Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm

Swati Paliwal
Assistant Professor
Dept. of C.S.E.
Sharda University,
Gr.Noida,
India

Ravindra Gupta
Assistant Professor
Dept. of C.S.E.
SSSIST, Sehore,
India

ABSTRACT
Nowadays it is very important to maintain a high level security to ensure safe and trusted communication of information between various organizations. But secured data communication over internet and any other network is always under threat of intrusions and misuses. To control these threats, recognition of attacks is critical matter. Probing, Denial of Service (DoS), Remote to User (R2L) Attacks are some of the attacks which affects large number of computers in the world daily. Detection of these attacks and prevention of computers from it is a major research topic for researchers throughout the world. In this paper idea for use of a Genetic Algorithm (GA) based approach for generation of rules to detect Probing, DoS and R2L attacks on the system is proposed.

General Terms
Network Security, Genetic Algorithms.

Keywords
Probing, Denial of Service and Remote to User attacks, Genetic Algorithm, Intrusion Detection System Rule Set, KDD Cup 99 Dataset.

1. INTRODUCTION
In 1987 Dorothy E. Denning proposed intrusion detection as is an approach to counter the computer and networking attacks and misuses [1]. Intrusion detection is implemented by an intrusion detection system and today there are many commercial intrusion detection systems available. Generally an intruder is defined as a system, program or person who tries to and may become successful to break into an information system or perform an action not legally allowed [2]. IDS are a tool that monitors events occurring in a computer system or network and analyzes them for signs of security threats [3]. There are two types of intruders: internal intruders and external intruders. The unauthorized users who enter the system and make changes to the system and access the resource in the network without authorization, is an external intruder. The internal intruder refers to those who have access permissions and wish to perform unauthorized activities. There are two groups of intrusion detection systems (IDS's): misuse detection and anomaly detection. The Misuse detection system performs the detection of intrusions through a matching with known patterns , and anomaly detection systems Identifying the abnormalities from the normal network behavior. Hybrid detection systems combine both the misuse and anomaly detection systems [4]. deviations from normal network behaviors and alert for potential unknown attacks. Some IDSs incorporate mutually misuse and anomaly detection and form hybrid detection systems. The IDSs can also be classified into two categories depending on where they look for intrusions. A host-based IDS monitors activities associated with a particular host, and a network based IDS listens to network traffic. Numbers of machine learning approaches are available for detecting network intrusions [5]. In this paper, a Genetic Algorithm based approach is presented for network misuse detection. The GA is implemented and evaluated on the KDD Cup 99 dataset.

2. PAPER ORGANIZATION
The paper is organized as follows. Section 3 focuses on the KDD Cup 99 Dataset features. Section 4 describes the related work. Section 5 focuses on the genetic algorithm and its application in intrusion detection and response systems. The proposed model based on the GA is explained in Section 6. Section 7 presents GA Based IDS.Section 8 focuses on training and testing data. And Section 9 concludes the paper.

3. KDDCUP99 DATASET
This data set [6] is helpful for to check and to work with the system classifier. MIT Lincoln labs provided this dataset, and worked with Air Force LAN. MIT Lincoln Labs having different types of intrusions present in military networking environment to take nine weeks of raw TCP/IP data combined with several attacks of different types. Each TCP/IP connection with features like duration, type of protocol, dst_hst_service_count etc., is named also normal with an exact form of attack International Journal of Wisdom Based Computing, Vol. 1 (3), December 2011 87 such as warezmaster, pod, perl, smurf etc. All TCP/IP connection was particularly explained by different 41 contiguous. The list of models for normal class and attack class done in 10% of the data set of DARPA with classification was present in Table 1 and amount of attacks in training KDDCUP data set in Table 2.

Table 1: Attack types from KDD Cup 99 Dataset

Normal	Probing	DOS	R2L	U2R
Normal (97277)	Nmap (231)	Land (21)	Spy(2)	Buffer_overflow(30)
	Portsweep (1040)	Pod (264)	Phf(4)	Rootkit(10)
	Ipsweep (1247)	Teardrop (979)	Multihop (7)	Loadmodule (9)
	Satan (1589)	Back (2203)	ftp_write (8)	Perl(3)
		Neptune (107201)	Imap(12)	
		Smurf (280790)	Warezmaster(20)	
			Guess_passwd (53)	

Table 2: Number of Attacks in Training KDDCUP99 Dataset

Data Set	Normal	Probing	Dos	R2L	U2R
10%KDD	97277	4107	391458	1126	52
Corrected KDD	60593	4106	229853	11347	70
Whole	972780	41102	3883370	1126	50

3.1Denial of Service Attacks (DoS)

A DoS attack is a type of attack in which the hacker makes a computing or memory resources too busy or too full to serve legitimate networking requests and hence denying users access to a machine e.g. apache, smurf, Neptune, ping of death, back, mail bomb, UDP storm etc. are all DoS attacks. B. Remote to User Attacks (R2L): A remote to user attack is an attack in which a user sends packets to a machine over the internet, which s/he does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer e.g. xlock, guest, xnsnoop, phf, sendmail dictionary etc.

3.2User to Root Attacks (U2R):

These attacks are exploitations in which the hacker starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges e.g. perl, xterm.

3.3Probing:

Probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system. This technique is commonly used in data mining e.g. saint, portsweep, mscan, nmap etc.[7]

4. RELATED WORKS

Various applications of soft computing techniques in Intrusion Detection are discussed briefly in this section. Xiao et al. [8] present an approach that uses information theory and GA to detect abnormal network behaviors. Based on the mutual information between network features and the types of network intrusions, a small number of network features are closely identified with network attacks. Then a linear structure rule is derived using the selected features and a GA. The use of mutual information reduces the complexity of GA, and the single resulting linear rule makes intrusion detection efficient in real-time environment. However, the approach considers only discrete features. Li [9] present to detect network anomalous using Genetic Algorithm. The detection rates may be increased due to quantitative features inclusion. However, no implementation results are available. Bridges [11] Implemented a method to detect both anomalies and network misuses by combing Genetic Algorithm's and Fuzzy data mining technologies. In this method select the most significant network features and locate the best possible parameters of the fuzzy function by using Genetic Algorithm. Crosbie [14] proposed a methodology to detect network anomalies using Genetic Programming (GP) and multiple agent technology. When the agents are not properly initialized, the training process takes long time. The communication among small autonomous agents is still a problem. Selvakani [12] Applied Genetic Algorithm to generate rules for training the IDS. Rules are generated for only Smurf (DoS) attack and Warzemaster (R2L) attack. This performance of this methodology detection rate is low. This survey shows that the proposed Intrusion Detection models for R2L, U2R, Probe attacks get low detection rates using KDDCup dataset. This paper studies two types of attacks for each category i.e., DoS, R2L, U2R and Probe. Observed all the features in the KDDCUP Dataset to detect the attacks. Lu [13] Develop a method to derive a set of classification rules by using Genetic Programming (GP) with help of past data of network. In this method using GP the practical implementation is more difficult due to the system required more data or time.

5. GENETIC ALGORITHM OVERVIEW

Genetic Algorithm (GA) is based on the Darwin's theory of evolution; the basic rule is" Survival of the fittest". GA handles a population of possible solutions. Each solution is represented through a chromosome. GA uses an evolution and natural selection that uses a Chromosome-like data structure and evolve the chromosomes using selection, recombination and mutation operators [6]. The process usually begins with randomly generated population of chromosomes, which represent all possible solution of a problem that are considered candidate solutions. From each chromosome different positions are encoded as bits, characters or numbers. These positions could be referred to as genes. An evaluation function is used to calculate the goodness of each chromosome according to the desired solution; this function is known as "Fitness Function". During the process of evaluation "Crossover" is used to simulate natural reproduction and "Mutation" is used to mutation of species [9]. For survival and combination the selection of chromosomes is biased towards the fittest chromosomes. When we use GA for solving various problems three factors will have vital impact on the effectiveness of the algorithm and also of the applications [10]. They are: i) the fitness function; ii) the representation of individuals; and iii) the GA parameters. The determination of these factors often depends on applications and/or implementation.

1. Initialize population
2. Calculate fitness of population.
3. Perform selection. Roulette wheel is technique that randomly selects chromosomes giving proportional weight to

chromosomes with higher fitness.
4. Perform crossover
5. Perform mutation
6. If stopping criteria not met, go back to step 2.
7. Quit

6. PROPOSED MODEL BASED GA

Figure 1 shows the structure of simple Genetic Algorithm. The process starts from an initial population of randomly generated individuals. Then the population is evolved for a number of generations while progressively improving the qualities of the individuals by increasing the fitness value as the measure of quality.

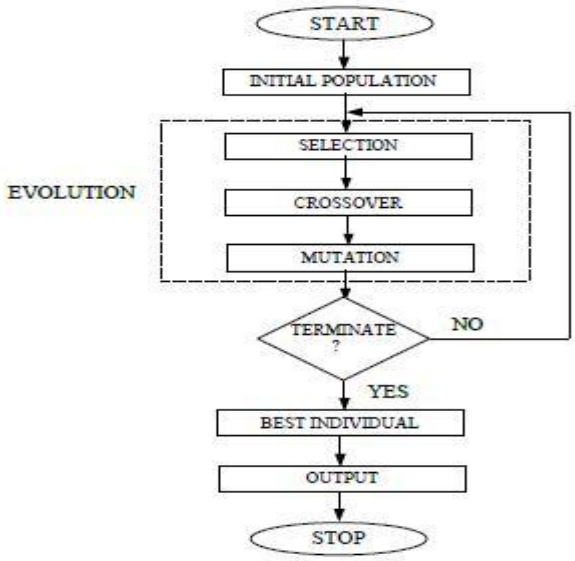


Figure 1: Flowchart of genetic algorithm

During each generation selection, cross over, and mutation are one after the other applied to each individual with certain probabilities. First, the numbers of best-fit individuals are selected based on a user-defined fitness function. The remaining individuals are selected and paired with each other. Each individual pair produces one offspring by partially exchanging their genes around one or more randomly selected crossing points. At the end, a certain number of individuals are selected and the mutation operations are applied. Selection is the phase where population individuals with better fitness are selected, otherwise it gets damaged.

6.1Crossover

A crossover operator is used to recombine two strings to get a better string. In crossover operation, recombination process creates different individuals in the successive generations by combining material from two individuals of the previous generation. The two strings participating in the crossover operation are known as parent strings and the resulting strings are known as children strings. the crossover operator recombines good sub-strings from good strings together,hopefully to create a better substring.

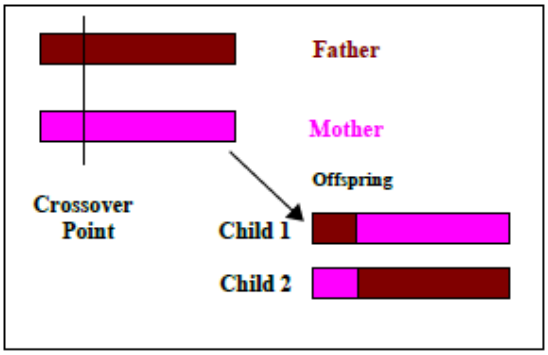
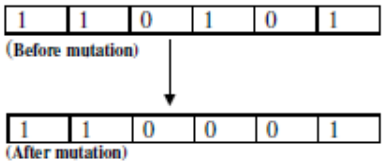


Figure 2: Crossover

Crossover shown in Figure 2 is a process where each pair of individuals selects randomly participates in exchanging their parents with each other, until a total new population has been generated.

6.3Mutation

Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operators. Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level; when the bits are being copied from the current string to the new string, there is probability that each bit may become mutated. This probability is usually a quite small value, called as mutation probability. The mutation operator alters a string locally expecting a better string.



Figurer 3: Mutation (mutation on 4th bit)

7. GA Based IDS (PROPOSED APPROACH)

In the future architecture (Shown in fig.4) contains two stages. Within the initial one, the learning stage, using network audit data to generated rule set for detecting intruders. The second stage, highest fitness value and best rule set is used for detect intruders in the internet globe.

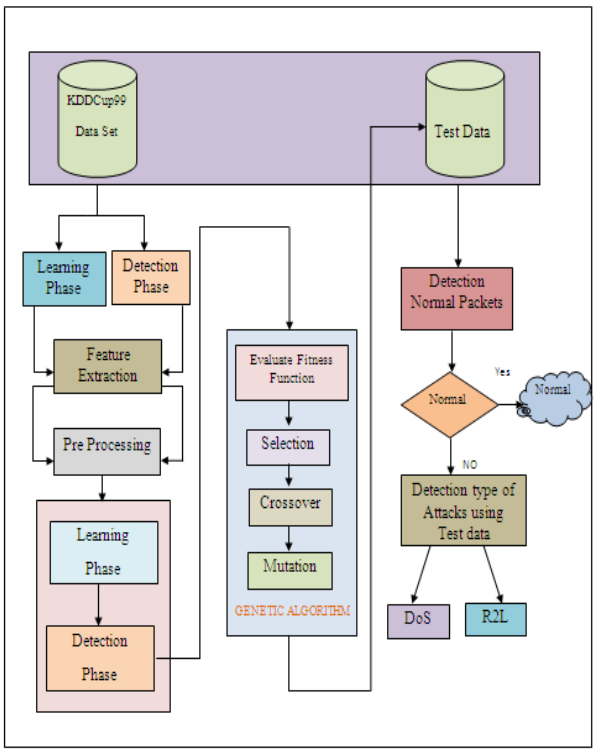


Figure 4: The simple design of the future model

In the above fig. Genetic Algorithm works on an individual called chromosome [15] and evolves the group of chromosomes to a population of quality individuals. Each chromosome represents a technique to solve the problem. A fitness function will be there for each rule which is a measure of each rules implementation. The evolution of population starts from an initial population of selected chromosomes which gradually improve the fitness value.

The three genetic operators selection, crossover, mutation are applied to each individual during the generation process. A group of suitable chromosomes are selected using a fitness function initially eliminating the other individuals. The process continuous by selecting a number of individuals and making pairs each other. The chromosome pair generates one off-string which exchanges their genes around selected cross points. Finally, some individuals are identified and the mutation operations are applied on it. The sub attack labels (smurf, mailbomb, saint, warezmaster etc.,) are recognized with respect to the fitness criteria by selecting the best-fit chromosomes capable of detecting the attacks from every population.

8. TRAINING AND TESTING DATA

We have used the KDDCUP 99 data set to train and test the system classifier. The dataset has been provided by MIT Lincoln Labs. It contains a wide variety of intrusions simulated in a military network environment set up to acquire nine weeks of raw TCP/IP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. The LAN was operated as if it were a true Air Force environment, peppered with multiple attacks. Hence, this is a high confidence and high quality data set. They set up an environment to collect TCP/IP dump raws from a host located on a simulated military network. Each TCP/IP connection is described by 41 discrete and continuous features (e.g. duration, protocol type, flag, etc.) and labeled as either normal, or as an attack, with exactly one specific attack type (e.g. Smurf, Perl, etc.). Attacks fall into four main categories: (i) Denial of Service Attacks (DOS) in which an attacker overwhelms the victim host with a huge number of requests. (ii) User to Root Attacks (U2R) in which an attacker or a hacker tries to get the access rights from a normal host in order, for instance, to gain the root access to the system. (iii) Remote to User Attacks (R2L) in which the intruder tries to exploit the system vulnerabilities in order to control the remote machine through the network as a local user. (iv) Probing in which an attacker attempts to gather useful information about machines and services available on the network in order to look for exploits. For our system, we have used 10% of the training set containing as published by Lincoln Labs which contains 494,021 connections. Our testing set is the entire set of labeled connections consisting of around 4.9 million connections. Thus, using the entire data set we have been able to test our system on unseen connections.Till the current stage of implementation, we have been able to generate a rule set comprising of six rules, each to correctly classify six different attack labels. We selected the top three distributions of attack labels from the two classes of attacks: DOS and Probe, in the 10% training data set. These six attack labels are: smurf, Neptune, land: type of DOS attacks and satan, ipsweep and portsweep: type of Probe attacks.

8.1Data Representation

Every network connection in KDD dataset has 41 features. Out of these 41 only those which are having high possibilities to be involved in intrusions has to be selected. Some features have symbolic form (e.g. protocol). These features will be converted into numerical ones by assigning a unique number for each feature. The resulting map will be used to do the same for the testing data set. Chromosomes will be encoded using binary encoding. Table 3 Shows the features and their formats. The feature names are given in the second column, while the third and fourth columns indicate how each of the network features is encoded in a chromosome. The third column represents the feature format and the fourth column shows the number of genes used for the corresponding feature. Note that different genes can be represented in different data types such as byte, integer, and float. This is necessary because of different formats and data ranges for different features. For example, the feature “Duration” has three components (hours, minutes, and seconds), each of which is represented by one gene of type byte. Similarly, each of the features “Protocol”,” Source port”, “Destination port” and “Attack name” is encoded using one gene of type integer, and each of the features “Source IP” and “Destination IP” has four components (a, b, c, and d), each of which is represented by one gene of type byte[16].

Table 3: Selected network features

S/no	Feature Name	Formate	No. of Genes
1	Duration	H:M:S	**3**
2	Protocol	Numeric(Int)	**1**
3	Source Port	Numeric(Int)	**1**
4	Destination Port	Numeric(Int)	**1**
5	Source IP	a.b.c.d	**4**
6	Destination IP	a.b.c.d	**4**
7	Service	Symbolic(http,ftp,smtp)	**1**
8	Attack Name	String(Int)	**1**

8.2 Rules For Ids

By analyzing the dataset, rules will be generated in the rule set. These rules will be in the form of an „if then‟ format as follows [17].

if {condition} then {act}

Which contains a "condition" and an "outcome"? The condition will result in a "true‟ or 'false‟. The attack name will be specified only if the condition is true. The first seven features in Table 3 are connected using the logical AND operations and compose the "condition" part of a rule. The feature "Attack name" is used in the "act". part, which indicates the classification of a network record (at training stage) or connection (at intrusion detection stage) when the "condition" part of a rule is matched. The following shows a rule example that classifies a network connection as the denial-of-service attack Neptune.

Examples

a) The denial-of-service attack Neptune

if (duration="0:0:1" and protocol="finger" and source_port=18982 and destination_ port=79 and source_ip="9.9.9.9" and destination_ip="172.16.112.50" and service="Private")then (attack_name="neptune")

The above rule expresses that if a network packet is originated from IP address 9.9.9.9 and port 18982, and sent to IP address 172.16.112.50 and port 79 using the protocol finger, and the connection duration is 1 second, then most likely it is a network attack of type neptune that may eventually cause the destination host out of service. Each rule is encoded as a chromosome using a fixed length vector, where each network feature is encoded using one or more genes of different types (see the third and fourth column of Table 3). In the above example, the encoded form of the rule would look like as follows:

{0, 0, 1, 2, 18982, 79, 9, 9, 9, 9, 9, 172, 16, 112, 50, 1,1}

To make the rules more general, wildcards are allowed in several network features. In case of a wildcard, the corresponding gene is encoded as -1. For example, if the above rule was generalized to be applicable to all packets originated from network 9.9.*.*, then the rule would be encoded as:

{0, 0, 1, 2, 18982, 79, 9, 9, -1, -1, 172, 16, 112, 50, 1,1}

b) The Denial-of-Service attack "Smurf"

if (duration="0:0:1" and protocol="finger" and source_port=18982 and destination_port=79 and source_ip="9.9.9.9" and destination_ip="172.16.112.50" and service="http") then (attack_name="smurf")

The above rule expresses that if a network packet is originated from IP address 9.9.9.9 and port 18982, and sent to IP address 172.16.112.50 and port 79 using the protocol finger, and the connection duration is 1second, then most likely it is a network attack of type smurf that may eventually cause the destination host out of service.

C) The Probing Attack "Port Scan"

if (duration="0:0:1" and protocol="TELNET" and source_port=18982 and destination_port=79 and source_ip="9.9.9.9" and destination_ip="172.16.112.50" and service="ftp") then (attack_name="port scan")

In the above example, the encoded form of the rule would look like as follows:
{0, 0, 1, 2, 18982, 79, 9, 9, 9, 9, 9, 172, 16, 112, 50,1, 1}

To make the rules more general, wildcards are allowed in several network features. In case of a wildcard, the corresponding gene is encoded as -1. For example, if the above rule was generalized to be applicable to all packets originated from network 9.9.*.*, then the rule would be encoded as:

{0, 0, 1, 2, 18982, 79, 9, 9, -1, -1, 172, 16, 112, 50,1, 1}

d) The Remote to User Attack "Guess password"

if (duration="0:0:22" and protocol="TELNET" and source_port=1867 and destination_port=23 and source_ip="192.168.1.30" and destination_ip="192.168.0.20" and service="ftp") then (attack_name="guesspassword")

The above examples shows the sample rules to classify DoS attack(Naptune,Smurf),Probing(Portscan),(R2L) attack.

8.3Fitness Function

In Genetic algorithm, fitness function plays a major role similar to that played by activation function in neural network. The fitness function is used to evaluate the fitness of each set of membership functions. Fitness Function is the numerical value which is proportional to the ability or utility of individual represented by that chromosome.

A fitness value is assigned to each individual in the population. Individuals are ranked and selected according to their fitness in such a way that more fit individuals are more likely to enter the relevancy group. The fitness is evaluated by determining how many attack connections the rule matches. To determine the fitness of a rule, the support-confidence framework [18] will be used. If a rule is represented as if A then B, then the fitness of the rule will be determined using following equations:

Support = |A and B| / N

Confidence = |A and B| / |A|

Fitness = X1 * support + X2 * confidence

Here, N is the total number of network connections in the audit data, |A| stands for the number of network connections matching the condition A, and |A and B| is the number of network connections that matches the rule if A then B. The weights X1 and X2 are used as a threshold.

8.3.1 Sample Fitness Calculation

N = 10 connections.

|A| = 2

|A and B| = 1

w1 = 0.2

w2 = 0.8

fitness = w1 * support + w2 * confidence

support = | A and B | / N = 1 / 10 = 0.1

confidence = | A and B | / A = 1 /2 = 0.5

fitness = 0.2 * 0.1 + 0.5 * 0.8 = 0.42

9. CONCLUSION

Artificial Intelligence methods are gaining the most attention at present regarding its ability to learn and evolve, which makes them more precise and efficient in facing the huge number of unpredictable attacks. Hence in this paper methodology based on Genetic Algorithm for detection of Probing, Denial of Service and Remote to user attacks is proposed. The proposed approach aims at gaining maximum detections of the Probing, R2L and DoS attacks with minimum false positive rate. Out of the total intrusions in testing dataset, detection of more than 97% of the intrusions is expected by this approach. This approach will be very useful for the attack detection in today's changing attack methodologies. If the rules are updated dynamically with the firewall's log data, then this method will be very effective against new attacks.

10. REFERENCES

[1] M. Botha, R. Solms, "Utilizing Neural Networks For Effective Intrusion Detection", ISSA,2004.

[2] R. Graham, "FAQ: Network Intrusion Detection Systems". March 21, 2000.

[3] Chittur A. "Model Generation for an Intrusion Detection System Using Genetic Algorithms, publications/gaids-thesis01.pdf, accessed in 2006.

[4] B. Uppalaiah, 2K. Anand, 3B. Narsimha, 4S. Swaraj, 5T. Bharat,"Genetic Algorithm Approach to Intrusion Detection System", IJCST Vol. 3, Issue 1, Jan. - March 2012.

[5] Mr. Anurag Andhare,Prof. Arvind Bhagat Patil," Denial-of-Service Attack Detection Using Genetic-Based Algorithm", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2, Mar-Apr 2012, pp.094-098.

[6] KDD cup 1999 data, http:// kdd.ics.uci.edu/ databases/kddcup99/kddcup99.html

[7] Mohammad Sazzadul Hoque1, Md. Abdul Mukit2 and Md. Abu Naser Bikas3," An Implementation of Intrusion Detection System Using Genetic Algorithm", International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012.

[8] T. Xiao, G. Qu, S. Hariri, and M. Yousif, "AnbEfficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm", Proceedings of the 24th IEEE International Performance Computing and Communications Conference (IPCCC '05), Phoenix, AZ, USA. 2005. Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Net.

[9] W. Li, "Using Genetic Algorithm for Network Intrusion Detection". "A Genetic Algorithm Approach to Network Intrusion Detection". SANS Institute, USA, 2004.

[10] R. H. Gong, M. Zulkernine, P. Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection", 2005.

[11] Bridges, Susan, Rayford B. Vaughn,"Intrusion Detection via Fuzzy Data Mining", In Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp. 109-122, Ottawa, Canada, 2000.

[12] Selvakani S, R.S. Rajesh," Genetic Algorithm for framing rules for Intrusion Detection", IJCSNS, Vol.7, No.11, 2007.

[13] W. Lu, I. Traore,"Detecting new forms of network intrusion using genetic programming", Computational Intelligence Vol.20, Issue 3, 2004, pp. 475-494.

[14] Crosbie, Mark, Gene Spafford,"Applying Genetic Programming to Intrusion Detection", In Proceeding of 1995 AAAI Fall Symposium on Genetic Programming, pp. 1-8, Cambridge, Massachusetts, 1995.

[15] Polhlheim H,"Genetic and Evolutionary Algorithms: Principles, Methods and Algorithms", (Online) Available: <http://www.geatbx.com/docu/index.html>, accessed in 2006.

[16] Ren Hui Gong, Mohammad Zulkernine, Purang Abolmaesumi," A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection".

[17] Gong RH, Zulkernine M, Abolmaesumi P. "A Software Implementation of a Genetic Algorithm based approach to Network Intrusion Detection". In: Proceedings of the sixth international conference on software engineering, artificial intelligence, networking and parallel/distributed computing and first ACIS international workshop on self-assembling wireless networks (SNPD/SAWN,,05), 2005.

[18] Middlemiss M and Dick G, "Feature Selection of Intrusion detection data using a hybrid genetic of hybrid Intelligent systems, IOS Press Amsterdam, PP.519-527, 2003