniques to represent the information.
- A random population of chromosomes is generated.
- Evaluate fitness f(x) of each chromosome on the basis of fitness function.
- Now create a new population by repeating the following steps until a new population is created.
    - ***Selection***- A proportion of individuals is selected from the current population to procreate a new generation. The individuals are selected on the basis of different techniques on the basis of fitness value and the one with higher fitness will have higher probability of being selected. The different selection techniques are tournament selection, roulette wheel, rank selection and steady state selection.
    - ***Crossover***- Two parent chromosomes are combined to produce two new off springs with user definable crossover probability which is kept as high as 0.9. The various types of crossover operators are one point crossover, two point, uniform, arithmetic and heuristic crossovers.
    - ***Mutation***- It is used to maintain genetic diversity by altering one or more gene. It occurs according to user definable mutation probability and is kept to low scale of 0.1. It helps to prevent the population to stagnate at local optima. Some of the techniques used in mutation are flip bit, boundary, non-uniform etc. The flip bit is the most famous mutation technique.
- Now a newly generated population is used for further run of algorithm.

- If the termination condition is satisfied, then the best solution in the current solution is returned as the most optimized solution.

1.5 Genetic Fuzzy Rule Based Systems

Genetic fuzzy rule based system is a fuzzy system augmented or hybridized with evolutionary computing to increase the learning capacity of the system, thus providing robust search capabilities in a complex environment. The main function is to adopt an evolutionary learning mechanism which can automatically generate /design knowledge base and thus help in search and optimization problems. The next section clearly explains how the genetic algorithm helps in learning process.

The generic code structure and domain independent features of genetic algorithm make them desirable to incorporate apriori knowledge. This apriori information is in the form of linguistic labels, membership function or fuzzy rules in the fuzzy rule based system. Thus genetic algorithms are used to acclimate or assimilate information from rule base or database. These systems are used in many applications such as classification, data mining, control processes and modeling. Figure1.4 depicts the basic genetic fuzzy rule based system.

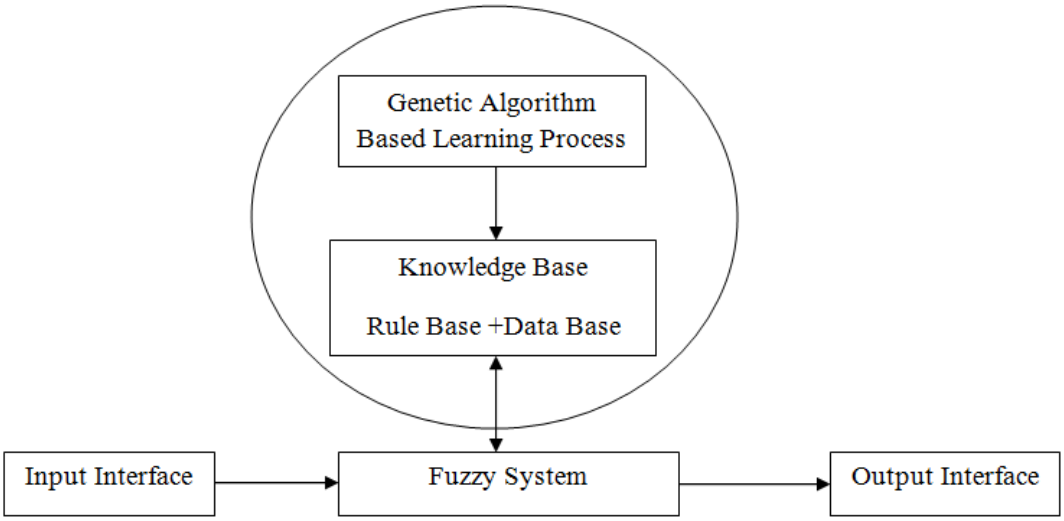

Figure1.4. Genetic Fuzzy Rule Based System

1.6 Learning with Genetic Algorithms

Genetic algorithms perform very well in search and optimization problems. Thus they offer a domain independent search method in the process of machine learning. Here

genetic algorithms have been implemented on linguistic rule based classification systems [7].

They can be implemented in learning processes in three alternative approaches.

- Michigan Approach
- Pittsburgh Approach
- Iterative Rule Learning

1.6.1 Michigan Approach

In Michigan approach, each chromosome represents an individual rule and the set of rules represent the population size. Thus each rule is assigned a fitness value. The population of the classifier is maintained and rule discovery process and genetic operations are applied to each individual rule. With time, the rules are altered through interaction with the environment. Thus this approach is mainly used in online process and simulated environment.

The algorithm of the Michigan approach is given as:-

1. Specify the parameters which are required i.e. number of linguistic rules or population size, crossover probability, mutation probability etc.
2. Generate an initial population of linguistic rules randomly.
3. Calculate fitness function corresponding to each linguistic rule. Select some linguistic rules from the current population through selection procedure and perform crossover and mutation on them.
4. Remove the worst linguistic rules and replace them with the newly generated rules in the current population.
5. If the stopping condition is not satisfied, then return to step 3 otherwise terminate the algorithm.

The final solution is the population or rule set which has highest classification rate.

1.6.2 Pittsburgh Approach

In Pittsburgh approach, a whole set of rules is encoded as an individual/chromosome. Each substring of the chromosome is represented as an individual rule. The fitness function of a rule set is evaluated here instead of each single rule. A new set of rules is obtained by crossover while mutation leads to new rules. This approach is mainly used in batch processes.

The basic steps of Pittsburgh approach is given below:-

1. Specify the population size or number of rule sets, number of linguistic rules in each rule set, crossover probability, and mutation probability and termination condition.
2. Randomly generate the rule sets containing given number of linguistic rules as an initial population.
3. Assign fitness function to each rule set and perform crossover and mutation on the rule sets selected in the current population.
4. Remove the rule sets which are worst and add the newly generated rule sets in the current population to perform next iteration.
5. Terminate the execution of algorithm if the stopping condition is satisfied else go to step 3.

The final solution is the best rule set selected from the population.

1.6.3 Iterative Rule Learning

It is similar to Michigan approach as each chromosome represents the rule but in contradiction to the Michigan approach, only the best rule is selected and all other remaining chromosomes are discarded. Thus it provides only the partial solution to the process of learning. Therefore GA has to be incorporated within an iterative approach in the following way:-

1. A rule is obtained by applying genetic algorithm in the system.
2. The rule is integrated with the new set of rules.
3. If the set of rules generated are sufficient to solve the problem, then they are considered as final set of rules otherwise go back to step 1.

In this approach, the fitness function of each chromosome is calculated individually without the requirement of cooperation of other chromosomes. Since only one rule is searched in iteration, so the search space is also reduced.

These approaches are the most commonly used traditional approaches but some new heuristic approaches have also been proposed to give more optimized results and perform well on high dimensional dataset.

1.7 Structure of Thesis

The summary of each of the chapter described in the thesis is given below:-

Chapter 2:- It represents the state of art which exhibit the evolution of IDS and the most commonly used dataset of IDS. Besides, the development of fuzzy rule based system and the work done in the field of computational intelligence in intrusion detection system has been evaluated.

Chapter 3:- The problem statement and the goals of the proposed work are illustrated here.

Chapter 4:- It depicts the proposed work which involves the collaboration of genetic algorithm with fuzzy if-then rules and also a compatibility model to increase precision.

Chapter 5:- Here the evaluation parameters and the results obtained in proposed work and its comparison with other approaches are discussed.

Chapter 6:- It renders the conclusion, summary and future scope of the proposed work.

Chapter 2

State of Art

2.1 Evolution of Intrusion Detection System

An intrusion is defined as an encroachment to a person's privacy or organization's important information. Previously various conventional methods such as firewall, encryption techniques have been used to prevent the computer systems from unauthorized use. But these mechanisms were not sufficient enough to prevent the intrusion as hackers and attackers grew more proficient and were able to find vulnerabilities in the network, thus violating computer security policies. Hence, an additional mechanism i.e. intrusion detection system was established and it became a vital component in the field of security infrastructure. The first intrusion detection model was given by Denning in 1987[9]. Since then many intrusion detection models have been constructed to determine the behavior of the network accurately and in an efficient manner.

Porras *et al.* [3] proposed three standards to evaluate the functioning of intrusion detection systems. These are:-

- ***Accuracy***- The attacks should be properly detected with no false alarms i.e. they should not be misclassified or misrepresented.
- ***Performance***- It is the rate of processing of audit data. For real time detection, the performance of IDS must be high.
- ***Completeness***- It is the ability to detect all the attacks. Thus the intrusion detection system must keep on updating itself and must have comprehensive information about the attacks.

The IDS has been classified into various categories by Debar *et.al*. [10, 11] as shown in figure 2.1.

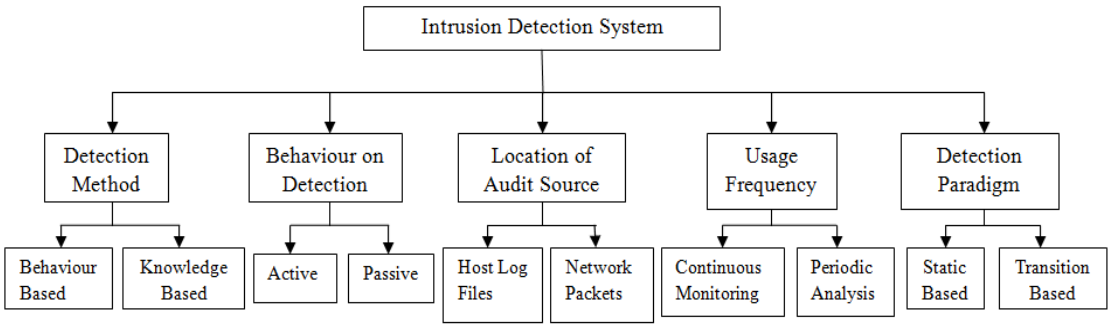

Figure2.1. Classification of Intrusion Detection System

Following is the brief description of each class of IDS-

1. ***Detection Method***- The features or the attributes of the detector is represented in this method. It consists of two schemes- (a) which extracts the information about the attacks from the database and triggers an alarm when vulnerability is evidently found out i.e. misuse detection.(b) when a baseline of the normal behavior is maintained and any abnormality from the baseline is detected as a malicious activity i.e. anomaly detection.
2. ***Behavior on Detection***- It refers to the reaction or conduct of the intrusion detection system. When only alarms are triggered and corresponding to it no action is taken, then the IDS is considered to be passive but when some countermeasures to eradicate or control these attacks are taken, it is said to be active.
3. ***Location of Audit Source***- The input information is examined and on the basis of which the intrusion detection system is classified. The information can be in the form of host log files, application log files or network packets.
4. ***Usage Frequency***- It implies the time analysis of the intrusion detection system in a particular environment. Continuous monitoring performs a continuous time analysis to know about the activities which are happening immediately while the periodic analysis refers to analyzing the activities on a periodic basis.
5. ***Detection Paradigm***- It describes the method used to detect the attacks in IDS. It can be either state based or on the basis of transition from one state to another.

Subsequently various artificial intelligence, machine learning and computational techniques have been applied on various intrusion detection models to obtain precise

results while confronting various problems such as vast network traffic, noisy information, continuous adaptation to changing environment.

2.2 Datasets used in Intrusion Detection System

To evaluate the performance of intrusion detection models, datasets are required which can clearly specify whether the Intrusion Detection System is able to comply with the standards or not. The data can be collected from various sources such as log files, data packets, command sequences etc. Two largely publically available and most used datasets are:-DARPA 98 Lincoln and KDD 99 dataset. In 1999, KDD99 dataset was derived from the DARPA98 network traffic dataset by ACM SIG-KDD International Conference on Knowledge Discovery and Data Mining. It consists of TCP connections and consists of 9 weeks of training data and 2 weeks of testing data. Each connection consists of 41 attributes and the features of the dataset were defined by Stolfo *et al.* [13]. Despite its high usage, the dataset has been criticized by McHugh [12] on the basis of unrealistic data rates.

2.3 Review of Fuzzy Rule Based System

Fuzzy systems with their potential to provide better accuracy and interpretability have the ability to build the models which resemble the real world systems [6]. They are based on fuzzy logic and predicates which includes the knowledge of human experts and consolidates numerical and symbolic representation of data.

Mamdani fuzzy rule based system consists of two parts-fuzzy knowledge base and fuzzy inference system. Ishibuchi *et al.* [6] gave a fuzzy rule based system in which a fuzzy rule structure is matched against the patterns and corresponding to it a class and a discrete value is given as a consequent. The fuzzy rule based classification system can be differentiated on the basis of certainty factor or the confidence consorted to the class in the consequent in the following ways:-

1. On the basis of class label only
2. On the basis of class label and certainty factor
3. On the basis of certainty factor only.

But the major drawback of Mamdani Fuzzy Rule Based System is lack of accuracy in case of complex, high dimensional dataset due to lack of flexibility in linguistic variables. Therefore, various extensions were given in order to increase the accuracy

of Mamdani fuzzy rule based system. Some of them include scatter fuzzy partitions, weighted rules, disjunctive normal form etc.

Genetic algorithms given by Dejong [16] are used in fuzzy rule based system to help in learning process and maintain parameter optimization. To find the best rules, the most important task is to find those parameters which can optimize the Knowledge Base. Karr [17] gave a proposal on genetic fuzzy rule based system which utilized binary coded genetic algorithm on fuzzy system based on triangular functions. Different coding schemes were applied on trapezoidal, triangular and Gaussian functions to obtain a more accurate linguistic fuzzy model. Real coded genetic tuning was given by Cordan *et al.* [18] in Mamdani fuzzy rule based system. Binary coding was used to represent the chromosomes and it was first given by Ishibuchi [4] as the foremost genetic rule selection process.

MOGUL was given by Cordan *et al.* [19] as a multi selection process in genetic algorithm applying iterative rule learning approach and is used in both classification and control processes.

The main GFS learning approaches i.e. Michigan, Pittsburgh, Iterative Rule Learning were used to obtain optimized results. In [39] as proposed by Giordana and Neri , the Michigan and Pittsburgh approaches were collaborated to generate an efficient system consisting of rules which can be used in high dimensional problems as the unnecessary rules can be eliminated by the don't care conditions. Another approach, GCCL (Genetic cooperative-competitive learning) approach was used where a chromosome is encoded by a single rule and each chromosome cooperates and competes to produce results which are more accurate and have good interpretability. Berjlanga *et al.* [35] proposed DNF fuzzy rule based system while a hybrid genetic algorithm has been proposed in [36] which constitute a mathematical model to increase the accuracy and interpretability of each rule.

2.4 Computational Intelligence in Intrusion Detection

Bezdeck [20] was the first who defined computational intelligence and he outlined it as:

A computationally intelligent system is the one which is concerned with only low-level data, incorporate constituents of pattern recognition, and does not apply knowledge in the sense of artificial intelligence; and manifest the following characteristics:-

- computational adaptability
- speed resembling turnaround like humans
- computational fault tolerance
- error rates that is roughly close to human performance

Computational Intelligence consists of four main paradigms as represented in figure 2.2:-

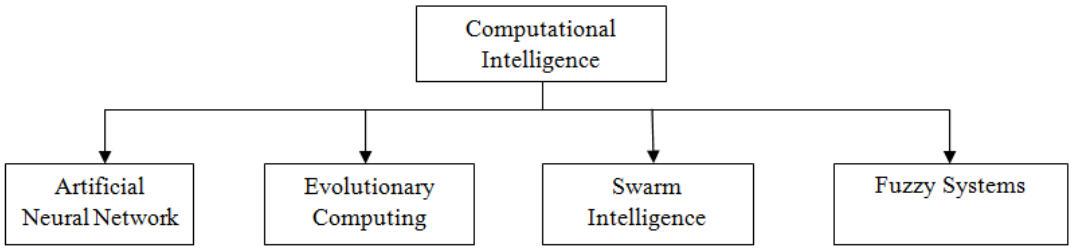

Figure2.2. Prototype of Computational Intelligence

1. ***Artificial Neural Network***- It is an information processing model inspired from the biological neurons. It is an interconnected group of artificial neurons working in unison to solve complex problems. It is considered as the weighted directed graph in which the nodes depict the neurons while the edges represent the interconnections between the neurons. Artificial neural networks are generally used due to their ability to adapt, exhibit fault tolerance, self organization and real time operation.
2. ***Evolutionary Computing***- The evolutionary computation is based on the mechanisms of genetics and natural selection is used in search and optimization problems. It uses an iterative approach where a population is selected from random space and different operations are applied to get a desired solution. It consists of genetic programming given by Koza, evolution strategies given by Rechenberg, evolutionary programming prescribed by Fogel and genetic algorithms given by Holland.
3. ***Swarm Intelligence***-Swarm intelligence is neighbor based system inspired from biological organisms where simple agents manifest collective intelligent neighbor to solve complex problems.

4. ***Fuzzy Systems***- Fuzzy systems as discussed earlier are based on fuzzy logic and deal with vague and imprecise data to get optimized results. They are manly used in control and classification problems.

Different techniques have been proposed to evaluate KDD 99 dataset on the basis of computational intelligence and provide an efficient and fault tolerant intrusion detection system.

Liao and Vumeri [22] proposed k-nearest neighbor method for feature selection in KDD99 dataset. Jiang *et al.*, Zhang [23] worked in artificial neural network on KDD-99 dataset and proved that back propagation as a part of misuse detection has a slight better performance than Radial Basis Function in anomaly detection in terms of detection rate but more training time is required.

Hofman *et al.* [30] gave evolutionary wrapper approach (combining radial basis function with genetic algorithm) and derived the same conclusion on DARPA-98 dataset. The wrapper approach uses genetic algorithm for feature selection while radial basis function networks are utilized for optimization and this approach provides better classification rules.

The 41 attributes rendered by KDD99 dataset were ranked by Mukkamala and Sung [28] through support vector machines. They again combined SVM and neural networks in [29] to rank the features but the time complexity of the whole system increased and the classification was not accurate.

Feature selection based on rough set to process feature selection and genetic algorithm to furnish optimal subsets was introduced by Wang in [26]. The algorithm was examined using SVM classifier for the evaluation of performance.

A multilayered Self Organizing Map was built by Rhodes [24] and Sarasamma *et al.* drawing conclusion that different subsets of features were efficient enough to detect different attacks.

Decision trees, association rules and fuzzy implication table were used to generate fuzzy rules. The collaboration of Hidden Markov Model with fuzzy inference engine was done by Cho [37] to detect normal connections. Fuzzy C-Means and Fuzzy C-Medoids algorithms are two clustering approaches used to detect abnormal behavior through the concept of outliers. Gomez *et al.*[32] first demonstrated the work of fuzzy classifiers on intrusion detection system.

Middlemiss and Dick [21] used KNN in conjunction with genetic algorithm to classify the normal and abnormal behavior. A genetic algorithm was implemented to generate weights for the attributes and the k-nearest neighbor classifier was used for the fitness function of genetic algorithm. The results were more accurate than the simple k-nearest approach but the results were mainly dependent on training data and some of the new attacks in the testing data were not classified accurately.

Stein [27] utilized decision tree classifiers with genetic algorithm in intrusion detection approach with the aim to diminish false alarm rates in network intrusion detection. Genetic algorithm along with naïve bayes classification was used by Lee *et al.* [25] to generate optimized results for classification on KDD99 dataset. Lu *et al.* [31] applied genetic algorithm to decide the number of clusters based upon Gaussian Mixture Models.

REGAL, a distributed genetic algorithm given by Mischiatti and Neri [38] used combination of Pittsburgh and Michigan learning approach to model network traffic. Linear Genetic Programming outperforms Support Vector Machines and Artificial Neural Network in terms of detecting intrusion detection with accuracy as proposed by Abraham *et al.*[41] and Song [40].

Artificial Immune System can be used to model intrusion detection system and the first Artificial Immune System model based on anomaly detection, was given in [42] to detect file alterations and call sequences.

Swarm Intelligence technique was also used in intrusion detection due to its self organizing and distributed properties to get optimized results but suffered from the problems of clustering high dimensional network data. An ant based clustering and sorting algorithm given by Deneubourgh *et al.* [43] was used by Romos and Abraham [44] to detect intrusion in KDD-99 dataset. A standard Particle Swarm Optimization technique was engrafted in genetic fuzzy system by Abadeh *et al.*[34]

Soft Computing techniques soon started to be hybridized to built fault tolerant, precise and robust intrusion detection systems. Since there are five classes in KDD-99 dataset, hence a five neuro-fuzzy classifier was developed by Toosi *et al.*[45]. Fuzzy cognitive map was introduced by Kosko to provide a graphical representation of the work and was used by Xin [46] to detect complicated or intricate attacks.

Tsang [47] highlighted the illustration of fuzzy if then rules in a genetic fuzzy system. Different approaches such as Michigan, Pittsburgh and Iterative Rule Learning were used by Abadeh *et al.* [33] to detect attacks in the network infrastructure.

Thus, soft computing models were more effective in building accurate, robust system with high performance.

In the proposed approach, hybrid genetic fuzzy rule based system is projected utilizing the technique by [35, 36] which is practiced in high dimensional problems to establish a network intrusion detection system which distinctly classify normal class and the different types of attacks prevalent in KDD99 dataset.

Chapter 3

Problem Statement

3.1 Barriers in the Previous work

The KDD-99 dataset constitutes of five major classes. But two of them i.e. U2R and R2L are very less in number in the training dataset. So these classes are not properly trained to get accurate results and perform poorly in the testing datasets which consists of 11 different types of attacks. These attacks are very difficult to be detected and therefore a major drawback in existing IDS.

Wu *et al.* analyzed different approaches and proved that soft computing techniques perform better than other techniques. The computational intelligence approach performed better than the decision trees. Evolved classification rules did not perform well because when overlapping occurs, the data cannot be separated into two classes. Also it is easier to apprehend the fuzzy rules alone.

Self organizing maps suffered from problems such as high dimensionality, higher detection rates with false positives and computational overhead. Evolutionary computing techniques do not have fair termination criteria and does not give accurate results when the data distribution is unbalanced.

Swarm Intelligence techniques are mainly used to learn clusters and classification rules but prove to be a constraint in high dimensional network and cannot differentiate dissimilar objects.

Therefore collaboration of various soft computing techniques is required which has the ability to learn in an uncertain and imprecise network. It encloses all the complementary features of different techniques and builds a robust and fault tolerant system.

3.2 Problem Statement

Computational intelligence systems have the ability to adapt, exhibit fault tolerance, high computational speed and error resilience against noisy information. The fuzzy logic constructs precise and flexible patterns while the genetic algorithm based on evolutionary computation helps in attaining an optimal solution, thus their collaboration will increase the robustness of intrusion detection system. Therefore a hybrid genetic fuzzy rule based inference engine has been proposed. The proposed

network intrusion detection system will be able to classify normal behavior as well as anomalies in the network accurately.

The fuzzy rule based system performs well in an uncertain and imprecise environment and establishes more concise and pliant patterns which enhances the adaptation capability and robustness of the intrusion detection system and classifies normal and abnormal connections correctly.

Evolutionary computing has the capability to learn with the changing environment and is used in designing optimized fuzzy rules. These fuzzy rules are constructed from the training dataset. Genetic algorithms are applicable in tuning membership functions of the fuzzy sets. The crossover operator interchanges the chromosomes between two parents to get more prominent rule/child while the mutation operator generates new rules. Thus new suspected attacks can also be detected with the adaptive capability. The genetic algorithm continues for specified number of generations and the best rules are extracted. These rules undergo a compatibility model which will yield more precise rules and therefore invigorate the performance of intrusion detection system.

3.3 Objectives of the Proposed Work

The main objective behind the proposed work is to build a model which has high detection rate and low or minimal false alarm rate. It should be accurate and complete to classify all the attacks in their true classes and should exhibit the property of high adaptability i.e. the ability to adjust according to changing behavior of the users and networks and modifying itself for proper functioning.

Thus the proposed model should be fault tolerant in nature. The malicious activities may create faults in the system but the IDS should have the potential to maintain reliability and accuracy in the system so as to prevent the systems from abuse. It should remain intact and update its database with the contemporary information about the network connections and therefore able to detect new anomalous activities.

Chapter 4

Proposed Work

The proposed approach is amalgamation of the fuzzy systems with that of genetic algorithm to bring out a hybridized genetic fuzzy rule based system which provides robust platform to detect intrusions existing in the network distinctly and classifies them into normal and different types of attacks according to their signatures. The work has been performed on KDD-99 data set which is a standard dataset used to detect intrusions in the network.

4.1 Evaluation of KDD-99 Dataset

The KDD-1999 intrusion detection dataset uses a version of database which was prepared in 1998 DARPA Intrusion Detection Evaluation Program (MIT Lincoln Labs) to evaluate their research in intrusion detection. It consisted of 9 weeks of raw TCP dump data as training dataset and 2 weeks of testing dataset. The KDD-99 dataset was used in Third International Knowledge Discovery and Data Mining Tools Competition to prepare an intrusion detector which can identify good or bad connections [48].

The dataset consists of connections and each connection is a sequence of TCP packets containing 41 attributes and labeled with either normal or specific attack type. Here 10% of the KDD-1999 dataset has been used to evaluate the whole process. The 10% KDD dataset consists of 4, 94,021 records /connections in the training data. The training dataset consists of 24 training attack types which are under 4 major classes of attacks while the testing dataset consists of 14 additional attacks to differentiate some signatures and check whether these variants are captured or not to increase efficiency.

Below is the table which consists of classes in which the whole training dataset has been divided and corresponding to each class the subclasses are mentioned.

Table 4.1 Classification of attacks

Sno.	Classes	Subclasses
1	Normal	NA
2	Denial of Service(DOS)	back,land,neptune,pod,smurf,teardrop
3	User to Root(U2R)	buffer overflow, loadmodule, multihop, perl, rootkit
4	Remote to User(R2L)	imap,phf,spy, ftp write, guess_passwd, warezclient, warezmaster
5	Probe	Nmap, portsweep, satan, ipsweep

The features of KDD-99 dataset as defined by Stolfo *et al.* [47] have been classified into following categories-

1. ***Basic Features***- This category enclose all the attributes that can be retrieved from an individual TCP connection and comprise 9 attributes.
2. ***Time Based Traffic Features***- It includes features which are calculated on the basis of time interval and is subdivided into two types:-
    a) ***Same Host Features***-It examines only the connections in the past two seconds that have the same destination host as the current connection.
    b) ***Same Service Features***-It examines only the connections in the past two seconds that have the same service as the current connection.
3. ***Host Based Traffic Features***- Here features were constructed using a window of 100 connections to the same host instead of a time window of 2 seconds.
4. ***Content Based Features***-It consists of 13 features that are extracted from domain knowledge and are used to indicate suspicious behavior in the network or unstructured data portions in the packet.

Table 4.2 describes the 41 attributes of the KDD- Cup 99 dataset and shows whether the feature is of symbolic type or continuous.

Table 4.2 KDD-99 CUP dataset attributes and their data types

S.No	Attributes	Type
1	duration	Continuous
2	protocol_type	Symbolic
3	Service	Symbolic
4	flag	Symbolic
5	src_bytes	Continuous
6	dst_bytes	Continuous
7	land	Continuous
8	wrong_fragment	Continuous
9	urgent	Continuous
10	hot	Continuous
11	num_failed_logins	Continuous
12	logged_in	Symbolic
13	num_compromised	Continuous
14	root_shell	Continuous
15	su_attempted	Continuous
16	num_root	Continuous
17	num_file_creations	Continuous
18	num_shells	Continuous
19	num_access_files	Continuous
20	num_outbound_cmds	Continuous
21	is_host_login	Continuous
22	is_guest_login	Symbolic
23	count	Continuous
24	srv_count	Continuous
25	serror_rate	Continuous
26	srv_serror_rate	Continuous
27	rerror_rate	Continuous
28	srv_rerror_rate	Continuous
29	same_srv_rate	Continuous
30	diff_srv_rate	Continuous

S.No	Attributes	Type
31	srv_diff_host_rate	Continuous
32	dst_host_count	Continuous
33	dst_host_srv_count	Continuous
34	dst_host_same_srv_rate	Continuous
35	dst_host_diff_srv_rate	Continuous
36	dst_host_same_src_port_rate	Continuous
37	dst_host_srv_diff_host_rate	Continuous
38	dst_host_serror_rate	Continuous
39	dst_host_srv_serror_rate	Continuous
40	dst_host_rerror_rate	Continuous
41	dst_host_srv_rerror_rate	Continuous

4.2 Fuzzy Rule Based System

In the proposed work, there are mainly three components- fuzzy rule based system, genetic algorithm and computational model. In this section the fuzzy rule based system to detect intrusions is discussed.

Step1-Representation of data

The 10% KDD-99 dataset consists of 4, 94,021 records in training dataset. Here 750 random samples are taken into account and there are 41 attributes in each record. Thus in a fuzzy rule based system, there are m i.e. 750 labeled patterns with n=41 dimensionality, i.e. each pattern is represented as

$$x_p = \{x_{p1}, x_{p2}, x_{p3}, x_{p4}, x_{p5}, \ldots\ldots\ldots\ldots\ldots, x_{p38}, x_{p39}, x_{p40}, x_{p41}\}$$

where p=1,2....,m training patterns and each pattern is required to be classified into c classes i.e. 5 classes. Here only continuous attributes have been considered for further processing.

Five linguistic variables (L_i) for each feature or attribute have been considered (i.e. Very Low, Low, Medium, High, and Very High) and each feature definition interval has been uniformly partitioned using triangular fuzzy sets. Thus the total number of fuzzy if then rules generated is 5^n where n represents the pattern of n dimensions. But in this case where the number of attributes is very high i.e. n=41, it is impossible to

generate 5^{41} rules. Therefore a heuristic approach is applied to generate fuzzy if-then rules.

Step2- Normalization of data

Since the space of pattern taken into account is $[0,1]^n$, so the attribute values of each pattern is normalized in the range of [0,1] i.e. $x_{pi} \in [0,1]$ where the formula for normalization is given by:-

$$y = \frac{x - min}{max - min} \tag{4}$$

Where *x* represents the original attribute value, min represents the minimum boundary value i.e. 0, max represents the maximum boundary value i.e. 1 and *y* represents the normalized value i.e. $0 \leq y \leq 1$.

Step3-Calculating membership function

After normalizing the data, the membership function of every feature of each training pattern is determined by the following formula:-

$$\mu_j(x) = max\left\{0, 1 - \frac{|x - x_j|}{v}\right\} \tag{5}$$

Where, *x* represents the normalized value of each feature,

$$x_j = \frac{j-1}{L-1} \tag{6}$$

where j=1,2....L,

and

$$v = \frac{1}{L-1} \tag{7}$$

where *L* represents the number of linguistic labels. Here *L* varies from 1-5 and help in calculating the membership function of each attribute.

Step4-Determining compatibility of each training pattern

The compatibility of each training pattern x_p is calculated with the fuzzy if then rule R_j by using the following formula:-

$$\mu_j(x_p) = \prod_{i=1}^{n} \mu_{ji}(x_{pi}) \tag{8}$$

Where p refers to 1,2.....,m training patterns and $\mu_{ji}(x_{pi})$ is the membership function of the i^{th} attribute of p^{th} pattern.

Step5- Finding compatibility grade for each class

After obtaining the compatibility of each training pattern, relative sum of compatibility grades of the training patterns with rule R_j for each class is calculated. This is given by:-

$$\beta_{Class\ h}(R_j) = \frac{\sum_{x_p \epsilon Class\ h} \mu_j(x_p)}{N_{Class\ h}} \quad (9)$$

where $\beta_{Class\ h}(R_j)$ represents the mean sum of compatibility grades in *Class h* with fuzzy if then rule,

$N_{Class\ h}$ is the number of training patterns taken into consideration corresponding to each class h and h ranges from 1,2,…..,c. where c=5 i.e. there are mainly five classes- normal, dos attacks, u2r attacks, r2l attacks and probe attacks.

Step6-Selecting fuzzy if-then rule for a particular class

A fuzzy if then rule is selected according to the given training patterns. The consequent class C_j for a given rule R_j is calculated as follows:

$$\beta_{\text{Class h}}(R_j) = \max_{\text{h}=1,..,\text{c}}(\beta_{\text{Class h}}(\text{R}_\text{j})) \quad (10)$$

The maximum of $\beta_{\text{Class h}}(R_j)$ is evaluated and the one with the maximum value is considered to be the class of that fuzzy if then rule R_j. If the maximum value comes out to be true for more than one class, then the consequent class C_j cannot be determined uniquely and is taken as φ and the corresponding rule is rejected.

Step7-Assessing the certainty degree CD_j

The confidence or certainty degree refers to how precisely an input pattern is classified by the fuzzy rule based system i.e. determining the authenticity of the class with a particular confidence value. It is determined by the following:-

$$CD_j = \frac{\beta_{Class\ h_j}(R_j) - \overline{\beta}}{\sum_{h=1}^{c} \beta_{Class\ h_r}(R_j)} \quad (11)$$

Where

$$\overline{\beta} = \frac{\sum_{\text{h} \neq \text{h}_\text{r}} \beta_{\text{Class h}}(\text{R}_\text{j})}{\text{c}-1} \quad (12)$$

The certainty degree lies in the unit interval [0, 1]. If the consequent class C_j is φ , then the confidence is also φ, this determines the absence of validity of that class in

that particular rule. The value of CD_j =1 represents very high confidence which denotes that the rule belongs to that specific class.

Step8- Generation of fuzzy if then rules

Thus the fuzzy if then rules are generated in the following manner:-

Rule R_j=If x_1 is A_{j1} and x_2 is A_{j2} and ....., x_n is A_{jn} , then the class is C_j with CD_j , j=1,2...N

where R_j is the label of the j^{th} fuzzy if then rule, A_{j1}, A_{j2}, A_{jn} denotes the antecedent fuzzy sets, C_j represents the consequent class, CD_j refers to the certainty degree or the confidence in the class label and N is the number of rules.

For example- If protocol is 1.0,.....,and dst_host_srv_count is 1.0,.....,and dst_host_srv_diff_host_rate is 0.92,.....and dst_host_srv_rerror_rate is 1.0 ,then the class is r2l attack with certainty degree=1.0.

4.3 Genetic Algorithm to generate Classification Rules

Genetic Algorithm is very successful in optimization and search problems. This algorithm exploits the historical information in order to get optimized results. Following are the steps followed in genetic algorithm to generate classification rules:-

Step1-Binary Encoding

Before applying a genetic algorithm, the potential solutions are required to be encoded. The most commonly used encoding scheme is Binary Encoding. It is one of the most frequently used encoding schemes because of its simplicity. Let the population size be N_{pop}, i.e., the number of fuzzy if then rules to be generated. Here the population size is taken as 30.

In genetic algorithm, each individual of the population is represented by chromosome. Here each chromosome represents a rule for a specific class. Since each feature or attribute is divided into five linguistic labels, so the length of the chromosome is equal to the product of the number of features and the number of linguistic labels.

Each chromosome is a set of genes. Here the first five genes represent first feature, next five second feature and so on. Figure 4.1 represents the sample chromosome structure.

Each rule is expressed as:-

R_j= If x1 is l_1^2 or l_1^5, x2 is l_2^1 and x3 is l_1^3, then the class is C_j with CD_j.

L_1^1	L_1^2	L_1^3	L_1^4	L_1^5	L_2^1	L_2^2	L_2^3	L_2^4	L_2^5	L_3^1	L_3^2	L_3^3	L_3^4	L_3^5
0	1	0	0	1	1	0	0	0	0	0	0	1	0	0

Figure4.1 Sample chromosome structure

For example- If protocol is l_1^1,.....,and dst_host_srv_count is l_{33}^2,.....,and dst_host_srv_diff_host_rate is l_{37}^2or l_{37}^3,.....and dst_host_srv_rerror_rate is l_{41}^1,then the class is r2l attack with certainty degree=1.0. Figure 4.2 represents the chromosome of r2l attack with its attributes.

L_1^1	L_1^2	L_1^3	L_1^4	L_1^5	.....	L_{33}^1	L_{33}^2	L_{33}^3	L_{33}^4	L_{33}^5
1	0	0	0	0		0	1	0	0	0

L_{37}^1	L_{37}^2	L_{37}^3	L_{37}^4	L_{37}^5	.....	L_{41}^1	L_{41}^2	L_{41}^3	L_{41}^4	L_{41}^5
0	1	1	0	0		1	0	0	0	0

Figure4.2 Chromosome of r2l class

Step2-Calibrate fitness value for each if-then rule

Let the number of fuzzy if-then rules taken into consideration in the population is denoted by N_{pop}. Now, the fitness function is evaluated for each fuzzy if then rule by classifying all the training patterns with the N_{pop} i.e. 30 fuzzy if then rules generated. The formula of the fitness function is given as[21]:-

$$Fitness = w_1 \times CD_j + w_2 \times (1 - var_N) + w_3 \times (1 - lab_N) + w_4 \times Rule_j \quad (13)$$

where CD_j represents the confidence for each fuzzy if-then rule,

var_N are the normalized values of the number of features(min=1 and max=n_v),

lab_N denotes the normalized value of linguistic labels used. (Here minimum value of lab_N is 1 and maximum value is n_v*(l_i-1).)

$Rule_j$ is given as:-

$$Rule_j = \frac{n_j}{N} \quad (14)$$

where n_j is the number of samples covered by the rule.

N is the total number of training samples.

w_1, w_2, w_3 and w_4 represents the weights and $w_1 + w_2 + w_3 + w_4 = 1$.

Hence from the above example- If protocol is l_1^1,.....,and dst_host_same_srv_rate is l_{33}^5,.....,and dst_host_srv_diff_host_rate is l_{37}^2or l_{37}^3,.....and dst_host_srv_rerror_rate is l_{41}^1,then the class is r2l attack with certainty degree=1.0 and fitness=0.5013.

Step3-Selection mechanism

Now, a steady state selection mechanism is applied to get top 10% of the population which has high fitness value. On the rest of the population, rank based roulette wheel selection mechanism is applied.

In the rank selection method, the population is sorted according to their fitness value; the worst chromosome is ranked 1 and the best with rank N. Here, rank is calculated by the following formula:-

$$P_{rank}(i) = \frac{(2-s)}{\mu} + \frac{2i(s-1)}{\mu(\mu-1)} \quad (15)$$

Where μ represents the rank of fittest individual and i denotes the rank of the current individual.

Thus, from the present population, pair of fuzzy if then rules is chosen to produce new fuzzy rules for next generation.

Step-4 Crossover operation

Now a crossover operation is applied on the randomly selected fuzzy if-then rule.

The cross over operation on the fuzzy if then rule is performed with specific crossover probability and is given as:-

Parent 1-If X_1 is l_1^1 and X_2 is l_2^3 and X_4 is l_4^2, then the class is C2 with $CD_{j1.}$

Parent 2-If X_1 is l_1^1 and X_2 is l_2^3 and X_4 is l_4^4, then the class is C2 with CD_{j2}

Offspring-If X_1 is l_1^1 and X_2 is l_2^3 and X_4 is l_4^2or l_4^4, then the class is C2 with CD_{j3}.

The consequent class of the generated offspring is determined. If the class is same to that of the corresponding parents and covers more samples then the offspring is accepted otherwise it is rejected and the process repeats until we obtain the desired results.

For example- The two chromosomes of class r2l are selected to perform crossover operation as shown in figure 4.3.

Parent1-

L_1^1	L_1^2	L_1^3	L_1^4	L_1^5	.....	L_{33}^1	L_{33}^2	L_{33}^3	L_{33}^4	L_{33}^5
1	0	0	0	0		0	1	0	0	0

L_{37}^1	L_{37}^2	L_{37}^3	L_{37}^4	L_{37}^5	.....	L_{41}^1	L_{41}^2	L_{41}^3	L_{41}^4	L_{41}^5
0	1	1	0	0		1	0	0	0	0

Parent2-

L_1^1	L_1^2	L_1^3	L_1^4	L_1^5	.....	L_{33}^1	L_{33}^2	L_{33}^3	L_{33}^4	L_{33}^5
1	0	0	0	0		1	0	0	0	0

L_{37}^1	L_{37}^2	L_{37}^3	L_{37}^4	L_{37}^5	.....	L_{41}^1	L_{41}^2	L_{41}^3	L_{41}^4	L_{41}^5
1	0	0	0	0		1	0	0	0	0

Offspring-

L_1^1	L_1^2	L_1^3	L_1^4	L_1^5	.....	L_{33}^1	L_{33}^2	L_{33}^3	L_{33}^4	L_{33}^5
1	0	0	0	0		1	1	0	0	0

L_{37}^1	L_{37}^2	L_{37}^3	L_{37}^4	L_{37}^5	.....	L_{41}^1	L_{41}^2	L_{41}^3	L_{41}^4	L_{41}^5
1	1	1	0	0		1	0	0	0	0

Figure4.3 Crossover operation performed on two chromosomes of r2l class

The generated offspring covers more features and therefore has a better certainty degree and more samples are covered in a single rule.

Step5-Mutation Operation

Some of the features are randomly chosen from the chromosome and bit flip mutation is applied to it with pre-specified mutation probability. It means the value of the selected gene will be converted to 1 if its value corresponds to 0 or vice versa. If the consequent class before the application of mutation is same after the mutation operation, then the mutant chromosome is accepted otherwise this step is repeated until a specific number (M_R).

For example- in r2l attacks the 5th bit of 33th attribute is flipped to 1 to get desired results as depicted in figure 4.4.

L_1^1	L_1^2	L_1^3	L_1^4	L_1^5	.....	L_{33}^1	L_{33}^2	L_{33}^3	L_{33}^4	L_{33}^5
1	0	0	0	0		0	1	0	0	1

L_{37}^1	L_{37}^2	L_{37}^3	L_{37}^4	L_{37}^5	.....	L_{41}^1	L_{41}^2	L_{41}^3	L_{41}^4	L_{41}^5
0	1	1	0	0		1	0	0	0	0

Figure 4.4 Mutation operation performed on a chromosome of r2l class

Step-6 Heuristic Approach

A local heuristic approach is applied on the gene pool where a chromosome is selected and value of the gene is randomly changed. If the obtained results are better than the previous one, then that rule is taken as an offspring otherwise it is rejected. Also the chromosomes with worst fitness function are removed to obtain good results. The process terminates when the specified number of generations are completed.

Step7-Termination

The newly generated rules with better fitness value and more amount of sample covering are substituted with old bad rules. The whole process continues until a specific termination condition is given. Here the termination condition is the number of generations i.e. 100.

4.4 Mathematical Model

After performing all the above steps, a new rule set is obtained. Now a mathematical model is applied on it to obtain more accurate and optimized results. The major focus is that the generated rules cover almost all the training samples. Here are the following sets and parameters available on the basis of which more samples will be covered in the fuzzy if then rules and accuracy will be maximized as given by [36]:-

- C -represent set of classes
- F-set of features
- S-represent set of samples
- R-represent set of rules
- μ_{sr} -represent membership function of sample s for rule r.

- α_r-represent accuracy value of rule r
- C_s- denotes the class label for the sample s
- C_r-represent class label for rule r
- $M-$a very high number

Decision variables are given by:-

- $x_r = \begin{Bmatrix} 1 & if\ rule\ r\ is\ selected \\ 0 & else \end{Bmatrix}$ (16)
- $y_{sc} = \begin{Bmatrix} 1 & if\ sample\ s\ is\ classified\ as\ class\ c \\ 0 & else \end{Bmatrix}$ (17)

The model is given by:-

Step1- Classification of samples

More number of samples should be classified correctly. Therefore each sample must be classified into their correct class otherwise any attack can be misclassified as normal and may pose a serious threat to the network and this is done by maximizing the number of correct classification.

$$Maximize\ Z = \sum_{s\in S, c\in C, C_s=c} y_{sc} \quad (18)$$

while satisfying the following conditions:-

$$\sum_{r\in R:C_r=c} x_r \geq 1 \ \ \forall c \in C \quad (19)$$

which represents that at least one rule is selected from each class and the value of $x_r = \{0,1\}\ \forall r \in R$

$$\sum_{c\in C} y_{sc} \leq 1 \ \ \forall s \in S \quad (20)$$

Equation 20 checks that each sample should be classified in maximum one class and should not show ambiguous behavior. To obtain the value of y_{sc} steps2 and 3 are followed.

Step2-Calculating accuracy of the rule

To obtain the preciseness of each selected rule, the accuracy is calculated as:-

$$Accuracy_s \geq \sum_{r\in R:C_{r=c}} \mu_{sr}\, \alpha_r x_r\ \forall s \in S, \forall c \in C \quad (21)$$

where $Accuracy_s \geq 0\ \forall s \in S$ is a constraint.

Step3-Assurance of correct class for each sample

The following formula ensures that for each rule, the corresponding class selected is correct.

$$y_{sc} \leq 1 - \left(\frac{1}{M}\right)\left(Accuracy_{sc} - \sum_{r \in R: C_{r=c}} \mu_{sr}\, \alpha_r x_r\right) \forall s \in S, \forall c \in C \qquad (22)$$

For the selected class, $Accuracy_{sc} - \sum_{r \in R: C_{r=c}} \mu_{sr}\, \alpha_r x_r = 0$. Hence $y_{sc} \leq 1$.

But for the unselected class, $Accuracy_{sc} - \sum_{r \in R: C_{r=c}} \mu_{sr}\, \alpha_r x_r \geq 0$. Therefore, $y_{sc} < 1$. By the integrality constraint, $y_{sc} = \{0,1\} \forall s \in S, \forall c \in C$, thus for the unselected rule, $y_{sc} = 0$ and for the selected rule $y_{sc} = 1$.

The constraint $y_{sc} \leq \sum_{r \in R: C_r = c} x_r \;\; \forall s \in S, \forall c \in C$ prevents the classification of unselected rule for each sample.

Hence the main objective is to maximize Z and thus cover approximately all the samples of training dataset. Therefore the feasibility of obtaining the best rule set will be maximized which will improve the classification of attacks in testing dataset as more features will be available for classification.

The overall steps have been given in the following flowchart in figure 4.5:-

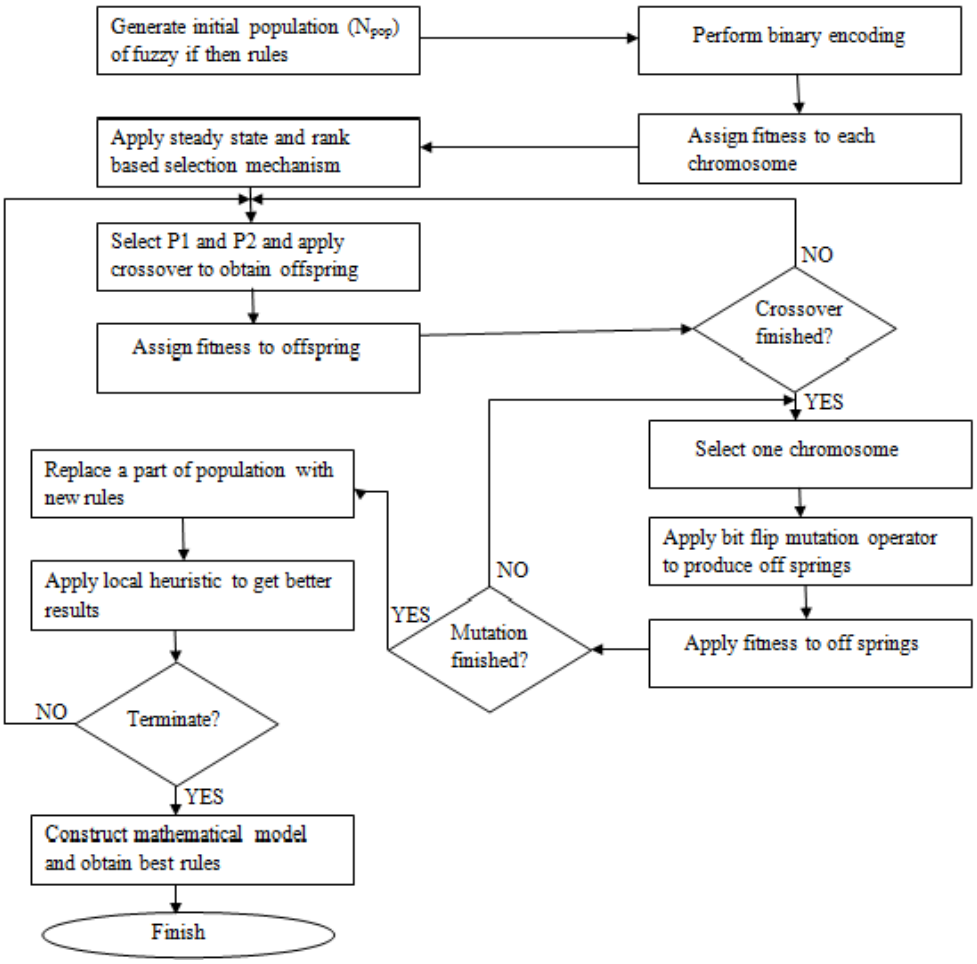

Figure4.5. Flow chart of Hybrid Genetic Fuzzy System

Chapter 5
Simulation Results

The experiment has been laid down on KDD-99 dataset to set up an intrusion detection system and was implemented in Python 2.7. This dataset contains standard audit data including a large number of intrusions. In the proposed approach, 10% of the KDD99 dataset has been used i.e. the total number of records are 494,021. The proposed hybrid genetic fuzzy rule based system consists of rules which are generated through training dataset while the tests are performed on testing dataset to validate the performance of the rules.

In the given experiment, 750 samples were randomly chosen from the training dataset and the value of each attribute was normalized in the unit interval [0,1]. Fuzzy if-then rules were generated of a particular size as an initial population and then various genetic operations were applied to obtain the best rules. The process continues until a termination condition is satisfied. The parameters which were used to make a genetic fuzzy rule based inference engine are given below:-

Table 5.1 Parameters needed for fuzzy inference engine

S.No	Parameter	Assigned value
1	Population Size(N_{pop})	30
2	Crossover probability(P_c)	0.9
3	Mutation probability($P_{m)}$	0.1
4	Mutation rate(M_R)	20
5	Maximum number of generations	100

The hybrid genetic fuzzy based intrusion detection system is formed using the above parameters and a pool of about 100 rules is obtained which is checked with the testing dataset to affirm the rules developed.

The rules are checked on randomly selected testing connections and the conclusions were drawn about the feasibility of the genetic fuzzy intrusion detection system through various parameters.

5.1 Evaluation Parameters

The IDS is basically evaluated through the confusion matrix or contingency table which was given by Provost and Kohavi. The matrix contains information about the

actual and predicted classifications done by the system. It consists of the following records:-

- True Negative (TN) – It refers to the number of correct events which predicted them as authentic connections.
- False Positive (FN) – It implies the number of erroneous predictions which analyzed genuine events as fake events.
- False Negative (FN) – The number of incorrect predictions which evaluated false connections as correct connections.
- True Positive (TP) – It refers to the number of correct predictions which anticipate that the instance is fake or anomalous.

Here is the confusion matrix which clearly specifies the outcomes.

Table 5.2 Confusion Matrix

		Predicted Class	
		Negative	Positive
Actual Class	Negative	TN	FP
	Positive	FN	TP

Therefore for this confusion matrix, various standards have been evaluated to measure the performance of IDS.

- Accuracy- It is described as the ratio of total number of connections that were correct and is given by the formula:-

$$Accuracy = \frac{TN+TP}{TN+FP+FN+TP} \quad (23)$$

- True Negative Rate-It is defined as the ratio of false or negative connections that were classified precisely. It is also known as specificity and is calculated as:-

$$TNR = \frac{TN}{TN+FP} \quad (24)$$

- True Positive Rate- It is the proportion of actual events that were classified correctly. It is also known as detection rate or sensitivity or recall. It is given as:-

$$TPR = \frac{TP}{TP+FN} \quad (25)$$

- False Positive Rate- It is the ratio of authentic events that were misclassified in another class. It is also referred to as False Alarm Rate and is calculated as:-

$$FPR = \frac{FP}{TN+FP} \quad (26)$$

Or

$$FPR = 1 - TNR \quad (27)$$

- False Negative Rate-It is the proportion of faux connections that were incorrectly classified as normal events and is denoted by:-

$$FNR = \frac{FN}{TP+FN} \quad (28)$$

Or

$$FNR = 1 - TPR \quad (29)$$

5.2 Results

The proposed approach is applied and tested on testing dataset to recognize the efficiency and effectiveness of the hybrid genetic fuzzy rule based IDS.

Here are the results obtained from the confusion matrix:-

Table 5.3 Confusion Matrix of the Proposed Approach

Actual Class	Predicted Class					
	Normal	DoS	U2R	R2L	Probe	Total
Normal	52010	279	465	2536	1579	56908
DoS	5625	214765	-	248	5199	225876
U2R	-	-	40	6	1	47
R2L	422	-	224	14185	815	15646
Probe	11	3	1	168	3206	3389

The results obtained are analyzed for different set of classes on basis of different parameters and is given as:-

Table 5.4 Outcomes of different parameters on different types of classes

Classes	TNR	TPR	FPR	FNR	Accuracy
Normal	97.45	91.45	2.5	8.54	96.30
DoS	99.59	97.09	0.405	2.90	96.15
U2R	99.75	86.10	0.242	13.89	99.75
R2L	98.91	91.66	1.08	8.33	98.46
Probe	97.36	94.60	2.63	5.39	97.33

The recall or detection rate for each class comes out to be of admirable value especially for the u2r and r2l classes where different algorithms do not succeed in getting a high precision. The false positive rate or false alarm rate is meager not exceeding 2.5 which depict more accuracy and lower misclassification of connections.

Different approaches such as Michigan, Pittsburgh and Iterative Rule Learning have also been previously applied on KDD-99 dataset to generate robust IDS. The detection rate for these approaches have been adopted from Abadeh *et al.* who have tested them on 10,000 randomly selected training dataset. The results of these existing genetic fuzzy systems have been compared with the current approach and validated to check whether the consequences or the outcomes are more accurate and better than traditional approaches. Following is the graph which represents the consequences of different schemes on each class.

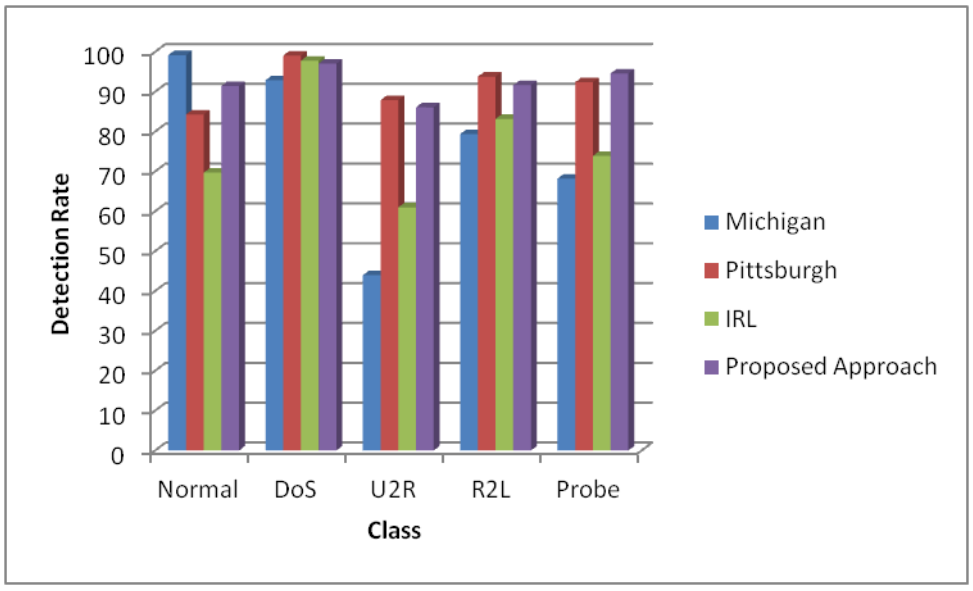

Figure5.1 Evaluation of Detection Rate on each class for different approaches

The Michigan approach performs well in identifying normal class as compared to all other approaches with high percentage of 99.2 but does not function as well in other classes. The proposed approach provides output of around 91% in normal connections. The Pittsburgh approach achieves high detection rate in case of U2R of around 87% and 93% in R2L. The Michigan and IRL approach performs poorly in detecting U2R class with recall of around 44% and 61%. Even in R2L the detection rate is at lower rate of around 79% in case of Michigan approach and 83.2 % in Iterative Rule Learning. The intended methodology gives nearly equivalent results in comparison to Pittsburgh approach with True Positive Rate as 86% in U2R and 92% in R2L. The analogous behavior is shown by all the approaches in case of DoS attacks. The intended approach outperforms all the existing genetic fuzzy approaches with recall of 94.6% while Michigan and Iterative Rule Learning performs poorly with recall of 68.2% and 73.9%. The graph clearly depicts that the purported methodology outdoes Pittsburgh approach in case of detecting normal and probe class and give equivalent results in case of other classes too. Therefore, the proposed genetic fuzzy inference engine is a proficient approach which can be used to detect intrusions in network.

Now, the performance of different algorithms used in the past has been compared with the proposed approach on the basis of two parameters i.e. overall detection rate and false alarm rate.

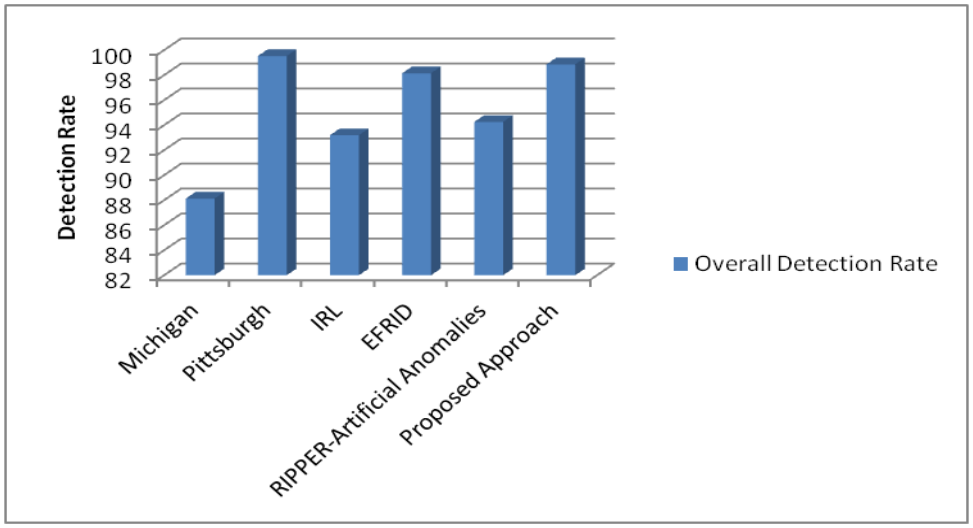

Figure5.2 Comparison of overall detection rate among different approaches

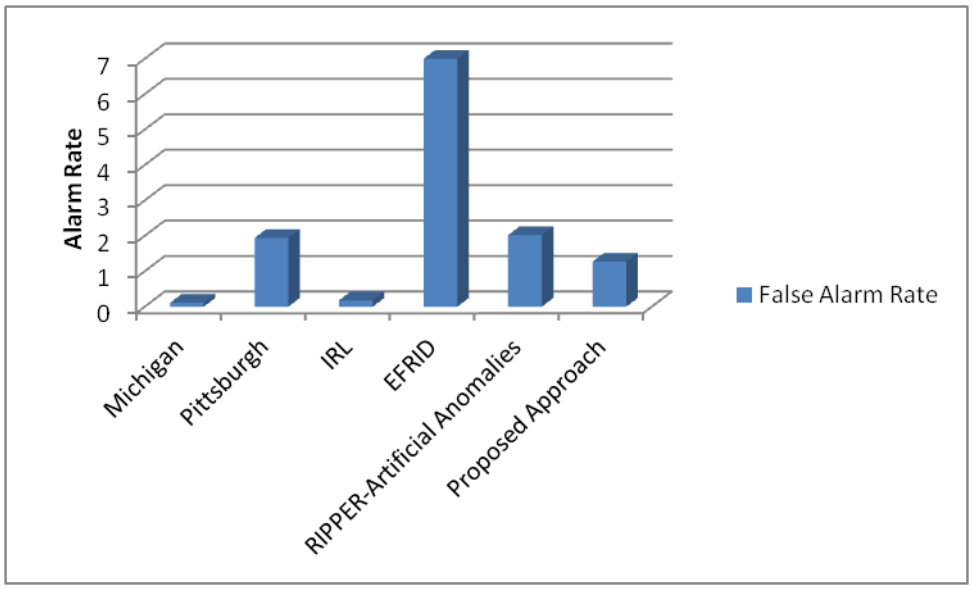

Figure5.3 Comparison of False Alarm rate among different approaches

Different methods such as RIPPER, EFRID and other genetic fuzzy systems have been used to compare with the current approach and perceive which approach has overall better detection rate and lower false alarm rate. The Michigan algorithm and IRL approach perform moderately in detecting class with 88.13% and 93.2% but very low false alarm rate of 0.11% and 0.18% which depicts a very less possibility of misclassification. The Pittsburgh approach has a very high recall which shows its efficiency in detecting the attacks but the false positive rate is comparatively higher rounding about to 2%. The EFRID approach has a good detection rate but exorbitant false alarm rate of 7%. Similarly the RIPPER approach has high false positive rate of 2.02%. The proposed approach gives more adept result with high recall value rounding to about 99% which is nearly equal to Pittsburgh approach and false alarm to about 1.27% which is comparatively very less than Pittsburgh approach.

Hence the genetic fuzzy systems perform more skillfully than other existing approaches and are more reliable. The proposed methodology gives high performance than all other genetic fuzzy systems with suitable detection rate and meager false alarm rate, therefore stabilizing all the parameters.

5.3 Evaluation of Results through Snapshots

This section contains the snapshots which provide clear view about the anomaly detection scheme implemented.

To detect whether the incoming connection is normal or some type of attack, the connection is submitted in the anomaly detection system. Figure 5.4 represents the submission of connection for its correct detection.

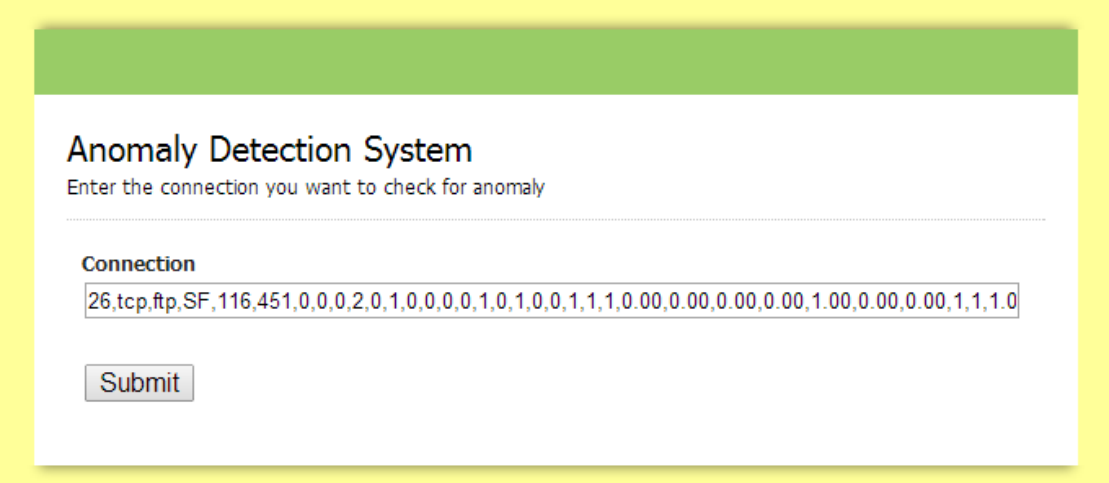

Figure5.4. Entering the connection in the system

After submission, the intrusion detection system refers to the rules made and detects the category of connection whether it is normal or malfunctioned. Figure5.5 depicts the classification of connection as an r2l attack on the system.

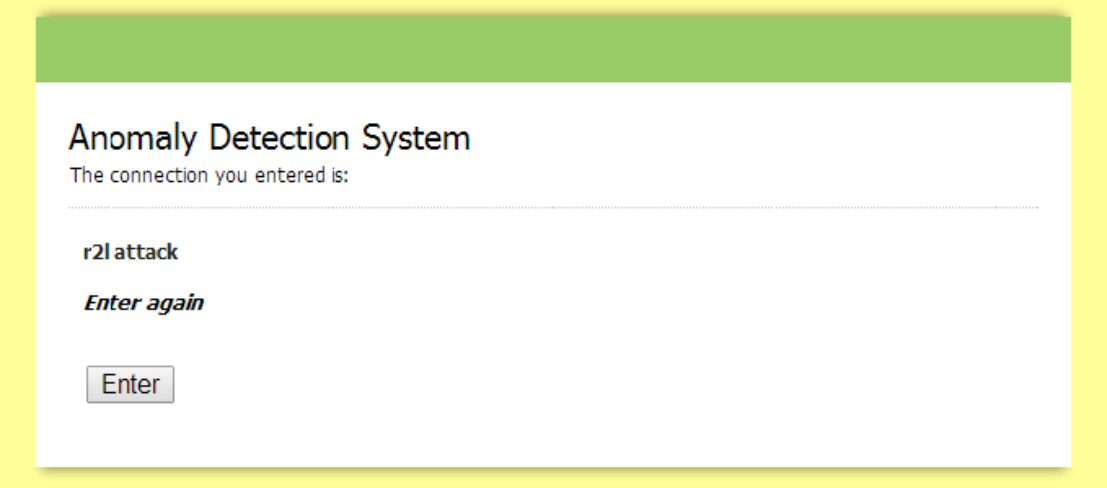

Figure5.5. Retrieval of result corresponding to the connection

To check the preciseness or accuracy of the rules formed from the genetic fuzzy intrusion detection system, the testing files corresponding to each attack is uploaded and thus validity of the rules is checked. Figure5.6 demonstrates the uploading of a file corresponding to any type of attack. Here probe attack is being chosen.

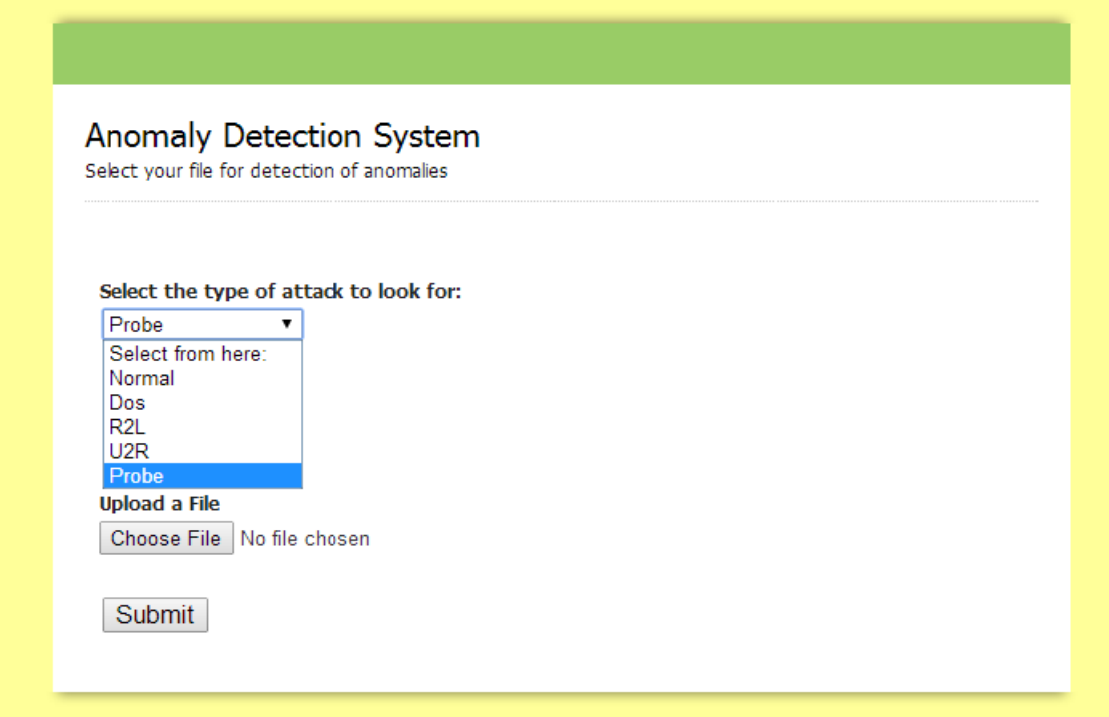

Figure5.6. Selecting the type of attack for which file is required to be uploaded

Figure5.7 represents the uploading of file corresponding to probe attack to check the validity of rules.

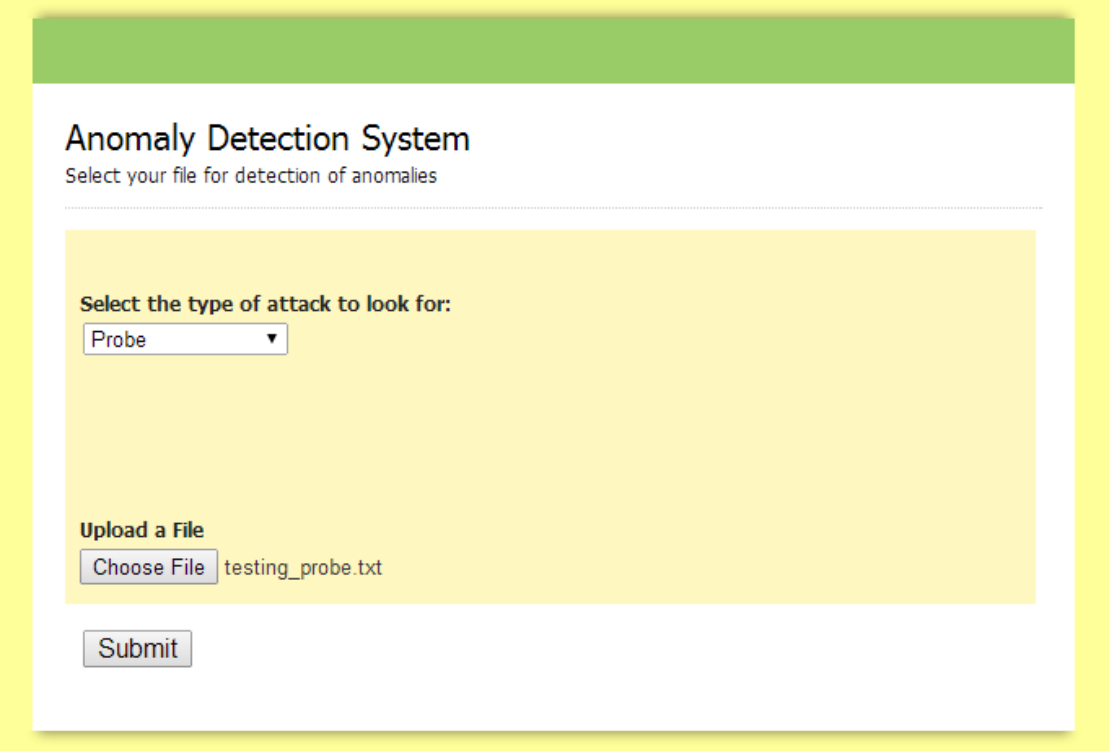

Figure5.7. Uploading of file corresponding to the attack chosen.

Now, each connection is the testing dataset is checked with the rules to determine the authenticity and effectiveness of the rules and to determine whether the rules are able to clearly classify each connection or not. Figure5.8 depicts that when the testing dataset of probe was uploaded, out of 3389 connections, 3206 were correctly classified as probe attacks while in other connections, 11 were misclassified as normal, 3 as dos, 1 as u2r attack and 168 were reported as r2l attacks.

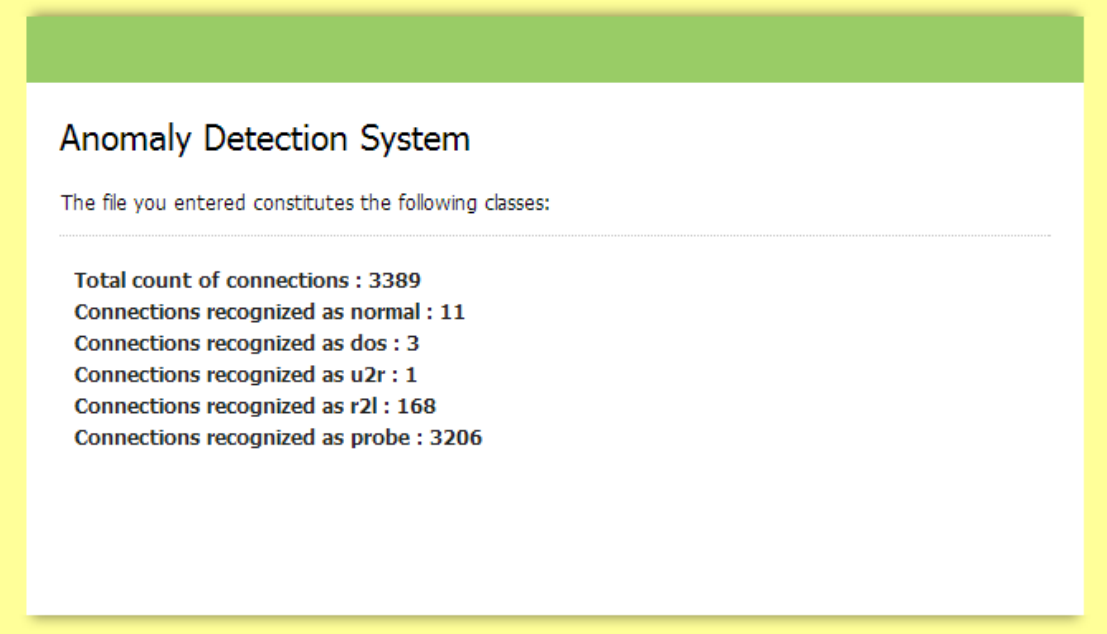

Figure5.8. Results corresponding to the testing dataset.

Chapter6

Conclusion and Future Scope

6.1 Conclusion

With the hybridization of genetic algorithm and a fuzzy rule based system, an efficient approach has been designed to detect intrusions in an effective manner.

The proposed approach gives more reliable and efficient results than existing genetic fuzzy systems. With the establishment of mathematical model, best rules are preserved, thus providing a good approach to detect attacks. Thus the recall or detection rate is very high while the false alarm rate is very low, which are most important aspects to detect the quality of Intrusion Detection Systems.

The current approach is able to detect U2R and R2L attacks with high recall value which the other approaches fail to do so due to their less number in training dataset as shown in figure5.1.

The intrusion detection system is able to classify the connections into their correct classes i.e. a high value of accuracy is achieved which has been shown in the Table 5.4.

The genetic operators guarantee substantial individuals, as the class of generated rules is matched with their parent class. If they are same, then only the newly generated rule is accepted, otherwise it is rejected and the whole process is repeated. This approach reduces misclassification of rules and thereby increases accuracy.

The genetic algorithm is continued for some specific generations which provide more validity and accuracy to the rules. The fuzzy if then else rules perform exceedingly well on imprecise and uncertain data, therefore the system is fault- tolerant.

As the rules keep on updating themselves in the database with the changing connections therefore the intrusion detection system has the property of high adaptability.

Thus, the classification rules are able to classify the normal and abnormal behavior in the network with good accuracy, thus leaving fewer loopholes for the misjudgment.

6.2 Summary

The proposed approach intermingles the features of fuzzy rue based system and genetic approach to build a robust intrusion detection system.

The fuzzy if then rules extract features even from such small and imprecise dataset to extract relevant features which will help in further classification. The genetic algorithm helps in obtaining optimized rules due to their adeptness in exploiting historical information and yielding better results.

The genetic fuzzy rule based inference engine has the ability to work in high dimensional network and is able to handle huge amount of audit data. KDD-99 data set is a high dimensional dataset with 41 attributes containing over 4, 00, 000 connections. So the IDS have the capability to handle large amount of data with ease.

Genetic algorithms are used in combination with fuzzy if then rules to prevent overlapping of rules and therefore each variable is distinguishable. For each connection, compatibility factor is calculated at the stage of fitness evaluation and then is classified into respective class which has higher compatibility factor. Genetic algorithms are also used to maintain diversity.

Also, the mathematical model seeks to cover each sample of the training dataset by classifying them through rule set, thus strengthening precision.

6.3 Future Scope

Reducing the false detection rate and substantially increasing the recall is one of the major concerns of any successful intrusion detection system. The proposed approach has a very high detection rate but a little higher false alarm rate of about 1.27%, so the primary aim would be decrementing the false alarm rate so as to avoid misclassification of connections.

The number of rules to be considered and kept in the database has not yet been evaluated. Thus the scope of this work can be extended by working on making concise and compact rules. Therefore, it would help in maximum classification in minimum number of rules.

This approach can also be made more effective by using better local search algorithms to generate more compact and accurate rules as these algorithms would look more into local optima, thereby yielding multiple local optimal solutions which would maintain diversity and validity in rules.

References

[1] G. Luger, R. Heady, A. Maccabe, and M. Servilla, "The architecture of a network-level intrusion detection system." Department of Computer Science, College of Engineering, University of New Mexico, 1990.

[2] SANS Institute, "Intrusion Detection Systems: Definition, Need and Challenges",http://www.sans.org/reading_room/whitepapers/detection/intrusion-detection-systems-definition-challenges_343, SANS Institute, 2001.

[3] PA. Porras, and A. Valdes, "Live Traffic Analysis of TCP/IP Gateways." In Proceedings of Network and Distributed System Security Symposium, 1998.

[4] H. Ishibuchi, K. Nozaki, N. Yamamoto, and Hideo Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms." IEEE Transactions on Fuzzy Systems, vol. 3(3), pp. 260-270, 1995.

[5] H. Ishibuchi, T. Yamamoto, "Fuzzy Rule Selection by Data Mining Criteria and Genetic Algorithms", In Proc. of Genetic and Evolutionary Computation Conference, pp. 399-406, 2002.

[6] H. Ishibuchi, T. Yamamoto, "Comparison of heuristic criteria for fuzzy rule selection in classification problems." Fuzzy Optimization and Decision Making, vol. 3(2), pp. 119-139, 2004.

[7] H. Ishibuchi, T. Nakashima, and M. Nii, "Classification and modeling with linguistic information granules", Berlin, Springer, 2005.

[8] T.Nakashima, A.Ghosh, "Classification Confidence of Fuzzy Rule-Based Classifiers" In Proceedings of 25th European Conference on Modeling and Simulation, pp. 466-471, 2011.

[9] DE.Denning, "An intrusion-detection model", IEEE Transactions on Software Engineering, vol. 13(2), pp. 222-232, 1987.

[10] H.Debar, M.Dacier, and A.Wespi, "Towards a taxonomy of intrusion-detection systems", Computer Networks vol. 31(8), pp. 805-822, 1999.

[11] H.Debar, M.Dacier, and A.Wespi, "A revised taxonomy for intrusion-detection systems", In Annales des télécommunications, Springer-Verlag, vol. 55(7-8), pp. 361-378, 2000.

[12] J.McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by

Lincoln Laboratory", ACM transactions on Information and system Security, vol. 3(4), pp. 262-294, 2000.

[13] SJ.Stolfo, W.Fan, W.Lee, A.Prodromidis, and PK.Chan, "Cost-based modeling for fraud and intrusion detection: Results from the JAM project", In DARPA Information Survivability Conference and Exposition, DISCEX'00. Proceedings, IEEE, vol. 2, pp. 130-144, 2000.

[14] RP.Lippman, DJ.Fried, I.Graf, JW.Haines, KR. Kendall, D.McClung, D.Weber et al., "Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation", In DARPA Information Survivability Conference and Exposition, DISCEX'00. Proceedings, IEEE, vol. 2, pp. 12-26, 2000.

[15] F.Herrera, L.Magdalena, "Genetic fuzzy systems: A tutorial." Tatra Mt. Math. Publ. (Slovakia), vol. 13, pp. 93-121, 1997.

[16] KA.Dejong, WM.Spears, "Learning Concept Classification Rules Using Genetic Algorithms", George Mason Univ Fairfax Va, 1990.

[17] C.Karr, "Genetic algorithms for fuzzy controllers", AI Expert, vol.6 (2), pp. 26-33, 1991.

[18] O.Cordón, F. Herrera, F. Gomide, F.Hoffmann, L.Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends." Joint 9th IFSA World Congress and 20th NAFIPS International Conference, IEEE, vol. 3, pp. 1241-1246, 2001.

[19] O. Cordón, M. J. D. Jesus, F. Herrera, and M. Lozano, "MOGUL: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach", International Journal of Intelligent Systems, vol. 14(11), pp. 1123-1153, 1999.

[20] J.C Bezdek, "Computational Intelligence Defined-by Everyone." Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, Springer Berlin Heidelberg, pp. 10-37, 1998.

[21] MJ.Middlemiss, and G.Dick, "Weighted feature extraction using a genetic algorithm for intrusion detection", The 2003 Congress on Evolutionary Computation, CEC'03, vol. 3, pp. 1669-1675, 2003.

[22] Y. Liao, and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection", Computers & Security, vol. 21(5), pp. 439-448, 2002.

[23] C. Zhang, J. Jiang, and M. Kamel, “Intrusion detection using hierarchical neural networks”, Pattern Recognition Letters vol. 6(6), pp. 779-791, 2005.

[24] B. C. Rhodes, J. A. Mahaffey, J. D. Cannady, “Multiple self-organizing maps for intrusion detection”, Proceedings of the 23rd National Information Systems Security Conference, pp. 16-19, 2000.

[25] C. H. Lee, S. W. Shin, J. W. Chung, “Network intrusion detection through genetic feature selection”, Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD, IEEE, pp. 109-114, 2006.

[26] X. Wang, J. Yang, X. Teng, W. Xia, R. Jensen, “Feature selection based on rough sets and particle swarm optimization”, Pattern Recognition Letters, vol. 28(4) pp. 459-471, 2007.

[27] G. Stein, B. Chen, A.S. Wu, K. A. Hua. “Decision tree classifier for network intrusion detection with GA-based feature selection.” Proceedings of the 43rd Annual Southeast Regional Conference, ACM, Vol.2, pp. 136-141, 2005.

[28] A. H. Sung, S. Mukkamala, “Feature ranking and selection for intrusion detection systems using support vector machines”, Proceedings of the Second Digital Forensic Research Workshop, 2002.

[29] S. Mukkamala, G. Janoski, and A. Sung, “Intrusion detection using neural networks and support vector machines”, Proceedings of the 2002 International Joint Conference in Neural Networks, IJCNN'02, IEEE, vol. 2, pp. 1702-1707, 2002.

[30] A. Hofmann, T. Horeis, and B. Sick, “Feature selection for intrusion detection: an evolutionary wrapper approach”, Proceedings of the 2004 International Joint Conference in Neural Networks, IEEE, vol. 2, pp. 1563-1568, 2004.

[31] W. Lu, and I. Traore, “A new evolutionary algorithm for determining the optimal number of clusters”, Computational Intelligence for Modeling, Control and Automation, 2005 and International Conference on Intelligent Agents, International Conference on Web Technologies and Internet Commerce, IEEE, vol. 1, pp. 648-653, 2005.

[32] J. Gomez, D. Dasgupta, “Evolving fuzzy classifiers for intrusion detection”, Proceedings of the 2002 IEEE Workshop on Information Assurance, New York: IEEE Computer Press, vol. 6(3), pp. 321-323, 2002.

[33] M. S. Abadeh, H. Mohamadi, J. Habibi, “Design and analysis of genetic fuzzy systems for intrusion detection in computer networks”, Expert Systems with Applications, vol. 38(6), pp. 7067-7075, 2011.

[34] M. S. Abadeh, J. Habibi, S. Aliari. “Using a particle swarm optimization approach for evolutionary fuzzy rule learning: a case study of intrusion detection.” Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU), pp. 2-7, 2006.

[35] F. J. Berlanga, A. J. Rivera, M. J. D. Jesús, F. Herrera, “GP-COACH: Genetic Programming-based learning of Compact and Accurate fuzzy rule-based classification systems for High-dimensional problems”, Information Sciences vol. 180(8), pp. 1183-1200, 2010.

[36] E. K. Aydogan, I. Karaoglan, P. M. Pardalos, “HGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems”, Applied Soft Computing vol. 12(2), pp. 800-806, 2012.

[37] S. B. Cho, “Incorporating soft computing techniques into a probabilistic intrusion detection system”, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, vol. 32(2), pp. 154-160, 2002.

[38] M. Mischiatti, and F. Neri, “Applying local search and genetic evolution in concept learning systems to detect intrusion in computer networks”, Proceedings of Workshop about Machine Learning and Data Mining. Seventh conference AI* IA" Intelligenza Artificiale, 2000.

[39] A. Giordana, and F. Neri, “Search-intensive concept induction”, Evolutionary computation, vol. 3(4), pp. 375-416, 1995.

[40] D. Song, M. I. Heywood, and A. N Zincir-Heywood, “A linear genetic programming approach to intrusion detection”, In Genetic and Evolutionary Computation-GECCO, Springer Berlin Heidelberg, pp. 2325-2336, 2003.

[41] A. Abraham, and C. Grosan, “Evolving intrusion detection systems”, Genetic Systems Programming, Springer Berlin Heidelberg, pp. 57-79, 2006.

[42] S.Forrest, S.A. Hofmeyr, A. Somayaji, “Computer immunology”, Communications of the ACM vol. 40(10), pp.88-96, 1997.

[43] J. L. Deneubourg, S. Goss, N. Franks, A. S. Franks, C. Detrain, L. Chrétien, "The dynamics of collective sorting robot-like ants and ant-like robots" Proceedings of the first international conference on simulation of adaptive behavior on from animals to animals, pp. 356-363, 1991.

[44] V. Ramos, A. Abraham, "ANTIDS: Self Organized Ant-Based Clustering Model for Intrusion Detection System", Soft Computing as Transdisciplinary Science and Technology, Springer Berlin Heidelberg, pp. 977-986, 2005.

[45] A. N. Toosi, M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers", Computer communications, vol. 30(10), pp. 2201-2212, 2007.

[46] J. Xin, J. E. Dickerson, J. A. Dickerson, "Fuzzy feature extraction and visualization for intrusion detection", The 12th IEEE International Conference Fuzzy Systems, FUZZ'03, vol. 2, pp. 1249-1254, 2003.

[47] C. H. Tsang, S. Kwong, H. Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection", Pattern Recognition, vol. 40(9), pp. 2373-2391, 2007.

[48] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

List of Publications

1. Kriti Chadha and Dr. Sushma Jain, ”Hybrid Genetic Fuzzy Rule Based Inference Engine to Detect Intrusion in Networks”, Third International Symposium on Intelligent Informatics (ISI'14), Springer, 2014.[Accepted]
2. Kriti Chadha and Dr. Sushma Jain, “Impact of Black Hole and Gray Hole attack in AODV Protocol”, International Conference on Recent Advances and Innovations in Engineering” ,IEEE,2014. [Accepted]
3. Kriti Chadha and Dr. Sushma Jain,” Network Intrusion Detection System using Intelligent Algorithms”, Applied Soft Computing, Elsevier.[To be communicated]