# COMPUTER INTRUSION DETECTION BY TWO-OBJECTIVE FUZZY GENETIC ALGORITHM

Madhuri Agravat[1] and Udai Pratap Rao[2]

[1]PG Student, Dept. Of Computer Engineering, S.V.National Institute of Technology, Surat, Gujarat, India
a.madhuri@coed.svnit.ac.in
[2]Dept. Of Computer Engineering, S.V.National Institute of Technology, Surat, Gujarat, India
UPR@COED.SVNIT.AC.IN

***ABSTRACT***

*The purpose of this paper is to describe two objective fuzzy genetics-based learning algorithms and discusses its usage to detect intrusion in a computer network. Experiments were performed with KDD-cup data set, which have information on computer networks, during normal behavior and intrusive behavior. The performance of final fuzzy classification system has been investigated using intrusion detection problem as a high dimensional classification problem. This task is formulated as optimization problem with two objectives: To minimize the number of fuzzy rules and to maximize the classification rate. We show a two-objective genetic algorithm for finding non-dominated solutions of the fuzzy rule selection problem.*

***KEYWORDS***

*Intrusion Detection System, Rule Generation using Fuzzy system, Non-dominated Rule Sets, Two Objective Genetic Algorithm*

## 1. INTRODUCTION

An intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource [1]. Intrusion Detection Systems (IDS) are effective security tools, placing inside a protected network and looking for known or potential threats in network traffic and/or audit data recorded by hosts. Basically, an intrusion detection system (IDS) monitors and restricts user access (behavior) to the computer system by applying certain rules. The rules are based on expert knowledge extracted from skilled administrators who construct attack scenarios and apply them to find system exploits [2].

Intrusion detection is classified into two types: misuse intrusion detection and anomaly intrusion detection. A misuse detection model takes decision based on comparison of user's session or commands with the rule or signature of attacks previously used by attackers. The main advantage of misuse detection is that it can accurately and efficiently detect occurrence of known attacks. However, these systems are not capable of detecting attacks whose signatures are not available [1]. To remedy the problem of detecting novel attacks, anomaly detection attempts to construct a model according to the statistical knowledge about the normal activity of the computer system [2].

Fuzzy systems based on fuzzy if-then rules have been applied to various problems [3]. One advantage of fuzzy-rule-based systems is their clarity. Human users of such systems can easily understand each fuzzy if-then rule because its antecedent and consequent are related to linguistic values such as “low”, “medium” and “high”. The number of fuzzy if-then rules is also closely connected to the clarity of fuzzy systems. If a single fuzzy system consists of thousands of fuzzy if-then rules, it is difficult for human users to carefully examine each rule. Therefore we should

D.C. Wyld, et al. (Eds): CCSEA 2011, CS & IT 02, pp. 281–292, 2011.
 DOI: 10.5121/csit.2011.1226

choose a small number of fuzzy if-then rules for constructing a fuzzy system that is easily understood by human users. Recently a genetic-algorithm-based approach was proposed for constructing a compact fuzzy classification system with a small number of fuzzy if-then rules. Genetic algorithms have been used as rule selection and optimization tools in the design of fuzzy rule-based systems. Those GA-based studies on the design of fuzzy rule-based systems are usually referred to as fuzzy genetics-based machine learning methods (fuzzy GBML methods)[4][5], each of which can be classified into the Michigan, Pittsburgh or iterative rule learning (IRL) approaches [2][6].

In this paper, we use fuzzy GBML methods to develop a two objective IDS based on misuse detection. We are generating signatures in the form of rules for every known attack. Our aim is to generate signatures which,

i) Maximize detection rate,

(ii) Contains minimum number of rules with low false rate.

These two objectives were combined into a single scalar fitness function and genetic algorithms are applied on same fitness function which generates rules for classification of known patterns. The whole block diagram of the system is shown in Figure 1.

The rest of the paper is as follows: Related Work is presented in 1.1. Background is presented in II. Fuzzy rule base for pattern classification is presented in section III. Two objective genetic algo is presented in IV. Experimental results are reported in Section V. And last we conclude the work.

## 1.1Related Work

Nowadays, There are many approaches for solving intrusion detection problems. Lee built intrusion detection models that can that can recognize anomalies and known intrusions. He proposed to use the association rules and frequent episodes computed from audit data as the basis for guiding the audit data gathering and feature selection processes [7].

Mukkamala shows Feature Selection for Intrusion Detection using Neural Networks and Support Vector Machines. He addresses the related issue of ranking the importance of input features that elimination of the insignificant and/or useless inputs leads to a simplification of the problem and possibly faster and more accurate detection, feature selection is very important in intrusion detection[8].

Some other applied techniques on intrusion detection problem are genetic algorithms Mohammad Saniee Abadeh [2] proposed Computer Intrusion Detection Using an Iterative Fuzzy Rule Learning Approach. The proposed method is based on the iterative rule learning approach (IRL) to fuzzy rule base system design. The fuzzy rule base is generated in an incremental fashion, in that the evolutionary algorithm optimizes one fuzzy classifier rule at a time. Performance of this system has been evaluated using intrusion detection problem as a high dimensional classification problem. Tansel O zyer [9] proposed a method based on iterative rule learning using a fuzzy rule-based genetic classifier. His approach is mainly composed of two phases. First, a large number of candidate rules are generated and they are pre-screened using two rule evaluation criteria. He employs Boosting genetic algorithm that evaluates the weight of each data item to help the rule extraction mechanism focus more on data having relatively more weight.

Cho and Cha[10] empirically demonstrate that the Bayesian parameter estimation method is effective in analysing web logs and detecting anomalous sessions. They developed a technique, session anomaly detection (SAD) which has detected nearly all such attacks without having to rely on attack signatures at all. SAD works by first developing normal usage profile and comparing the web logs, as they are generated, against the expected frequencies. He develops SAD to provide secure and reliable web services only.

Saqib Ashfaq[11] has proposed Efficient Rule Generation for Cost-Sensitive Misuse Detection Using Genetic Algorithms. He employs only the five most relevant features for each attack

category for rule generation. Furthermore, it incorporates the different costs of misclassifying attacks in its fitness function to yield rules that are cost sensitive.

M. Saniee Abadeh[12] proposed Design and analysis of genetic fuzzy systems for intrusion detection in computer networks. He present three kinds of genetic fuzzy systems based on Michigan, Pittsburgh and iterative rule learning (IRL) approaches to deal with intrusion detection as a high-dimensional classification problem.

Hu proposes a data mining technique to discover fuzzy classification rules based on the Apriori algorithm. In his technique, genetic algorithms are incorporated into the proposed method to determine minimum support and confidence with binary chromosomes[13].

Some recent researches have utilized artificial immune systems to detect intrusive behaviors in a computer network [14].

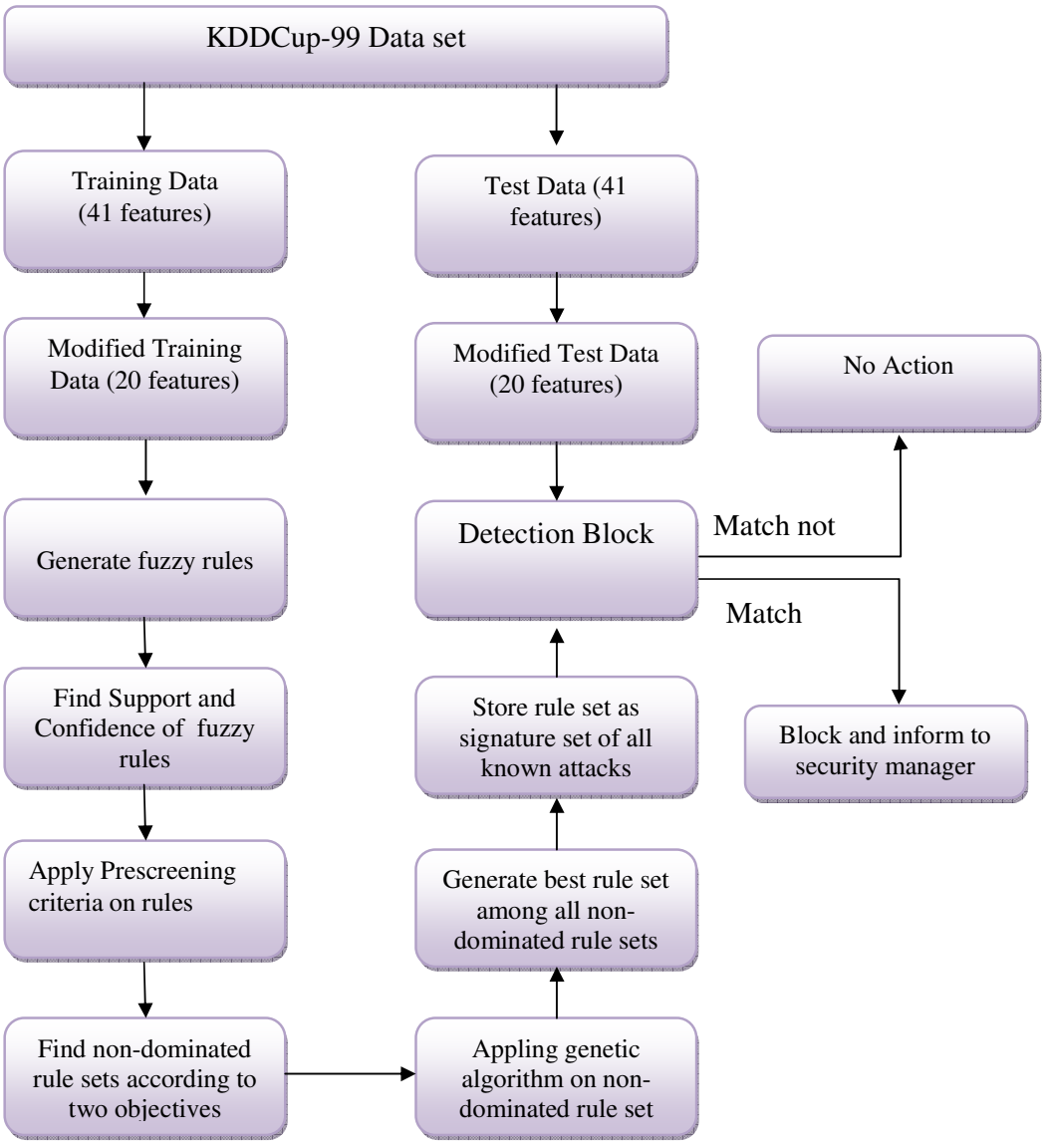


Figure 1. Block Diagram of Computer Intrusion Detection by Two Objective Genetic Algorithms

## 2. THE BACKGROUND

### 2.1. Fuzzy logic

A classical set is characterized by having the membership degree of an element takes only one of two values as either 0 or 1. It is a set with a sharp boundary, where there are no unambiguous boundaries. In other words, an object is either entirely belongs to set or not. Whereas a fuzzy set as its name implies is a set without sharp boundaries. The transition from ''belonging to a set'' to ''not belonging to a set'' is gradual; and this smooth transition is characterized by membership functions that give flexibility in modeling commonly used linguistic expressions. Membership is not restricted to two values; rather it may take any value from the range (0, 1). This reflects a degree of membership and this represents uncertainty as practiced daily by humans. Fuzziness comes from the uncertain and imprecise nature of abstract thoughts and concepts [3,4,5].

Let assume, X represents the universe of discourse. If X is a collection of objects denoted each by x, then fuzzy set A is a set of ordered pairs as below:

$A = \{x, \mu_A(x) | x \square X\}$,

Where $\mu_A$ is called the membership function that maps each object x of domain X to a continuous membership value between 0 and 1.

There are several classes of parameterized ways to define membership functions, like trapezoidal, bell functions, Gaussian and triangular. A parameterized membership function can be defined in terms of a number of parameters. For example, a triangular membership function is specified by three parameters (a, b, c); and for a given value x, with known a, b, and c, the membership of x may be computed as:

$$\text{Triangle}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right).$$

A fuzzy space having a normalized domain may be partitioned with five linguistic variables (L, LM, M, MH, H) and each linguistic variable is a parameterized triangular membership function as shown in Figure 2.

A given object x may be member of a given fuzzy set with a certain membership degree. Object x may also be member of other fuzzy sets at the same time, but with different membership degree values.

IF $x_1$ is $A_{j1}$ and $x_2$ is $A_{j2}$ and….and n is $A_{jn}$ THEN class is $c_j$,where $R_j$ is the jth fuzzy rule, x=( $x_1$, $x_2$, . . . , $x_n$) is an n-dimensional object of X, $c_j$ is the consequent class and each $A_{ji}$ is an antecedent fuzzy set. If the degree of membership () of an object with each corresponding antecedent $A_{ji}$ is denoted $\mu_i$, then the strength $\mu_{Aj}$ of a rule is $\mu_{Aj} = \min(m_1, m_2, \ldots, m_n)$.

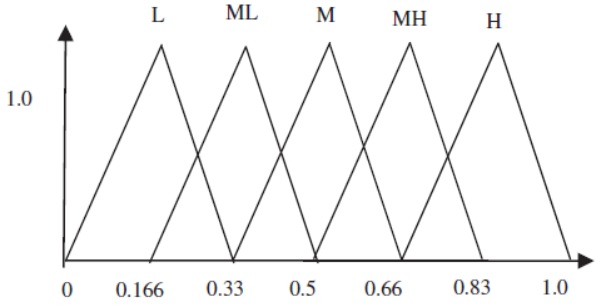


Figure 2. Fuzzy space partitioned with five fuzzy classes (L—low, LM—low medium, M medium, MH—medium high, H—high)[9].

## 3. RULE GENERATION FROM TRAINING DATA

Let us assume that our pattern classification problem is a c- class problem in the n-dimensional pattern space with continuous attributes. We also assume that m real vectors $x_p = (x_{p1}, x_{p2}, ..., x_{pn})$, p = 1, 2,..., m, are given as training patterns from the c classes ( c << m ).Because the pattern space is $[0, 1]^n$, attribute values of each pattern are $x_{pi} \in [0,1]$ for p=1,2,...,m and i= 1,2,...,n. In computer simulations of this paper, we normalize all attribute values of each data set into the unit interval [0, 1]. In the presented fuzzy classifier system, we use fuzzy if then rules of the following form.

*Rule $R_j$: If $x_1$ is $A_{j1}$ and ... and $x_n$ is $A_{jn}$ , then Class $C_j$ with CF=$CF_j$.* (1)

Where $R_j$ is the label of the $j^{th}$ fuzzy if-then rule, $A_{j1}$ . ..$A_{jn}$ are antecedent fuzzy sets on the unit interval [0,1], $C_j$ is the consequent class (i.e., one of the given c classes), and $CF_j$ is the grade of certainty of the fuzzy if-then rule $R_j$. In computer simulations, we use a typical set of linguistic values in Fig. 1 as antecedent fuzzy sets. The membership function of each linguistic value in Fig. 1 is specified by homogeneously partitioning the domain of each attribute into symmetric triangular fuzzy sets. We use such a simple specification in computer simulations to show the high performance of our fuzzy classifier system, even if the membership function of each antecedent fuzzy set is not tailored. However, we can use any tailored membership functions in our fuzzy classifier system for a particular pattern classification problem.

The total number of fuzzy if-then rules is $5^n$ in the case of the n-dimensional pattern classification problem. It is Impossible to use all the $5^n$ fuzzy if-then rules in a single fuzzy rule base when the number of attributes (i.e. n) is large (e.g., intrusion detection problem which n = 41). Our fuzzy classifier system searches for a relatively small number of fuzzy if-then rules with high classification ability. Initially, we consider all the training pattern as rules. Since the consequent class and the certainty grade of each fuzzy if-then rule can be determined from training patterns by a simple heuristic procedure, the task of our fuzzy classifier system is to generate combinations of antecedent fuzzy sets for a set of fuzzy if-then rules.

To determine $C_j$ and $CF_j$ of each rule in the population the following steps should be done:

Step 1: Calculate the compatibility of each training pattern $x_p = (x_{p1}, x_{p2}, ... ,x_{pn})$ with the fuzzy if-then rule $R_j$ by the following product operation:

$$\mu_j(x_p) = \mu_{j1}(x_{p1}) \times ..... \times \mu_{jn}(x_{pn}), \quad (2)$$

where $\mu_{ji}(x_{pi})$ is the membership function of $i^{th}$ attribute of $p^{th}$ pattern and m denotes total number of patterns.

Step 2: For each class, calculate the relative sum of the compatibility grades of the training patterns with the fuzzy if-then rule $R_j$ :

$$\beta_{Class\ h}(R_j) = \sum_{x_p \in classh} \mu_j(x_p) \Big/ N_{classh} \quad , h = 1,2, ... ,c \quad (3)$$

where $\beta_{Class\ h}(R_j)$ is the sum of the compatibility grades of the training patterns in Class h with the fuzzy if-then rule $R_j$ and $N_{classh}$ is the number of training patterns which their corresponding class is Class h .

Step 3: Find Class $h_j$ that has the maximum value of $\beta_{Class\ h}(R_j)$ :

$$\beta_{Class\ hj}(R_j) = max\{ \beta_{Class\ 1}(R_j), ....., \beta_{Class\ c}(R_j) \}. \quad (4)$$

If two or more classes take the maximum value, the consequent Class $C_j$ of the fuzzy if-then rule $R_j$ cannot be determined uniquely. In this case, let $C_j$ be null. If a single class takes the maximum

value, let $C_j$ be Class $_{hi}$ .If there is no training pattern compatible with the fuzzy if-then rule $R_j$ (i.e., if $\beta_{Class\ h}(R_j) = 0$ for h = 1, 2,. . ., c ) the consequent Class $C_j$ is also specified as null. When $c_j$ is null we don't consider them in ruleset.

Step 4: When the consequent class $C_j$ is determined by (4), the certainty grade $CF_j$ is specified as

$$CF_j = ( \beta_{Class\ hj}(R_j) - \beta\square)/ \sum_{h=1}^{c} \beta_{Class\ h}(R_j) , \quad (5)$$

Where,

$$\beta\square = \sum_{h\neq c_j} \beta_{Class\ h}(R_j)/(c-1) \quad (6)$$

By the proposed heuristic procedure we can specify the consequent class and the certainty grade for any combination of antecedent fuzzy sets. Such a combination is generated by a fuzzy classifier system. The task of our fuzzy classifier system is to generate combinations of antecedent fuzzy sets for generating a rule set S with high classification ability. When a rule set S is given, an input pattern $x_p = (x_{p1}, x_{p2} \ldots ,x_{pn})$ is classified by a single winner rule $R_j$ in S, which is determined as follows:

$$R_j (x_p).CF_j = \max\{ R_j(x_p).CF_j | R_j \in S\}. \quad (7)$$

That is, the winner rule has the maximum product of the compatibility and the certainty grade $CF_j$ In this procedure, a new pattern $x_p = ( x_{p1},..., x_{pn} )$ is classified by the linguistic rule that has the maximum product of $R_j ( x_p )$ and $CF_j$ [1].

Fuzzy if-then rules in this approach are coded as a string. The following symbols are used for denoting the five languishing values:(Fig.1) 1:low(L), 2:medium low(ML), 3:medium(M), 4:medium high(MH), 5:high(H). This approach consists c the number of classes. Each classifier contains a subset of rules with the same labels. The proposed algorithm focuses on learning of each class to improve the total accuracy of the main classifier. Therefore, this evolutionary fuzzy rule learning algorithm repeated for each class of the classification problem separately.

## 4. PROBLEM FORMULATION

Our rule selection problem is to select a smaller number of linguistic rules from the rule set S to construct a compact classification system. Therefore our problem can be written as follows:

Maximize $NCP(S)$ and (i)

minimize $|S|$ , subject to $S \in S_{ALL}$ , (ii)

where $NCP(S)$ is the number of correctly classified training patterns by linguistic rules in a rule set $S$, and $|S|$ is the number of the linguistic rules in $S$. here we select $N_{pop}$ rules from the descending order of $CF_j$.

### 4.1 Two objective Genetic Algorithm

We use two objective genetic algorithms to the rule selection problem. Its scalar fitness function is defined below:

$$f(S) = W_{NCP}.NCP(S) - W_S .|S| \quad (8)$$

Where $W_{NCP}$ and $W_S$ are positive constant weights. Because the weight for each objective in the fitness function is constant, the choice of the weight values in (8) has a significant effect on the final solution (*i.e.*, rule set *S*) obtained by the genetic algorithm. Because the importance of each objective in the rule selection problem depends on the preference of human users, it is not easy to assign constant values to the weights $W_{NCP}$ and $W_S$. Therefore we can find multiple non-dominated solutions of the two-objective rule selection problem. Before we go further, let first discuss how to find non dominated set with respect to two objectives.

Based on this discussion, we formulate our task of designing comprehensible fuzzy rule based with high classification systems as the following :

Maximize $f_1(S)$, minimize$f_2(S)$,

Where $f_1(S)$ is correctly classified training patterns by ruleset S, $f_2(S)$ is number of fuzzy rules in S.A ruleset S is said to be dominated by another ruleset S* if two following in equalities' hold:

$$f_1(S) <= f_1(S^*) \quad , \quad f_2(S) >= f_2(S^*), \tag{9}$$

and at least one of the following inequalities' holds:

$$f_1(S) < f_1(S^*) \quad , \quad f_2(S) > f_2(S^*), \tag{10}$$

The first condition means that no objective of S* is worst than S. The second condition means that at least one objective of S* is better than S. If there exists no S* that satisfies above both conditions than S* is called non-dominated ruleset with respect to S [4]. The characteristic feature of two objective genetic algorithm is that non-dominated rule sets are stored in a tentative pool separately from the current population. Tentative pool is updated at every generation in order to store only non-dominated rule sets among examined ones. From the tentative pool, $N_{elite}$ ruleset randomly selected as elite individuals, which are added to new population. Human users will choose a final solution (*i.e.*, rule set *S*) from the obtained non-dominated solutions.

Here, each rule set *S* is treated as an individual in our two-objective genetic algorithm. Each rule set *S* (*i.e.*, each individual) is presented by a string as $S = s_1\ s_2 \ldots\ s_r$, where *r* is the number of all the linguistic rules in S. So, any rule set is presented in "0","1" string sequence.suppose,$s_1$ is in S then its place is filled by 1 if its not in S then its place is filled by 0. So, this way any ruleset is coded in the sequence of "0","1"string.

The selection probability in our two-objective genetic algorithm is specified according to the fitness function $f(S)$ in (8) with randomly specified weight values. That is, when each pair of parent individuals are selected, the values of the weights $W_{NCP}$ and $W_S$ are assigned as,

$W_{NCP}$ : a random real number in [0, 1],
$W_S$ : $W_S = 1 - W_{NCP}$ .

In two-objective genetic algorithm, multiple solutions are preserved from the current generation to the next generation as elite solutions. Those elite solutions are randomly selected from a tentative set of non-dominated solutions that is stored and updated at each generation of two-objective genetic algorithm IDS. Multiple search directions are realized by the selection procedure with random weight values and the elitist strategy with multiple elite solutions [15].

The outline of two-objective genetic algorithm can be written as follows:
Step 0 (Initialization): Generate an initial population containing $N_{pop}$ strings where $N_{pop}$ is the number of possible solution strings for current scenario.
Step 1 (Evaluation): Calculate the values of the two objectives for the generated strings. Update the tentative set of non-dominated solutions.

Step 2 (Selection): Calculate the fitness value of each string using random weight values. Select a pair of strings from the current population according to the following selection probability. The selection probability $P(S)$ of a string $S$ in a population $S_{ALL}$ is specified as

$$p(S)\frac{fitness(S)-fitness_{\min}(S)}{\sum_{S\in S_{All}}\{fitness(S)-fitness_{\min}(S)\}},$$

*Where* $fitness_{min}(S) = min\{fitness(S) \mid S \in S_{All}\}$ .
Here we are using "*fitness "and "f* "in the same meaning so don't be confused in. This procedure is repeated for selecting $N_{pop}$ / 2 pairs of parent strings.
Step 3 (Crossover): For each selected pair, apply a crossover operation to generate two strings.
Step 4 (Mutation): For each value of the generated strings by the crossover operation, apply a mutation operation with a pre-specified mutation probability.
Step 5 (Elitist strategy): Randomly remove $N_{elite}$ strings from the generate $N_{pop}$ strings, and add $N_{elite}$ solutions that are randomly selected from the tentative set of non-dominated solutions.
Step 6 (Termination test): If a pre-specified stopping condition is not satisfied, return to Step1.

## 5. EXPERIMENTAL EVALUATION

The fuzzy genetic algorithm for misuse detection is implemented in Java, tested and evaluated on the KDDCup 99 dataset [16]. We use the 10% labeled data (file name: kddcup.data 10_percent.gz) for training and testing of the genetic algorithm. KDDCup 99 dataset has 41 attributes in which we have used 20 attributes; 8 basic and remaining are domain knowledge features. Five output classes are namely Normal, PRB-probe, DOS-denial of service, U2R-user to root and R2L-remote to local. Selected attributes are shown in below table 1.

Table 1. Selected attributes with description

| Selected attributes | description | Types |
|---|---|---|
| duration | length (number of seconds) of the connection | continuous |
| protocol_type | type of the protocol, e.g. tcp, udp, etc. | symbolic |
| flag | normal or error status of the connection | symbolic |
| src_bytes | number of data bytes from source to destination | continuous |
| Dest_bytes | number of data bytes from destination to source | continuous |
| land | 1 if connection is from/to the same host/port; 0 otherwise | symbolic |
| Wrong_fragment | number of "wrong" fragments | continuous |
| urgent | number of urgent packets | continuous |
| Hot | number of ``hot" indicators | continuous |
| Num_failed_logins | number of failed login attempts | continuous |
| Logged_in | 1 if successfully logged in; 0 otherwise | symbolic |
| Num_compromised | number of ``compromised" conditions | continuous |
| Root_shells | 1 if root shell is obtained; 0 otherwise | continuous |
| Su_attempted | 1 if ``su root" command attempted; 0 otherwise | continuous |
| Num_root | number of ``root" accesses | continuous |
| Num_file_creations | number of file creation operations | continuous |
| Num_shells | number of shell prompts | continuous |
| num_access_files | number of operations on access control files | continuous |

| num_outbound_cmds | number of outbound commands in an ftp session | continuous |
|---|---|---|
| is_host_login | 1 if successfully host logged in; 0 otherwise | symbolic |

We discuss the experimental evaluation of applying genetic fuzzy systems on the intrusion detection data set. The parameter specifications that we have used in our computer simulations are shown below.

Number of elite solutions: $N_{elite}$ =20%
Crossover probability: =0.9
Mutation probability: = 0.1
Number of generation=50

Table 2. Specification of number of training and testing data

| Class | Train | Test |
|---|---|---|
| Normal | 500 | 200 |
| U2R | 100 | 100 |
| R2L | 200 | 200 |
| DOS | 500 | 1000 |
| PRB | 100 | 200 |

Here we are going to find non dominated ruleset. So, first we decide constraint for that suppose $f_1(S)$ =1600 and $f_2(S)$ =70. We get bellow different non dominated rule set. We can find many different solutions for the same constraints. We have shown very few among them in Table 4. , by which two objectives are going to be satisfied. Here suppose our rule is:"attribute a1 is low, attribute a2 is medium-high, attribute a10 is low and attribute a15 is medium Then Class is normal", this rule's antecedent part is coded as "L MH L M". We are not coded its consequent part because that we are going to find out for new pattern. Here we are doing crossover and mutation operation among same class.

The performance of the system is evaluated using Precision, recall and Overall accuracy.

Precision= $\frac{TP}{TP + FP}$

Recall = $\frac{TP}{TP + FN}$

Overall accuracy = $\frac{TP + TN}{TP + TN + FN + FP}$

Where,

TP = True positive
TN = True negative
FN = False negative
FP = False positive

These are computed using the confusion matrix in Table 3, and defined as follows:

Table 3. The confusion matrix

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

Here, Positive Class means all attack class and Negative Class means Normal class.TP(True Positive) means attack class is classified correctly. FP(False Positive) means normal class misclassified to attack class. TN(True Negative) means normal class is classified correctly. FN(False Negative) means attack class is misclassified to normal class.

Table 4. Non dominated ruleset

| \|S\| | 30 | 40 | 50 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|
| NCP(S) | 1100 | 1100 | 1250 | 1400 | 1650 | 1695 |

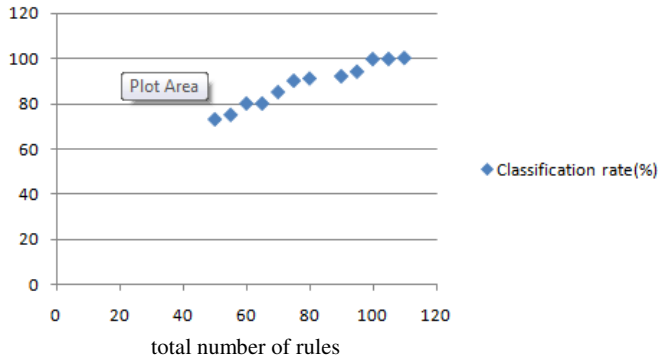


Figure 3. Number of rules vs classification rate

Computer Intrusion Detection by two objective genetic algorithms is a five-class problem with 19 attributes. Non-dominated solutions are obtained by genetic algorithms are shown in Table 4, where 1650 patterns are correctly classified by 100 rules, 1650 patterns by 90 linguistic rules and so on. If the human user prefers a high detection rate, he/she would choose the rule set with 100 linguistic rules in Table 4. On the contrary, if the human user prefers the compactness of rule sets to the high classification performance, he/she would choose the rule set with 90 linguistic rules and gives 99.8% detection rate. In Figure 3 we shows classification rate of the system. By analysing the result from Table 5, the overall performance of the proposed system is improved significantly and it achieves 99% accuracy for all types of attacks.

Table 5. The Classification performance of the proposed NIDS

| | Metric | Proposed system |
|---|---|---|
| \|S\|=90 | Precision | 0.9979 |
| | Recall | 1 |
| | Accuracy | 0.985 |

| | | |
|---|---|---|
| \|S\|=100 | Precision | 1 |
| | Recall | 0.998 |
| | Accuracy | 0.9983 |

## APPENDIX A. LEARNING OF LINGUISTIC CLASSIFICATION RULES

The classification accuracy of fuzzy rule-based systems can be improved by adjusting the rule weight of each fuzzy if-then rule [15]. When a training pattern $x_p$ is correctly classified by the winner rule $R\hat{}_j$ in a rule set, its rule weight $CF\hat{}_j$ is increased as

$$CF\hat{}_j^{new} = CF\hat{}_j^{old} + \eta_1 \cdot (1 - CF\hat{}_j^{old}), \quad (11)$$

Where $\eta_1$ is a learning rate for increasing rule weights. The rule weights of the other rules in the rule set are not changed. On the other hand, when the training pattern xp is misclassified by the winner rule $R\hat{}_j$, its rule weight $CF\hat{}_j$ is decreased as

$$CF\hat{}_j^{new} = CF\hat{}_j^{old} - \eta_2 \cdot (1 - CF\hat{}_j^{old}), \quad (12)$$

Where $\eta_2$ is a learning rate for decreasing rule weights. The rule weights of the other rules are not changed.

## 3. CONCLUSIONS

We use the idea of two objective fuzzy genetic based classifications for intrusion detection. This classification algorithm uses a specified number of fuzzy rules obtained from the non-dominated ruleset with two objectives: to maximize the number of correctly classified training patterns and to minimize the number of selected rules. It will reduce the search space of finding rules for new patterns and also takes lesser CPU time than without usage of non-dominated rule sets. We can extend the two-objective genetic algorithm to a hybrid algorithm where a learning method given in appendix A could be applied to rule sets generated by genetic operations. The selection of a final rule set from the obtained non-dominated solutions should be done based on the preference of human users for a NIDS.

## REFERENCES


[1] Abhinav Srivastava, Shamik Sural and A.K. Majumdar, "Database Intrusion Detection using Weighted Sequence Mining" Journal Of Computers, Vol. 1, No. 4, July 2006.

[2] Mohammad Saniee Abadeh and Jafar Habibi,"Computer Intrusion Detection Using an Iterative Fuzzy Rule Learning Approach", 1-4244-1210-2/07,2007 IEEE. Technologies, Page(s):233 - 240, 27-28 Aug. 2005.

[3] 0. Cordon, F. Gomide, F. Herrera, F. Hofmann, L. Magdalena, "Ten years of genetic fuzzy systems current framework and new trends", Fuzzy Sets and Systems 141, pp. 5-31, 2004.

[4] Ishibuchi et al., 2001 H. Ishibuchi, T. Nakashima and T. Murata, Three-objective genetics-based machine learning for linguistic rule extraction, Information Sciences (2001), pp. 109–133.

[5] Ishibuchi et al., 2005 H. Ishibuchi, T. Yamamoto and T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification problems, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 35 (2) (2005).

[6] Hisao Ishibuchi, Tomoharu Nakashima and Tadahiko Murata," The Comparison of Michigan and Pittsburgh Approaches to the design of Fuzzy Classification System",Electronics and Communications in Japan,Part 3,Vol. 80,No. 12,1997.

[7] Lee et al., 1998 Lee, W., Salvatore, J. S., & Mok, K. W. M. (1998). Mining audit data to build intrusion detection models. In Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining (pp. 66–72).

[8] Mukkamala et al., 2003 Mukkamala, S., & Sung, A. H. (2003). Feature selection for intrusion detection using neural networks and support vector machines. Journal of the Transport Research Board National Academy. Transport research record no. 1822, pp. 33–39. Full Text via CrossRef | View Record in Scopus | Cited By in Scopus (21).

[9] Tansel O¨ zyer, Reda Alhajja, Ken Barker," Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening ",Journal of Network and Computer Applications 30 (2007) 99–113.

[10] Cho and Cha, 2004 S. Cho and S. Cha, SAD: Web session anomaly detection based on parameter estimation, Computers & Security 23 (4) (2004), pp. 265–351.

[11] Saqib Ashfaq, M. Umar Farooq, Asim Karim", Efficient Rule Generation for Cost-Sensitive Misuse Detection",Using Genetic Algorithms",1-4244-0605-6/06/,2006 IEEE.

[12] Mohammad Saniee Abadeh, Hamid Mohamad and Jafar Habibi,"Design and analysis of genetic fuzzy systems for intrusion detection in computer networks", Available online 24 December 2010.

[13] Yi-Chung Hu, Ruey-Shun Chen, Gwo-Hshiung Tzeng, "Finding fuzzy classification rules using data mining techniques," Pattern Recognition Letters 24, pp. 509-519, 2003.

[14] Dasgupta and González, 2002 D. Dasgupta and F. González, An immunity-based technique to characterize intrusions in computer networks, IEEE Transactions on Evolutionary Computation 6 (3) (2002).

[15] H. Ishibuchi, T. Murata, and I. B. Turksen, " Selecting linguistic classification rules by two-objective genetic algorithms," Proc. of 1995 IEEE International Conference on Systems, Man and Cybernetics, pp. 1410-1415, Vancouver, Canada, October 1995.

[16] KDD-Cup data set :< http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

**Authors**

**Madhuri Agravat** a M.Tech. student from Sardar Vallabhbhai National Institute Technology, Surat, Gujarat, India. She comleted her graduate from Nirma University, Ahmedabad, Gujarat, India.

**Udai Pratap Rao** received the B.E. degree in Computer Science and Engineering in 2002 & M.Tech degree in Computer Science and Engineering in 2006, and currently working as Assistant Professor in the Department of Computer Engineering at S. V. National Institute of Technology Surat (Gujarat)-INDIA. His research interests include Data Mining, Database security, Information Security, and distributed systems.