E-ISSN: 2321–9637

Volume 2, Issue 1, January 2014

International Journal of Research in Advent Technology

Available Online at: http://www.ijrat.org

FUZZY CLASSIFIER FOR IDS ALERTS USING GENETIC ALGORITHM

Mostaque Md. Morshedur Hassan[1], Hemanta K. Baruah[2]
[1] Assistant Professor of Department of Computer Science and IT
[2]Vice-Chancellor of Bodoland Univeristy, Kokrajhar, Assam (India)
[1] Lalit Chandra Bharali College, Maligaon, Guwahati (India)
[1]Email- mostaq786@gmail.com

ABSTARCT:

Intrusion Detection System (IDS) is designed to monitor a protected network for the authentication of malicious activities by analysing the network traffic and classifying the records as either normal or abnormal. After identifying a suspicious traffic, IDS generates and logs an alert. There are various approaches being utilized in intrusion detections, but unluckily any of the systems so far is not completely perfect. Most of the alerts generated using this prediction process is false positive. The abundance of false positive alerts makes it difficult for the security analyst to find successful attacks and take remedial action. This paper proposes an intrusion detection system that applies genetic algorithm and fuzzy logic to efficiently detect various types of intrusive activities within a network. The proposed method employs a two phase automatic alert classification system to support the human analyst in identifying the false positives. In the first phase, the alerts collected from one or more sensors are normalized and similar alerts are grouped to form a meta-alert. These meta-alerts are passively tested with an asset database to find out irrelevant alerts. Furthermore, an optional alert generalization is also performed for root cause analysis and thereby reduces false positives with human interaction. In the second phase, the reduced alerts are labelled and passed to an alert classifier which uses genetic based fuzzy logic techniques for building the classification rules. This benefits the analyst in automatic classification of the alerts. The proposed system is tested using KDD Cup'99 datasets and found to be effective in reducing the false positive alerts using the efficient fuzzy rules significantly, and thereby reducing the workload of human analyst.

Keywords: *Intrusion Detection System (IDS); Alert Classification; Alert Generalization; Alert Verification; Genetic Algorithm; Fuzzy Logic; KDD Cup 99 Dataset.*

1. INTRODUCTION

With the increased use of computers and ease of access to internet, the ways to attack and mislead a system has also increased. Though there are various ways to provide security such as anti-virus, malwares, spywares, cryptography, etc., it is not possible to provide complete secured systems. Therefore the need for Intrusion Detection System ([1] and [2]) occurred and has become the second line of defense. To identify intruders, differentiating normal user behavior and attack behavior is essential. Efficient IDS can be developed by defining a proper rule set for classifying the network traffic log records into normal or attack pattern. Moreover, frequent abnormal traffic on network requires more advanced technologies for monitoring and analyzing the network traffic.

The number of intrusions into computer systems is growing because new automated intrusion tools appearing every day, and these tools and different system vulnerability information are easily available on the web. These intrusions can come from inside (insider or legal users) or outside (outsider users) the system. An intrusion can be defined as any set of actions that attempt to compromise the reliability, privacy or accessibility of a resource. The problem of intrusion detection has been studied extensively in computer security ([1], [2], [7] and [8]), and has received a lot of attention in machine learning [8] and fuzzy logic ([1], [2] and [7]).

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 produce fake or irrelevant alarms. In reality, signature based IDS produces more false alarms than predictable. It occurs due to the general signatures used and lack of built in verification tool to validate the success of the attack. The huge quantity of false positive ratios in the alert log compels the method of taking corrective action to obtain successful attacks i.e. true positives.

Same intrusion event [8] can trigger hundreds of similar alerts. For example, a single network scan may cause to generate several alerts which differ by a little amount of interval. These generated alerts can be fused together before passing to human analyst. Likewise, various types of alerts will be having same underlying event as the root cause. We can simplify each attributes of all alerts to find out the associated alerts. This will benefit the method of root cause analysis and hence eliminate more number of false positives [7].

Usually, the IDSs gather and analyze information in a network to identify possible security breaches. If an intrusion is detected, the IDS provides a warning called an alert or alarm. Normal traffic and daily operations usually make IDSs generate many alerts ([1] and [2]), most of which are false alerts. The IDSs are known to generate huge volumes of alerts. Without proper alert management, the IDS performance may degenerate because of the difficulties in dealing with overpowering unnecessary numbers of alerts.

Practically, there is no IDS that can completely eliminate false alerts. Techniques such as fine tuning and disabling signatures help to reduce false alerts but they might degrade the security level thereby increasing the risk of missing the real intrusions. This calls for better mechanisms of dealing with huge and offensive number of alerts.

In this work, we focus on extended definition fuzzy set to define the complement of a fuzzy set and genetic algorithm to generate efficient fuzzy rules for intrusion detection system. The system which we propose is a fuzzy classifier whose knowledge base is modelled as a fuzzy rule such as "if-then" that can be improved by our proposed fuzzy based system. The main objective is to design an efficient fuzzy classifier able to distinguish normal and abnormal behaviors of the alerts. The system starts with an initial set of fuzzy rules generated randomly, and then a fuzzy logic based process is launched to optimize the fuzzy rules.

To test the effectiveness of the proposed solution, our experiments were based on two environments namely DARPA 1999 dataset [11] and using typical network setup to generate real time dataset. In this experiment, the proposed solution was very effective. We can boldly state that our solution reduces the false alarms and improves the quality of alerts. In our work, we precisely handle 10% of the KDD Cup'99 [11] dataset. The KDD Cup'99 dataset contains 22 different types of attacks which could be classified into four types of remotely launched attacks: probes, denial of service (DOS), U2R and R2L.

We have prepared our work as follows. The second section reviews related works, the third section describes the proposed approach, the fourth section discusses experimental results and analysis, and finally, the fifth section concludes the work.

2. RELATED WORKS

Hassan ([1] and [2]), Baruah ([3] and [4]), Neog and Sut [5] have forwarded an extended definition of fuzzy set which enables us to define the complement of a fuzzy set. Our proposed method agrees with them as this new definition satisfies all the properties regarding the complement of a fuzzy set.

With the introduction of genetic based fuzzy logic technique, the false alarm rate in determining intrusive behavior can be minimized, where a set of fuzzy rules is employed to describe the normal and abnormal behavior ([1] and [2]) in a computer network. This system proposed a technique to generate efficient fuzzy rules that are able to detect intrusive behaviors of the network connections. This system presented an approach for the performance of generated fuzzy rules in classifying different types of network intrusions.

Fuzzy set theory was introduced by Zadeh [6] 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.

The normal and abnormal behaviors [7] in networked computers are hard to evaluate, as the limitations cannot be well demarcated. This evaluation process typically generates false alarms in many anomaly based intrusion detection systems.

Subhalakshmi, Mathew, and Shalinie [8] described a two phase alert management system that helps a human analyst to reduce false positives as fast as possible. They proposed a method for correct labelling of alert for real-time data with the help of human analysts using alert fusion and alert generalization that minimizes the job of the analyst significantly.

In this system, we proposed to design an efficient fuzzy classifier by introducing the concept of genetic fuzzy logic based technique to reduce false positive ratio.

3. PROPOSED APPROACH

In this paper we describe a two phase alert classification system. Fig. 1 gives an overview about this architecture of the proposed system. The first phase preprocesses and normalizes the alerts, fuse them and generalize them for root cause analysis and alert verification. After the first phase, alerts which are marked as false positives can be safely removed or labelled alerts can be passed to second phase.

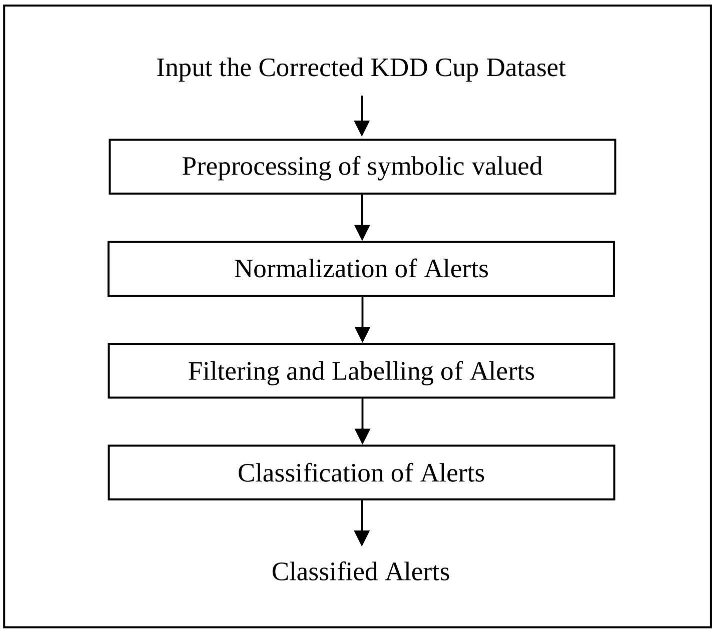
Input the Corrected KDD Cup Dataset
Preprocessing of symbolic valued
Normalization of Alerts
Filtering and Labelling of Alerts
Classification of Alerts
Classified Alerts

Fig. 1. Architecture of the proposed system

In second phase, we make use of machine learning techniques to build a classifier that automatically distinguishes true and false positives. It assists the human analysts by providing an option to discard the false positives it has classified with high classification confidence. The labelled alerts from first phase are used for the purpose of learning. Upon arrival of next batch of alerts, the classifier can be updated in a batch incremental manner. The classification rules for each batch of alerts are mentored by a human analyst. This ensures the efficiency of the proposed fuzzy classifier.

3.1. *Preprocessing of alerts*

Alerts generated by one or more IDS can be set to log into a centralized database. If we are using different types of IDS (Network, Application, and Host based) the attack messages also will be in various formats. Therefore we need a preprocessing step to be run, preferably in batch mode, before passing into the normalization component. While preprocessing the alert we try to supply best effort values for the missing features. Similarly the timestamp is converted into seconds for the purpose of comparison.

Since different IDS may use different naming conventions for the same event, we need to standardize the messages. For example, the messages ‘scanning’, ‘nmap scan’, ’port scan’ all belongs to the category ‘port scan’. The standard names are chosen from CVE (Common Vulnerabilities and Exposures) and in some cases names from one of the IDS is taken as standard. In addition, a unique id is also added to every alert for the purpose of tracking the alerts.

3.2. *Normalization of alerts*

The dataset used in the experimental study of this work are those of KDD Cup'99 [11]. The KDD Cup’99 dataset contains 22 different attack types which could be classified into four main categories namely Denial of Service (DOS), Remote to User (R2L), User to Root (U2R) and Probing. The full DARPA dataset contains 4885950 lines of connections.

Each line of the KDD Cup'99 dataset called "connection" includes a set of 41 features and a label which specifies the status of connection as either normal or specific attack type. The features of a connection include the duration of the connection, the type of the protocol (TCP, UDP, etc.), the network service (http, telnet, etc.), the number of failed login attempts, and the service and so on. These features had all forms of continuous, discrete, and symbolic, with significantly varying ranges. Among the 41 attributes of the connection, we consider only sixteen significant attributes which are: *A8, A9, A10, A11, A13, A16, A17, A18, A19, A23, A24, A32, A33, A1, A5 and A6*. These attributes are normalized. The normalization formula given in “Eq. 1” is applied in order to set attribute numerical values in the range [0.0, 1.0].

$$\mathrm{P} = P - \frac{MIN}{MAX - MIN} \qquad (1)$$

Where *MIN* is the minimum value that the attribute *P* can get, *MAX* is the maximum value, and *P* is the numerical attribute value.

Significant attributes are the important ones that can help in classifying a connection correctly. After having analyzed the KDD Cup’99 dataset, the *MIN* and *MAX* values of each significant attributes which we have selected and considered in the current work are given as Table 1.

Table 1. Significant attributes and its value

Attributes	**Description**	**Range (Normalized Value)**
A8	Number of` “wrong" fragments	[0, 3]
A9	Number of urgent packets	[0, 14]
A10	Number of` “hot" indicators	[0, 10]
A11	Number of failed login attempts	[0, 5]
A13	Number of “compromised" conditions	[0, 9]
A16	Number of` “root" accesses	[0, 7468]
A17	number of file creation operations	[0, 100]
A18	Number of shell prompts	[0, 5]
A19	Number of operations on access control files	[0, 9]
A23	Number of connections to the same host as the current connection in the past two seconds	[0, 511]
A24	Number of connections to the same service as the current connection in the past two seconds	[0, 511]
A32	Number of connection to the same host	[0, 255]
A33	Number of connection to the same host for the same services	[0, 255]
A1	Duration is number of seconds of the connection	[0, 58329]
A5	Number of data bytes from source to destination	[0, 1.3]
A6	Number of data bytes from destination to source	[0, 1.3]

However, for the numerical attributes A1, A5 and A6, we have observed a big value of *MAX*, hence the need to modify the normalization formula given in “Eq. 1”. The logarithmic scaling (with base 10) is applied to these features to reduce the range. We used all the sixteen features as the inputs of our fuzzy classifier.

3.3. *Filtering and labelling of alerts*

Once the alerts are preprocessed and normalized, it is allowed to the first phase for the purpose of filtering and labeling. First, the alerts with similar attributes other than time and which differ only by a small amount of time are fused together for the purpose of alert reduction. This is possible since multiple IDS may be there in the network which produces redundant alerts and same event may cause to trigger hundreds of similar alerts. Alert fusion also makes the process of generalization fast.

For the purpose of generalization of alerts, we require to include hierarchical background knowledge for each attribute. One of the sample hierarchy is shown in Fig. 2. We carry out generalization as a step by step process. On every iteration, one of the selected attribute is generalized to the next higher level of hierarchy and those alerts which have become similar by this generalization are grouped together. This process is repeated until one of the generalized alerts reach a threshold count.

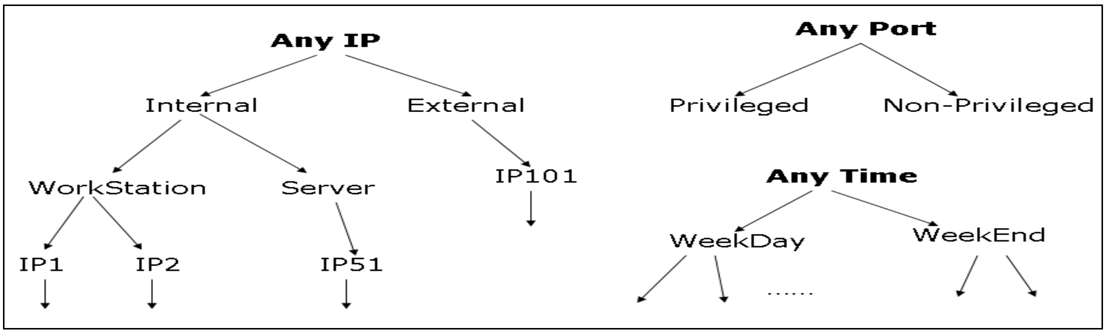

Fig. 2. Generalization hierarchies for IP address, port and timestamp

3.4. *Classification of alerts*

Unfortunately, alerts generated by IDS have to be reviewed by a mentor since no rule can assure hundred percent true positive or true negative rates. In the second phase, the labelled alerts from first phase are used for training the automatic classifier which uses fuzzy based genetic algorithm for learning the classification rules. The key objective of this phase is to construct an automatic alert classifier that reduces the workload of the human analyst. The analyst examines the rules formed by the classifier and modifies if required. The qualified rules are updated to an Alert filter which classifies the alerts as true and false positives. The generated alerts which have been categorized as false positive by the human analyst can be considered for training purpose. These rules are then used by the fuzzy classifier to classify alerts. The analyst can examine the rules to make sure they are correct.

The genetic based fuzzy classifier that we propose can be subdivided into two main stages. In the first stage, we generate randomly a set of “if-then” fuzzy rules. We used the concept of fuzzy logic in solving the problem of intrusion detection because fuzzy logic is an effective tool for introducing the concept of membership degree and the extended definition of fuzzy set [3] that determines the "strength" in which an object belongs to different classes. The goal of the second stage is to optimize the set of fuzzy rules already generated in the first stage.

The components of the fuzzy classifier for intrusion detection system are defined as follows:

3.4.1 Genetic algorithm overview

A Genetic Algorithm (GA) is a programming technique that uses biological evolution as a problem solving strategy ([2] and [7]). It is based on Darwinian’s principle of evolution and survival of fittest to optimize a population of candidate solutions towards a predefined fitness [2].

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 ([2] and [7]). 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 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 [7]. For survival and combination the selection of chromosomes is partial 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 ([2] and [7]). 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.

3.4.2 Fuzzy logic

It has been shown by Baruah [3] that a fuzzy number [a, b, c] is defined with reference to a membership function μ(x) lying between 0 and 1, a ≤ x ≤ c. Further, he has extended this definition in the following way. Let μ1(x) and μ2(x) be two functions, 0 ≤ μ2(x) ≤ μ1(x) ≤ 1. He has concluded μ1(x) the fuzzy membership function, and μ2(x) a reference function, such that (μ1(x) – μ2(x)) is the fuzzy membership value for any x. Finally he has characterized such a fuzzy number by {x, μ1(x), μ2(x); x ∈Ω}.

The complement of μ_x is always counted from the ground level in Zadehian's theory [6], 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.

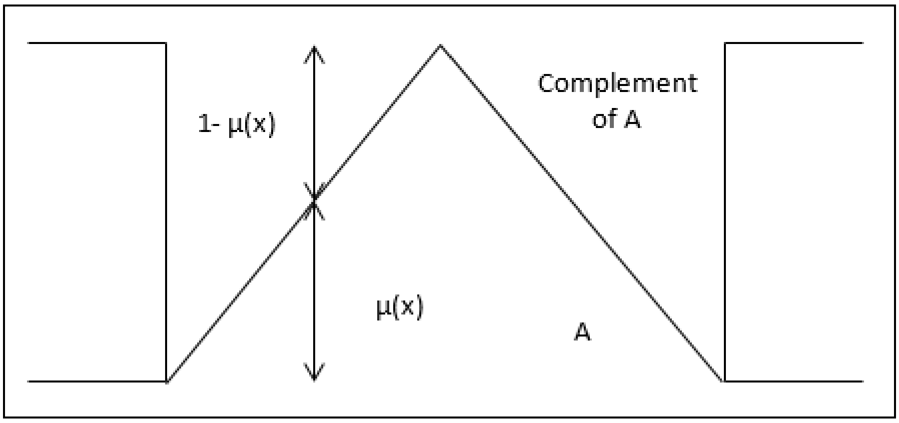

Fig. 3. Extended definition of fuzzy set

Our system forwarded a definition of complement of an extended fuzzy set where the fuzzy reference function is not always zero (Fig. 3). The definition of complement of a fuzzy set proposed by Hassan ([1] and [2]), Baruah ([3] and [4]), Neog and Sut [5] could be seen a particular case of what we are forwarding. We implement Hassan's and Baruah's definition of the complement of a normal fuzzy set in our work.

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.

3.4.3 Algorithm of the proposed system

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

```
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=0.7, W2=0.2, W3=0.5, 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.   else
11.        increment membership value of FP
12.     end if
13.   end for
14.   Fitness=W1*TP/(TP+FN)+W2*TN/(TN+FP)+W3*(1-chrom_length/10)
15.   if Fitness>T
16.     if N<1
17.          break
18.     else
19.          select the chromosome into the new population
20.          update the total number of  population
21.          N=N-1
22.     end if
23.   end if
24.   end for
25.   for each chromosome in the new population
26.     apply crossover operator to the chromosome
27.     apply mutation operator to the chromosome
28.   end for
29.   if the required number of generation is not reached, then go to step 5.
```

3.4.4 Fitness function

The authors in ([1], [2] and [7]) 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:

$$TP = \sum_{i=1}^{p} predicted\,(class\_data_i) \qquad (2)$$

$$TN = \sum_{i=1}^{q} 1 - predicted\,(other\_class\_data_i) \qquad (3)$$

$$FP = \sum_{i=1}^{q} predicted\,(other\_class\_data_i) \qquad (4)$$

$$FN = \sum_{i=1}^{p} 1 - predicted\,(class\_data_i) \qquad (5)$$

$$Sensitivity = \frac{TP}{(TP + FN)} \qquad (6)$$

$$Specificity = \frac{TN}{(TN + FP)} \qquad (7)$$

$$\text{Length} = 1 - \frac{chromosome\_length}{10} \qquad (8)$$

So finally Fitness of a chromosome is calculated as follows –

$$\text{Fitness} = W1 * Sensitivity + W2 * Specificity + W3 * Length \qquad (9)$$

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.

4. EXPERIMENT RESULTS AND ANALYSIS

4.1. *Training and testing data*

The KDD 99 intrusion detection datasets [11] is broadly used to evaluate IDSs. In this study, two subsets were extracted from the KDD Cup'99 datasets 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 ([1], [2], [7]), 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.*

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*.

4.2. *Experiments*

We have implemented the proposed system using Java. 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=0.7, W2=0.2, W3=0.5, 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.

4.3. *Numerical results and analysis*

Experimental results in Tables 2 to 6 shows some examples of the classification rates for five different classes, DOS, U2R, R2L, Probe and Normal achieved by fuzzy classifier using genetic algorithm on some network connections. The first column represents the rule taken in random order, the second column defines the classification done by fuzzy classier using the concept of genetic algorithm, the third column represents the detection rate, and the fourth column defines the fitness value of the rule.

Table 2. Fitness value for DOS class

Rule	Classification	Detection Rate	Fitness
R1	dos	0.905882353	0.69084815
R2	dos	0.901162791	0.68889915
R3	dos	0.90625	0.68721591
R4	dos	0.863157895	0.67377407
R5	u2r	0.117647059	0.24776006
R6	u2r	0.066666667	0.22213675
R7	r2l	0.166666667	0.26862745
R8	r2l	0.175257732	0.27281773

R9	r2l	0.323809524	0.35291486
R10	probe	0.626086957	0.5185074
R11	probe	0.309090909	0.34495362
R12	dos	0.076923077	0.21612111
R13	dos	0.201342282	0.28415229
R14	normal	0.36	0.37315789
R15	normal	0.746268657	0.58879956

Table 3. Fitness value for U2R class

Rule	Classification	Detection Rate	Fitness
R1	dos	0.588235294	0.53351694
R2	dos	0.2	0.33522523
R3	dos	0.279069767	0.37589036
R4	dos	0.4	0.43785714
R5	u2r	0.808510638	0.64991536
R6	u2r	0.810810811	0.64830061
R7	r2l	0.291139241	0.38520494
R8	r2l	0.168539326	0.32057103
R9	r2l	0.525641026	0.50937224
R10	probe	0.339805825	0.4138667
R11	probe	0.336734694	0.41159444
R12	dos	0.15942029	0.31758893
R13	u2r	0.626865672	0.56043096
R14	normal	0.141791045	0.3075908
R15	normal	0.225352113	0.35481579

Table 4. Fitness value for R2L class

Rule	Classification	Detection Rate	Fitness
R1	dos	0.269461078	0.35932638
R2	dos	0.308988764	0.38316737
R3	dos	0.583333333	0.53761261
R4	dos	0.48447205	0.48221325
R5	u2r	0.19047619	0.31207747
R6	u2r	0.315789474	0.37553673
R7	r2l	0.747572816	0.61453087
R8	r2l	0.584415584	0.52117529
R9	r2l	0.346938776	0.39717457
R10	probe	0.546218487	0.50855623
R11	probe	0.303571429	0.37473653
R12	r2l	0.71559633	0.59883686
R13	dos	0.203592814	0.32131142
R14	normal	0.424242424	0.44331779
R15	normal	0.564102564	0.51797261

Table 5. Fitness value for Probe class

Rule	Classification	Detection Rate	Fitness

R1	dos	0.191011236	0.30474732
R2	dos	0.335329341	0.38810347
R3	dos	0.596153846	0.53591476
R4	probe	0.86407767	0.66956399
R5	u2r	0.238095238	0.32932396
R6	u2r	0.130434783	0.2746628
R7	r2l	0.192307692	0.30592396
R8	r2l	0.168316832	0.29287585
R9	r2l	0.80733945	0.6406351
R10	probe	0.950413223	0.72311893
R11	probe	0.873873874	0.67752998
R12	dos	0.076923077	0.24165303
R13	dos	0.201342282	0.31075983
R14	normal	0.398305085	0.41886209
R15	normal	0.772727273	0.62867133

Table 6. Fitness value for Normal class

Rule	**Classification**	**Detection Rate**	**Fitness**
R1	dos	0.132867133	0.25910315
R2	normal	0.62804878	0.54246476
R3	dos	0.358974359	0.38714484
R4	dos	0.894308943	0.68038298
R5	u2r	0.19047619	0.29480632
R6	u2r	0.230769231	0.31529944
R7	r2l	0.507462687	0.46020414
R8	r2l	0.442307692	0.42480348
R9	r2l	0.623188406	0.52129288
R10	probe	0.288135593	0.34593502
R11	probe	0.195121951	0.29473079
R12	normal	0.891472868	0.68087444
R13	dos	0.577922078	0.51160678
R14	normal	0.835820896	0.65160143
R15	normal	0.96350365	0.72397645

According to the results obtained by the fuzzy classifier, it is found that the fuzzy classifier succeeds in finding good results for four classes DOS, U2R, Probe, and Normal, and false alarm rate is nominal. The success rates are 86% for DOS, 75% for Probe, 54% for Normal, and 19% for U2R classes. It is seen that the proposed method fails in achieving the success rate for R2L class, and the rate is only 10%. The following Fig. 4 explains the Detection Rate i.e. True Positive Rate (TPR) and False Positive Rate (FPR) for different classes of attacks in pyramid form.

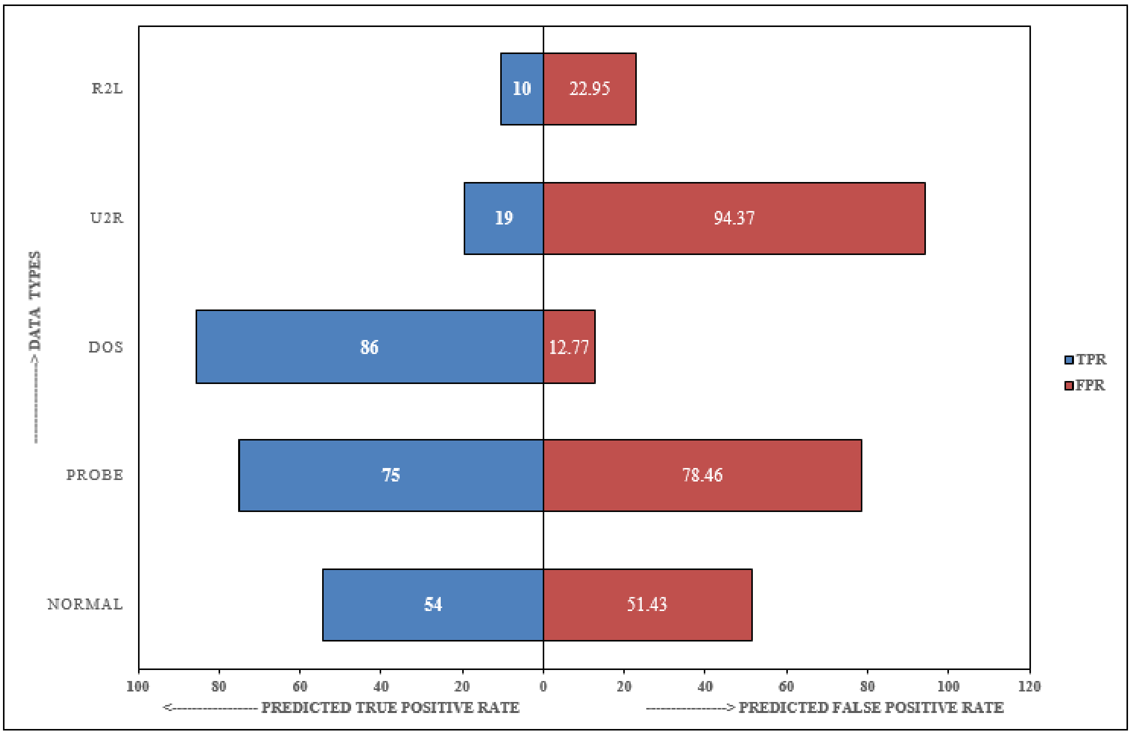

Fig. 4. Detection rate and false positive rate

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed and implemented a method of applying genetic algorithms with fuzzy logic for solving the problem of network intrusion detection system. The proposed method has produced positive alerts in reducing the false positive alerts and improved the quality of alerts sent to the analysts. The efficient rules generated by genetic based fuzzy classifier help the analysts in automatic classification of the alerts. We plan to study in future work the efficiency of fuzzy rules and genetic algorithm to detect the normal and abnormal behavior in real-time data capturing from different network communications.

References

[1] Mostaque Md. Morshedur 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.

[2] Mostaque Md. Morshedur Hassan, “Network Intrusion Detection System Using Genetic Algorithm and Fuzzy Logic”, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Issue 7, pp. 1435-1445, 2013.

[3] 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.

[4] 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.

[5] Tridiv Jyoti Neog and Dushmanta Kumar Sut, “Complement of an Extended Fuzzy Set”, International Journal of Computer Applications, Vol. 29, No.3, pp. 39-45, 2011.

[6] L A Zadeh, “Fuzzy Sets”, Information and Control, Vol.8, pp. 338-353, 1965.

[7] J. Gomez and D. Dasgupta, “Evolving Fuzzy Classifiers for Intrusion Detection”, Proceedings of the IEEE, 2005.

[8] T. Subhalakshmi, George Mathew and S. Mercy Shalinie, “Real Time Classification and Clustering of IDS Alerts using Machine Learning Algorithms”, International Journal of Artificial Intelligence and Applications (IJAIA), Vol. 1, No. 1, pp. 1-9, 2010.

[9] KDD Cup, Task Description, http://kdd.ics.uci.edu/databases/kddcup99/task.html, 1999.

[10] KDD Cup, Tasks,http://www.kdd.org/kddcup/index.php?section=1999&method=task, 1999.

[11] KDD Cup, Data, http://www.kdd.org/kddcup/index.php?section=1999&method=data, 1999.