ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

Network Intrusion Detection System Using Genetic Algorithm and Fuzzy Logic

Mostaque Md. Morshedur Hassan

Assistant Professor, Department of Computer Science and IT, Lalit Chandra Bharali College, Guwahati, India

ABSTRACT: These days Intrusion Detection System (IDS) which is defined as a solution of system security is employed to identify the abnormal activities in a computer system or network. So far different approaches have been utilized in intrusion detections, but unluckily any of the systems is not entirely ideal. Hence, the hunt of improved method goes on. In this progression, here I have designed an Intrusion Detection System (IDS), by applying genetic algorithm (GA) and fuzzy logic to efficiently detect various types of the intrusive activities within a network. The proposed fuzzy logic-based system could be able to detect the intrusive activities of the computer networks as the rule base holds a better set of rules. The experiments and evaluations of the proposed intrusion detection system are performed with the KDD Cup 99 intrusion detection benchmark dataset. The experimental results clearly show that the proposed system achieved higher accuracy rate in identifying whether the records are normal or abnormal ones and obtained reasonable detection rate.

Keywords: Intrusion Detection System (IDS), Anomaly Based Intrusion Detection, Genetic Algorithm, Fuzzy Logic, KDD Cup 99 Dataset.

I. INTRODUCTION

Intrusion detection is designed to monitor the malicious activities ([1], [2], [3],[4]) occurring in a computer system or network inside or outside and analysing them for signs of possible incidents, which are violations or forthcoming threats of violation of computer security policies, acceptable utilized policies, or standard security practices. Intrusion incidents to computer systems are increasing because of the commercialization of the internet and local networks [5] and new automated hacking tools. Computer systems are turning out to be more and more susceptible to attack, due to its extended network connectivity.

Nowadays, networked computer systems play an increasingly important role in our society and its economy. They have become the targets of a wide array of malicious attacks that invariably turn into actual intrusions. This is the reason computer security has become an essential concern for network administrators. Too often, intrusions cause havoc inside LANs and the time and cost to repair the damage can grow to extreme proportions. Instead of using passive measures to fix and patch security holes once they have been exploited, it is more effective to adopt a proactive approach to intrusions.

Intrusion Detection Systems (IDS) are primarily focused on identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators [6] in real-time, or near real-time, and those that process audit data with some delay (non-real-time). The latter approach would in turn delay the time of detection. In addition, organizations use IDSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDSs have become a necessary addition to the security infrastructure of nearly every organization.

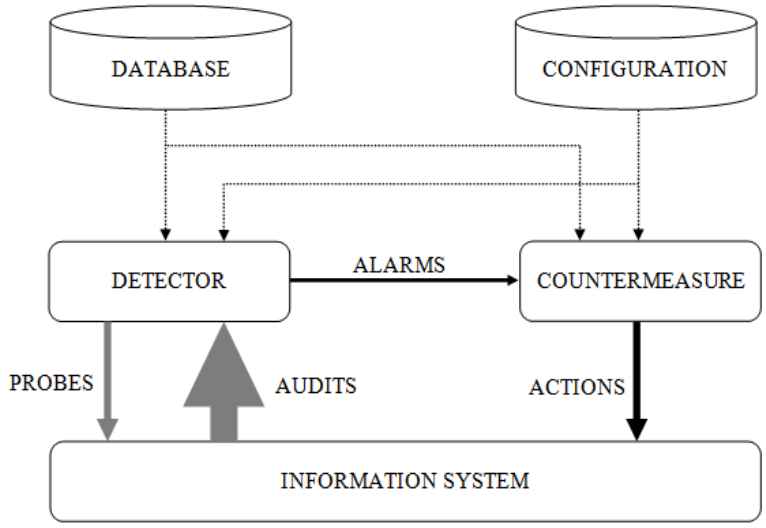
IDSs typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many IDSs can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDS stopping the attack itself, changing the security environment (e.g., reconfiguring a firewall), or changing the attack's content. A typical Intrusion Detection System is shown in figure 1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013



Note: The arrow thickness represents the amount of information flowing from one component to another
Figure 1: Very simple intrusion detection system

One of the major problems faced by IDS is huge number of false positive alerts, i.e. alerts that are mistakenly classified normal traffic as security violations. A perfect IDS does not generate false or irrelevant alarms. In practice, signature based IDS found to produce more false alarms than expected. This is because of the very general signatures and lack of built in verification tool to validate the success of the attack. The huge amount of false positives in the alert log makes the process of taking remedial action for the true positives, i.e. successful attacks, delayed and labour intensive.

My goal is to detect novel attacks by unauthorized users in network traffic. I consider an attack to be novel if the vulnerability is unknown to the target's owner or administrator, even if the attack is generally known and patches and detection tests are available. I am primarily interested in four types of remotely launched attacks: probes, denial of service (DOS), U2R and R2L. 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. A remote to user (U2R) attack is an attack in which a user sends packets to a machine over the internet, which he/she 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. A R2L 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. A probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system. This technique is commonly used in data mining e.g. saint, portsweep, mscan, nmap etc.

II. RELATED WORKS

The normal and abnormal behaviours [1] in networked computers are hard to predict, as the boundaries cannot be well defined. This prediction process usually generates fake alarms in many anomaly based intrusion detection systems.

With the introduction of fuzzy logic, the fake alarm rate in determining intrusive behaviour can be reduced, where a set of fuzzy rules is used to define the normal and abnormal behaviour [1] in a computer network. This system proposed a technique to generate fuzzy rules that are able to detect malicious activities and some specific intrusions.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

This system presented an approach for the performance of generated fuzzy rules in classifying different types of intrusions.

In this system, I explained the attack modes and point to the impact of this attack and its threats. From an attacker's perspective, I analyse each of the attack's modes, benefits and suitable conditions and think how to improve the attack by introducing the concept of fuzzy logic-based technique.

Fuzzy set theory was introduced by Zadeh [10] in 1965 and it was specifically designed mathematically represent uncertainty and vagueness with formalized logical tools for dealing with the imprecision inherent in many real world problems.

Hassan [4], Baruah ([7], [8]), Neog and Sut [9] have forwarded an extended definition of fuzzy set which enables us to define the complement of a fuzzy set. Our proposed system agrees with them as this new definition satisfies all the properties regarding the complement of a fuzzy set.

Gong [2] presented an implementation of GA based approach to Network Intrusion Detection using GA and showed software implementation. The approach derived a set of classification rules and utilizes a support-confidence framework to judge fitness function.

Xia, Hariri and Yousif [3] used GA to detect anomalous network behaviours based on information theory ([17], [18]). Some network features can be identified with network attacks based on mutual information between network features and type of intrusions and then using these features a linear structure rule and also a GA is derived. The approach of using mutual information and resulting linear rule seems very effective because of the reduced complexity and higher detection rate. The only problem is that it considered only the discrete features.

Abdullah [6] showed a GA based performance evaluation algorithm to network intrusion detection. The approach uses information theory for filtering the traffic data.

Lu and Traore [12] used historical network dataset using GP to derive a set of classification [17]. They used support-confidence framework as the fitness function and accurately classified several network intrusions. But their use of genetic programming made the implementation procedure very difficult and also for training procedure more data and time is required.

Goyal and Kumar [13] described a GA based algorithm to classify all types of smurf attack using the training dataset with false positive rate is very low (at 0.2%) and detection rate is almost 100%.

Li [14] described a method using GA to detect anomalous network intrusion ([17], [18]). The approach includes both quantitative and categorical features of network data for deriving classification rules. However, the inclusion of quantitative feature can increase detection rate but no experimental results are available.

III. INTRUSION DETECTION OVERVIEW

The following sections give a short overview of networking attacks, classifications and various components of Intrusion Detection System.

A. NETWORKING ATTACKS

This section is an overview of the four major categories of networking attacks. Every attack on a network can comfortably be placed into one of these groups [15] –

- **Denial of Service (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.
- **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 he/she 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

- **User 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.
- **Probing:** 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.

B. CLASSIFICATION OF INTRUSION DETECTION

Intrusions Detection can be classified into two main categories. They are as follow:

- **Host Based Intrusion Detection:** HIDSs evaluate information found on a single or multiple host systems, including contents of operating systems, system and application files ([11], [16]).
- **Network Based Intrusion Detection:** NIDSs evaluate information captured from network communications, analysing the stream of packets which travel across the network ([11], [16]).

C. COMPONENTS OF INTRUSION DETECTION SYSTEM

An intrusion detection system normally consists of three functional components [17]. The first component of an intrusion detection system, also known as the event generator, is a ***data source***. Data sources can be categorized into four categories namely Host-based monitors, Network-based monitors, Application-based monitors and Target-based monitors.

The second component of an intrusion detection system is known as the ***analysis engine***. This component takes information from the data source and examines the data for symptoms of attacks or other policy violations. The analysis engine can use one or both of the following analysis approaches:

- **Misuse/Signature-Based Detection:** This type of detection engine detects intrusions that follow Ill-known patterns of attacks (or signatures) that exploit known software vulnerabilities ([18], [19]). The main limitation of this approach is that it only looks for the known weaknesses and may not care about detecting unknown future intrusions [20].
- **Anomaly/Statistical Detection:** An anomaly based detection engine will search for something rare or unusual [20]. They analyses system event streams, using statistical techniques to find patterns of activity that appear to be abnormal. The primary disadvantages of this system are that they are highly expensive and they can recognize an intrusive behaviour as normal behaviour because of insufficient data
- The third component of an intrusion detection system is the ***response manager***. In basic terms, the response manager will only act when inaccuracies (possible intrusion attacks) are found on the system, by informing someone or something in the form of a response.

IV. GENETIC BASED IDS

A. GENETIC ALGORITHM OVERVIEW

A Genetic Algorithm (GA) is a programming technique that uses biological evolution as a problem solving strategy [21]. It is based on Darwinian's principle of evolution and survival of fittest to optimize a population of candidate solutions towards a predefined fitness [14].

The proposed GA based intrusion detection system contains two modules where each works in a different stage. In the training stage, a set of classification rules are generated from network audit data using the GA in an offline environment. In the intrusion detection stage, the generated rules are used to classify incoming network connections in the real-time environment. Once the rules are generated, the intrusion detection system becomes simple, experienced and efficient one.

GA uses an evolution and natural selection that uses a chromosome-like data structure and evolve the chromosomes using selection, recombination and mutation operators [14]. 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 decency of each chromosome according to the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

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 [14]. For survival and combination the selection of chromosomes is partial towards the fittest chromosomes.

When I use GA for solving various problems three factors will have vital impact on the effectiveness of the algorithm and also of the applications [2]. They are: i) the fitness function; ii) the representation of individuals; and iii) the GA parameters. The determination of these factors often depends on implementation of the system. In the following sections, I focus our discussions on deriving the set of rules using Genetic Algorithm.

B. FUZZY LOGIC

It has been shown by Baruah [7] that a fuzzy number [a, b, c] is defined with reference to a membership function $\mu(x)$ lying between 0 and 1, $a \leq x \leq c$. Further, he has extended this definition in the following way. Let $\mu1(x)$ and $\mu2(x)$ be two functions, $0 \leq \mu2(x) \leq \mu1(x) \leq 1$. He has concluded $\mu1(x)$ the fuzzy membership function, and $\mu2(x)$ a reference function, such that $(\mu1(x) - \mu2(x))$ is the fuzzy membership value for any x. Finally he has characterized such a fuzzy number by $\{x, \mu1(x), \mu2(x); x \in \Omega\}$.

The complement of μ_x is always counted from the ground level in Zadehian's theory [10], whereas it actually counted from the level if it is not as zero that is the surface value is not always zero. If other than zero, the problem arises and then we have to count the membership value from the surface for the complement of μ_x. Thus I could conclude the following statement –

Complement of $\mu_x = 1$ for the entire level

Membership value for the complement of $\mu_x = 1 - \mu_x$

My system forwarded a definition of complement of an extended fuzzy set where the fuzzy reference function is not always zero. The definition of complement of a fuzzy set proposed by Hassan [4], Baruah ([7], [8]), Neog and Sut [9] could be seen a particular case of what I am giving. I shall use Baruah's definition of the complement of a normal fuzzy set in my article.

In the two classes' classification problem, there are two classes where every object should be classified. These classes are called positive (abnormal) and negative (normal). The data set used by the learning algorithms consists of a set of objects, each object with n+1 attributes. The first n attributes define the object characteristics (monitored parameters) and the last attribute defines the class that the object belongs to the classification attribute.

A fuzzy classifier for solving the two class classification problem is a set of two rules, one for the normal class and other for the abnormal class, where the condition part is defined using only the monitored parameters and the conclusion part is an atomic expression for the classification attribute.

C. FLOWCHART

Figure 2 shows the operations of a general genetic algorithm according to which GA is implemented in our system.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

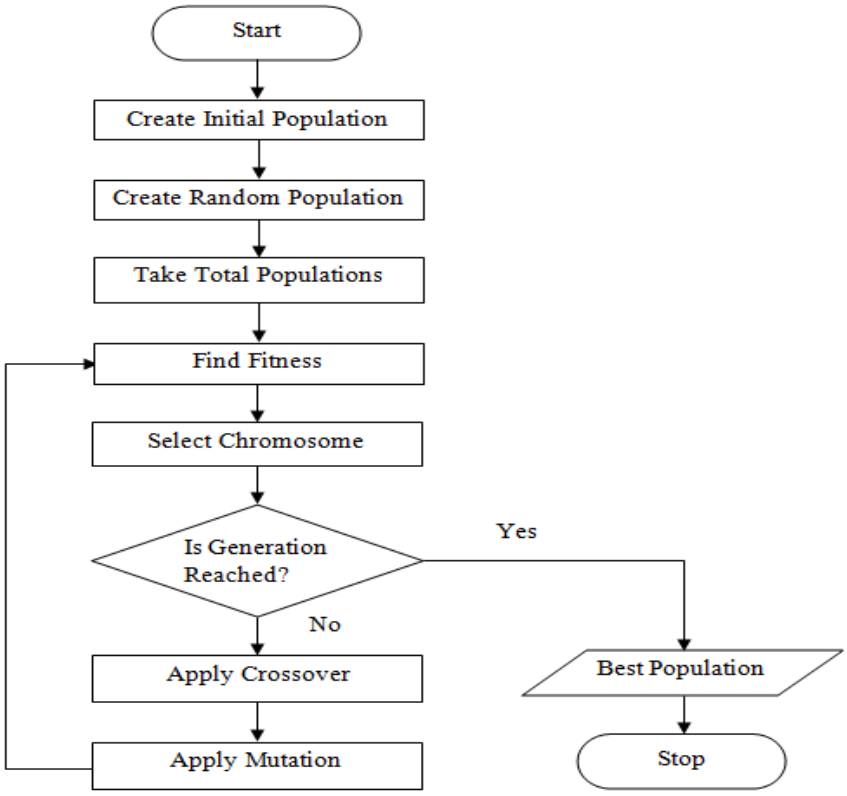


Figure 2: Flowchart of Genetic Algorithm

D. ALGORITHM OF THE PROPOSED SYSTEM

Algorithm – Rule set generation using GA
Input – Network audit data, number of generations, and population size
Output – A set of classification rules

1. initialize the population
2. generate random population
3. W1=0.2, W2=0.5, W3=0.3, T=0.5, chrom_length=9
4. N=total number of populations to be generated
5. for each chromosome in the population
6. TP=0, TN=0, FP=0, FN=0
7. for each record in the training set
8. if the record matches the chromosome
9. increment membership value of TP
10. end if
11. if the records do not match the chromosome
12. increment membership value of FP
13. end if
14. end for
15. Fitness=W1*TP/(TP+FN)+W2*FP/(FP+TN)+W3*(1-chrom_length/10)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

16. if Fitness>T
17. if N<1
18. break
19. else
20. select the chromosome into the new population
21. update the total number of population
22. N=N-1
23. end if
24. end if
25. end for
26. for each chromosome in the new population
27. apply crossover operator to the chromosome
28. apply mutation operator to the chromosome
29. end for
30. if the required number of generation is not reached, then go to step 5.

E. FITNESS FUNCTION

The authors in ([1], [4]) used the fuzzy confusion matrix to calculate the fitness of a chromosome. In the fuzzy confusion matrix the fuzzy truth degree of the condition represented by the chromosome and the fuzzy negation operator are used directly. In our case, the fitness of a chromosome for the abnormal class is evaluated according to the following set of equations:

$$TP = \sum_{i=1}^{p} \text{predicted}(\text{class\_data}_i)$$

$$TN = \sum_{i=1}^{q} 1 - \text{predicted}(\text{other\_class\_data}_i)$$

$$FP = \sum_{i=1}^{q} \text{predicted}(\text{other\_class\_data}_i)$$

$$FN = \sum_{i=1}^{p} 1 - \text{predicted}(\text{class\_data}_i)$$

Where,
Sensitivity = TP/(TP+FN)
Specificity = FP/(FP+TN)
Length = 1 – chromosome_length/10
So finally Fitness of a chromosome is calculated as follows –

Fitness = W1 * Sensitivity + W2 * Specificity + W3 * Length

Where,

TP, TN, FP, FN are true positive, true negative, false positive, false negative value for the rule, p is the number of samples of the evolved class in the training data set, q is the number of samples of the remaining class in the training data set, predicted is the fuzzy value of the conditional part of the rule, $class\_data_i$ is an element of the subset of the training samples of the evolved class, $other\_class\_data_i$ is an element of the subset of the remaining classes in the training samples, and W1, W2, W3 are the assigned weights for each rule characteristics respectively.

V. IDS IMPLEMENTATION

To implement the algorithm and to evaluate the performance of the system, I have used the standard dataset used in KDD Cup 1999 “Computer network intrusion detection” competition.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

A. KDD CUP SAMPLE DATASET

For the implementation of the proposed algorithm, I used the KDD 99 intrusion detection datasets which are based on the 1998 DARPA initiative, which provides designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies ([22], [25]). Hence, a simulation is being made of a factitious military network with three 'target' machines running various operating systems and services. They also used three additional machines to spoof different IP addresses for generate network traffic.

A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows from a source IP address to a target IP address under some well defined protocol ([22], [23], [25]). It results in 41 features for each connection.

Finally, there is a sniffer that records all network traffic using the TCP dump format [25]. The total simulated period is seven weeks. Normal connections are created to profile that expected in a military network and attacks fall into one of four categories: User to Root; Remote to Local; Denial of Service; and Probe.

The KDD 99 intrusion detection benchmark consists of different components [24]:

kddcup.data; kddcup.data_10_percent; kddcup.newtestdata_10_percent_unlabeled;

kddcup.testdata.unlabeled; kddcup.testdata.unlabeled_10_percent; corrected.

I have used "kddcup.data_10_percent" as training dataset and "corrected" as testing dataset. In this case the training set consists of 494,021 records among which 97,280 are normal connection records, while the test set contains 311,029 records among which 60,593 are normal connection records. Table 1 shows the distribution of each intrusion type in the training and the test set.

Dataset	Normal	Probe	Dos	U2r	R2l	Total
Train ("kddcup.data_10_percent")	97280	4107	391458	52	1124	494021
Test ("corrected")	60593	4166	229853	228	16189	311029

Table1: Distribution of intrusion types in datasets

B. IMPLEMENTATION PROCEDURE

In the calculation phase, I have made 23 groups of chromosomes according to training data. There were 23 (22+1) groups for each of attack and normal types presented in training data. Number of chromosomes in each group is variable and depends on the number of data and relationship among data in that group. Total number of chromosomes in all groups Ire tried to keep in reasonable level to optimize time consumption in testing phase.

In the testing/detection phase, for each test data, an initial population is made using the data and occurring mutation in different features. This population is compared with each chromosomes prepared in training phase. Portion of population, which are more loosely related with all training data than others, are removed. Crossover and mutation occurs in rest of the population which becomes the population of new generation. The process runs until the generation size comes down to 1 (one). The group of the chromosome which is closest relative of only surviving chromosome of test data is returned as the predicted type.

For the implementation, I have taken both continuous and discrete values from the extracted features of the datasets.

VI. EXPERIMENT RESULTS AND ANALYSIS

A. TRAINING AND TESTING DATA

The KDD 99 intrusion detection datasets [24] is broadly used to evaluate IDSs. In this study, two subsets were extracted from the 1998 DARPA data and used as the training and testing datasets. Each record of the datasets consists of 9 network features and 1 manually assigned record type. Nine network features have been used in the GA [14], which are *connection duration, protocol, flag, su_attempted, is_guest_login, same_srv_rate, dst_host_same_srv_rate, dst_host_srv_count,* and *count.*

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

The record type indicates whether a record is *a normal network connection or a particular network intrusion*. Most network packets in the selected datasets are normal, and four kinds of network attacks are present: *dos, probe, u2r,* and *r2l.*

B. EXPERIMENTS

In the experiment, the system was trained with the training dataset, and the default fitness function and the GA parameters were used, *i.e.*, *w1*=0.2, *w2*=0.5, w3=0.3, 10 genes of a chromosome, 2000 generations, 250 initial rules in the population, crossover rate of 0.5, two-point crossover, and mutation rate of 0.02. When the training process was finished, the top 15 best quality rules were taken as the final classification rules. The rules were then used to classify the training data and the testing data respectively.

The experimental results show that the proposed method resulted good detection rates when using the generated rules to classify the training data itself. The detection rates could be higher if the fitness function and the GA parameters were chosen more appropriately. The results are presented in Table 2.

		Predicted Label					% Correct
		Normal	**Probe**	**Dos**	**U2r**	**R2l**	
Actual Class	**Normal**	32871	1945	24835	597	345	54.25
	Probe	347	3132	536	2	149	75.18
	Dos	23987	8863	196984	14	5	85.70
	U2r	141	17	19	44	7	19.30
	R2l	10334	581	3450	125	1699	10.49
% Correct		48.57	21.54	87.23	5.63	77.05	

Table 2: Detected intrusion types in testing dataset

For simplified evaluation of our system, besides the classical accuracy measure, I have used two standard metrics of detection rate and false positive rate developed for network intrusions derived in ([4], [26]). Table 3 shows these standard metrics.

		Predicted Label	
		Normal	**Intrusion**
Actual Class	**Normal**	TN (32871)	FP (27722)
	Intrusion	FN (34809)	TP (215627)

Table 3: Standard metrics for system evaluation

Detection rate for each data type can be seen from figure 3.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

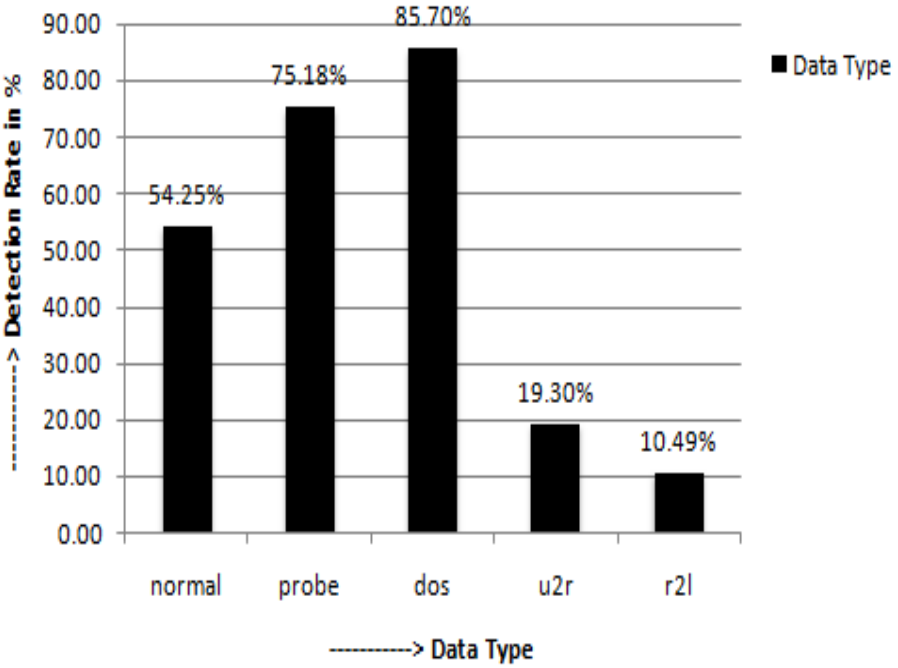


Figure 3: Detection rate for each class

Detection rate (DR) is calculated as the ratio between the number of correctly detected intrusions and the total number of intrusions ([4], [26]) that is:

$$DR=TP/(TP+FN)$$

Using table 3, detection rate, DR = 0.8610.

False positive rate (FPR) is calculated as the ratio between the numbers of normal connections that are incorrectly classifies as intrusions and the total number of normal connections ([4], [26]), that is:

$$FPR=FP/(FP+TN)$$

Using table 3, false positive rate, FPR = 0.4575.

VII. CONCLUSION AND FUTURE WORK

In this paper, a method of applying genetic algorithms with fuzzy logic is presented for network intrusion detection system to efficiently detect various types of network intrusions. To implement and measure the performance of the system I carried out a number of experiments using the standard KDD Cup 99 benchmark dataset and obtained reasonable detection rate. To measure the fitness of a chromosome I used the fuzzy confusion matrix where the fuzzy membership value and fuzzy membership function for the complement of a fuzzy set are two different concepts because the surface value is not always counted from the ground level. The proposed detection system can upload and update new rules to the systems as the new intrusions become known. Therefore, it is cost effective and adaptive. The method suffers from two aspects. Firstly, it generates false alarms which are very serious problem for IDS. Secondly, for high dimensional data, it is hard to generate rules that cover up all the attributes.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 7, September 2013

ACKNOWLEDGEMENT

I would like to extend our sincere thanks and gratefulness to H.K. Baruah, Professor, Department of Statistics, Gauhati University, Guwahati, India for his kind help and guidance in preparing this article.

REFERENCES

[1] J. Gomez and D. Dasgupta, "Evolving Fuzzy Classifiers for Intrusion Detection", Proceedings of the IEEE, 2005.
[2] R. H. Gong, M. Zulkernine, P. Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection", 2005.
[3] T. Xia, G. Qu, S. Hariri, M. Yousif, "An Efficient 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.
[4] M. M. M. Hassan, "Current Studies on Intrusion Detection System, Genetic Algorithm and Fuzzy Logic", International Journal of Distributed and Parallel Systems, Vol. 4, No. 2, pp. 35-47, 2013.
[5] Yao, J. T., S.L. Zhao, and L.V. Saxton, "A Study On Fuzzy Intrusion Detection", Proceedings of the Data Mining, Intrusion Detection, Information Assurance, And Data Networks Security", SPIE, Vol. 5812, Orlando, Florida, USA, pp. 23-30, 2005.
[6] B. Abdullah, I. Abd-alghafar, Gouda I. Salama, A. Abd-alhafez, "Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System", 2009.
[7] Hemanta K. Baruah, "Towards Forming A Field Of Fuzzy Sets", International Journal of Energy, Information and Communications, Vol. 2, Issue 1, pp. 16-20, 2011.
[8] Hemanta K. Baruah, "The Theory of Fuzzy Sets: Beliefs and Realities", International Journal of Energy, Information and Communications, Vol. 2, Issue 2, pp. 1-22, 2011.
[9] Tridiv Jyoti Neog, Dushmanta Kumar Sut, "Complement of an Extended Fuzzy Set", International Journal of Computer Applications, Vol. 29, No.3, pp. 39-45, 2011.
[10] Zadeh L A, "Fuzzy Sets", Information and Control, Vol.8, pp. 338-353, 1965.
[11] Y.Dhanalakshmi and Dr. I. Ramesh Babu, "Intrusion Detection Using Data Mining Along Fuzzy Logic and Genetic Algorithms", International Journal of Computer Science & Network Security (IJCSNS),Vol.8, No.2, pp. 27-32, 2008.
[12] W. Lu, I. Traore, "Detecting New Forms of Network Intrusion Using Genetic Programming", Computational Intelligence, vol. 20, pp. 3, BlackIll Publishing, Malden, pp. 475-494, 2004.
[13] Anup Goyal, Chetan Kumar, "GA-NIDS: A Genetic Algorithm based Network Intrusion Detection System", 2008.
[14] W. Li, "A Genetic Algorithm Approach to Network Intrusion Detection", SANS Institute, USA, 2004.
[15] A. Sung, S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks", in Symposium on Applications and the Internet, pp. 209– 216, 2003.
[16] J. P. Planquart, "Application of Neural Networks to Intrusion Detection", SANS Institute Reading Room.
[17] R. G. Bace, "Intrusion Detection", Macmillan Technical Publishing, 2000.
[18] S. Kumar, E. Spafford, "A Software architecture to Support Misuse Intrusion Detection", in the 18th National Information Security Conference, pp. 194-204, 1995.
[19] K. Ilgun, R. Kemmerer, P. A. Porras, "State Transition Analysis: A Rule-Based Intrusion Detection Approach", IEEE Transaction on Software Engineering, pp. 181-199, 1995.
[20] S. Kumar, "Classification and Detection of Computer Intrusions", Purdue University, 1995.
[21] V. Bobor, "Efficient Intrusion Detection System Architecture Based on Neural Networks and Genetic Algorithms", Department of Computer and Systems Sciences, Stockholm University/Royal Institute of Technology, KTH/DSV, 2006.
[22] KDD-CUP, Task Description, http://kdd.ics.uci.edu/databases/kddcup99/task.html, 1999.
[23] KDD Cup, Tasks, http://www.kdd.org/kddcup/index.php?section=1999&method=task, 1999.
[24] KDD Cup, Data, http://www.kdd.org/kddcup/index.php?section=1999&method=data, 1999.
[25] H. G. Kayacık, A. N. Zincir-Heywood, M. I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets", 2005.
[26] G. Folino, C. Pizzuti, G. Spezzano, "GP Ensemble for Distributed Intrusion Detection Systems", ICAPR, pp. 54-62, 2005.

BIOGRAPHY

MOSTAQUE MD. MORSHEDUR HASSAN was born in Mankachar, Dhuburi (District), Assam, India. He obtained his Master of Computer Application (MCA) degree from Allahabad Agricultural Institute Deemed University, Allahabad. Presently he is working as an Assistant Professor of Computer Science, in the department of Computer Science and Information Technology, Lalit Chandra Bharali College, Maligaon, Guwahati, Assam, India. And he is pursuing Ph.D. in Computer Science under Gauhati University, Guwahati, Assam. His research interests include Network Security, Intrusion Detection and Prevention, Wireless Security, Web Security, and Fuzzy Logic.