ISSN:0975-9646

Intrusion Detection System using Genetic-Fuzzy Classification

Prabhat Prakash[1], Dr. Rajendra Kumar Bharti[2]

Computer Science & Engineering Department,
Bipin Tripathi Kumaon Institute of technology
Dwarahat, Uttarakhand, India

Abstract— **Intrusion detection system has become the prime focus in the area of network security research. An effective intrusion detection system must detect the previously known attacks as well as variations of known attacks and unknown attacks. The challenging and critical problem in intrusion detection is the classification of intrusion attacks and normal network traffic. Fuzzy systems have been used to solve several classification problems. Genetic-fuzzy systems hybridize the approximate reasoning method of fuzzy systems with the learning capability of evolutionary algorithms. In this paper a novel intrusion detection method is presented, capable of detecting normal and intrusive behaviours, which extracts both accurate and interpretable fuzzy IF-THEN rules from network dataset for classification. This method uses the fuzzy association rule based classification method for high dimensional problems based on three stages to obtain an accurate and compact fuzzy rule based classifier with a low computational cost. Experiments were performed with KDD-cup 99 dataset, which contains information of computer networks, during normal and intrusive behaviours. The result of the proposed intrusion detection model is compared with some well-known classifiers.**

Keywords— **Intrusion detection, Genetic-fuzzy rule based classification, Fuzzy association rules, KDD-cup 99.**

I. INTRODUCTION

Intrusions refer to the actions; attempt to compromise the integrity, confidentiality or availability of a resource [1]. It is the act of a person or proxy attempting to break into or misuse one's system in violation of an established policy. Intrusions result in services being denied, system failing to respond, data stolen or being lost. Intrusion detection means detecting unauthorized use of a system or attacks on a system or network. An Intrusion Detection System monitors and restricts user access to the computer system by applying certain rules.

Based on analysis strategy, Intrusion detection system is categorized into misuse and anomaly IDS. When the IDS looks for events or sets of events that match a predefined pattern of a known attack, this analysis strategy is called misuse detection. The effectiveness of misuse IDS is largely based on the validity and expressiveness of their database of known attacks and misuse, and the efficiency of the matching engine that is used. The disadvantage of misuse IDS is that it requires frequent updates to keep up with the new stream of vulnerabilities discovered and it cannot detect unknown attacks. When the IDS identifies intrusions as unusual behaviour that differs from the normal behaviour of the monitored system, this analysis strategy is called anomaly detection. Anomaly detection approaches attempt to build some kind of a model over the normal data and then check to see how well new data fits into that model. In other words, anything that does not correspond to a previously learned behaviour is considered intrusive. Therefore, the intrusion detection system might not miss any attack, but its accuracy is a difficult issue, since it can generate a lot of false alarms.

One of the most effective methods to automate and simplify the development of intrusion signatures, and to predict novel attacks is learning classification rules from network data, if the generalized knowledge can be extracted from data. Fuzzy rule based classification systems (FRBCSs) are well known tools in the machine learning framework, since they can provide interpretable model [2]. Association discovery is one of the most common data mining techniques which are used to extract relationships between different items in a large dataset [3]. It has been used for classification under the name of associative classification [4]. Genetic algorithms have been used for rule generation and optimization methods in the design of fuzzy rule based classifier [5]. The genetic algorithm based design of FRBCSs is usually referred as GFRBCSs.

Genetic-fuzzy rule based classification and data mining have been used previously to solve the intrusion detection problem. In [6], a data mining framework is proposed for constructing intrusion detection models. In [7], a prototype IIDS (Intelligent Intrusion Detection System) is proposed, which is both anomaly and misuse detector. The anomaly-based components are developed using fuzzy data mining techniques. The method EFRID, proposed in [8], classifies the system behaviour by fuzzy rules. In [9], a multi-objective genetic fuzzy intrusion detection system (MOGFIDS) is proposed which applies an agent-based evolutionary computation framework to generate and evolve an accurate and interpretable fuzzy knowledge base for classification. In [10], authors proposed a novel fuzzy method with genetic algorithm for detecting intrusion data from the network database. In this approach GA is implemented using directed graph structures instead of strings in genetic algorithm or trees in genetic programming, which leads to enhancing the representation ability with compact programs derived from the reusability of nodes in a graph structure. In [11], the IDS uses fuzzy association rules for binding fuzzy classifiers. In this method an immune-inspired algorithm is proposed for mining fuzzy

association rule set, in which the fuzzy sets corresponding to each attribute and the final fuzzy rule set can be directly extracted from a given dataset. In [12], a hybrid fuzzy genetic rule based inference engine has been designed. The fuzzy logic constructs precise and flexible patterns while the GA helps in attaining optimal solution.

This paper presents an approach to IDS using genetic-fuzzy rule based system and association discovery. The experiments were performed out on KDD-cup 99 dataset [13] and the results were compared with some well-known IDS classifiers.

II. Preliminaries

This section discusses fuzzy rule based classification systems and fuzzy association rules for classification.

A. Fuzzy Rule Based Classification

Any classification problem consists of N training patterns, i.e., $x_p = (x_{p1}, \ldots, x_{pm})$, p = 1, 2, . . .,N; from S classes, where x_{pi} is the i^{th} attribute value (i = 1, 2, . . .,m) of the p^{th} training pattern. Fuzzy rule of the classifier is of the following form:

Rule R_j: IF x_1 is A_{j1} and $\cdots$ and x_m is A_{jm}
THEN Class = C_j with RW_j

where, R_j is the label of the j^{th} rule, $x = (x_1, \ldots, x_m)$ is an m-dimensional pattern vector, A_{ji} is an antecedent fuzzy set, C_j is a class label, and RW_j is the rule weight.

The performance of fuzzy rule-based classifiers depends on the rule weight of each fuzzy rule R_j [14]. The most common rule weight is the fuzzy confidence value or certainty factor (CF) [15].

$$RW_j = CF_j = \frac{\sum_{x_p \in Class\ C_j} \mu_{A_j}(x_p)}{\sum_{p=1}^{N} \mu_{A_j}(x_p)} \quad (1)$$

where, $\mu_{A_j}(x_p)$ is the matching degree of the pattern x_p with the antecedent part of the fuzzy rule R_j. Fuzzy reasoning method of the weighted vote or additive combination is used to classify new patterns by the rule base [16]. With this method, each fuzzy rule casts a vote for its consequent class. The total strength of the vote for each class is computed as follows:

$$V_{Class_h}(x_p) = \sum_{R_j \in RB; C_j = h} \mu_{A_j}(x_p).\, CF_j \quad (2)$$

The new pattern x_p is classified as the class with the maximum total strength of the vote. If multiple class labels have the same maximum value for x_p or no fuzzy rule is compatible with x_p, this pattern is classified as the class with most patterns in the training data.

B. Fuzzy Association Rules for Classification

A fuzzy association rule can be considered to be a classification rule if the antecedent contains fuzzy item sets, and the consequent part contains only one class label (C = $\{C_1, \ldots, C_j, \ldots, C_S\}$). A fuzzy associative classification rule, i.e., $A_j \Rightarrow C_j$, can be measured directly in terms of support and confidence [17]. Support measures the reliability by the relative frequency of co-occurrences of the rule's item. Confidence measures the rule accuracy.

$$\text{Support}(A_j \Rightarrow C_j) = \frac{\sum_{x_p \in \text{Class } C_j} \mu_{A_j}(x_p)}{N} \quad (3)$$

$$\text{Confidence}(A_j \Rightarrow C_j) = \frac{\sum_{x_p \in \text{Class } C_j} \mu_{A_j}(x_p)}{\sum_{x_p \in T} \mu_{A_j}(x_p)} \quad (4)$$

where, T is the dataset.

III. Proposed IDS

This section describes the proposed method of intrusion detection. This method is based on the following three stages:

1) *Listing of all frequent fuzzy item sets:* A search tree is employed to list all the possible frequent fuzzy item sets and to generate fuzzy association rules for classification.
2) *Selection of candidate fuzzy association rules:* A rule evaluation criterion is used to preselect candidate fuzzy association rules.
3) *Genetic rule selection and lateral tuning:* The best cooperative rules are selected and tuned by means of a GA, considering the positive synergy between both techniques within the same process.

A. Listing of all Frequent Fuzzy Item Sets

A search tree is employed to list all the possible fuzzy item sets of a class for rule base generation. The root or level 0 of the search tree is an empty set. The order of attributes (features of network connection) is according to their appearance in the training data, and the one-item sets that correspond to the attributes are listed in the first level of the search tree according to their order. If an attribute has j possible outcomes (j linguistic terms for each quantitative attribute), it will have j one-item sets that are listed in the first level. The children of a one-item node for an attribute are the two-item sets that include the one-item set of that attribute and a one-item set for another attribute behind that attribute in the order, and so on. If an attribute has j > 2 possible outcomes, it can be replaced by j binary variables to ensure that no more than one of these j binary attributes can appear in the same node in a search tree. In this method five fuzzy linguistic terms Low, Low Medium, Medium, Medium High and High, are used for quantitative attributes. An example with two attributes V_1 and V_2 with two linguistic terms L (Low) and H (High) is depicted in Fig. 1.

An item set with a support higher than the minimum support is a frequent item set. If the support of an n-item set in a node J is less than the minimum support, it does not need to be extended more because the support of any item set in a node in the sub tree, which is led by node J, will also be less than the minimum support. Likewise, if a candidate item set generates a classification rule with confidence higher than the maximum confidence, this rule has reached the quality level that is demanded by the user, and it is again unnecessary to extend it further. The maximum confidence value is taken 0.8. These properties greatly reduce the number of nodes needed for searching. The fuzzy support of an item set can be calculated as:

$$\text{Support(A)} = \frac{\sum_{x_p \in T} \mu_A(x_p)}{N} \quad (5)$$

where, $\mu_A(x_p)$ is the matching degree of the pattern (connection in a network dataset)x_p with the item set, T is the training network dataset and N is the number of connections in T. The matching degree $\mu_A(x_p)$ of x_p to the different fuzzy regions is computed by the use of a conjunction operator, in this case, the product T-norm.

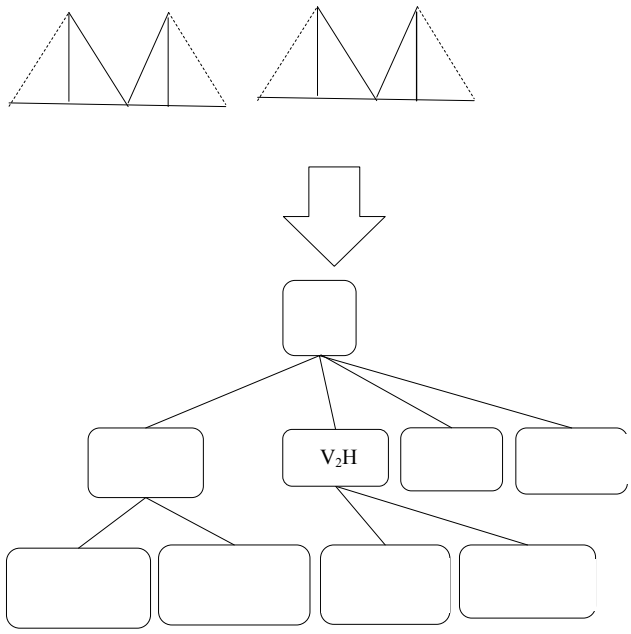


Fig. 1 Search tree for two quantitative attributes V_1 and V_2 with two linguistic terms L and H

Once all frequent fuzzy item sets have been obtained, the candidate fuzzy association rules for classification can be generated, setting the frequent fuzzy item sets in the antecedent of the rules and the corresponding class in the consequent. This process is repeated for each class.

The number of frequent fuzzy item sets that are extracted, depends directly on the minimum support. This algorithm determines the minimum support of each class (normal and attack types) by the distributions of the classes over the dataset. Thus, the minimum support for class C_j is defined as:

$$\text{MinimumSupport}C_j = \text{minSup} * fC_j \quad (6)$$

where, minSup is the minimum support, taken 0.05, and fC_j is the pattern ratio of the class C_j .

In this stage, a large number of candidate fuzzy association rules are generated for classification. To generate short fuzzy rules with only a small number of antecedent conditions, the depth of the trees is limited to a fixed value that is 3 in our approach.

B. Selection of Candidate Fuzzy Association Rules

To reduce the computational costs of next stage, subgroup discovery is used to preselect the most interesting rules from the RB, which are obtained in the previous stage, by means of a pattern weighting scheme. In this scheme the patterns are treated in such a way that covered positive patterns are not deleted when the current best rule is selected. Instead, each time a rule is selected, the algorithm stores a count i for each pattern of how many times (with how many of the selected rules) the pattern has been covered.

By using the formula $w(e_j,i) = 1/i+1$, weights of positive patterns covered by the selected rule is decreased . In the first iteration, all target class patterns are assigned the same weight, i.e., $w(e_j,0)=1$, while in the following iterations the contributions of patterns are inversely proportional to their coverage by previously selected rules. This way, the patterns that are already covered by one or more selected rules decrease their weights while uncovered target class patterns whose weights have not been decreased will have a greater chance of being covered in the following iterations. Covered patterns are completely eliminated when they have been covered more than two times.

Thus, in each iteration of the process, the rules are ordered according to a rule evaluation criterion from best to worst. The best rule is selected, covered patterns are reweighted, and the procedure repeats these steps until one of the stopping criteria is satisfied: either all patterns have been covered more than two times, or there are no more rules in the RB. This process is to be repeated for each class.

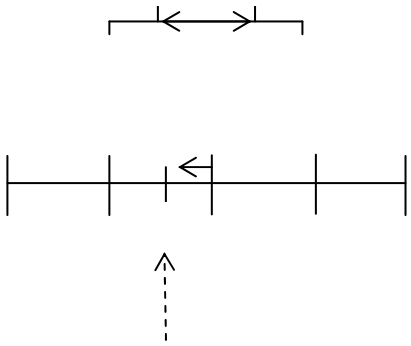
Fuzzy rules are handled using the weighted relative accuracy (WRAcc) measure, which is a modified measure used to evaluate the quality intervalar rules in APRIORI-SD [18]. The modified measure is as follows:

$$\text{WRAcc}(A \Rightarrow C_j) = \frac{n''(A \cdot C_j)}{n'(C_j)} \cdot \left(\frac{n''(A \cdot C_j)}{n''(A)} - \frac{n(C_j)}{N} \right) \quad (7)$$

where, $n(C_j)$ is the number of patterns of class C_j , N is the number of all patterns, n'' (A) is the sum of the products of the weights of all covered patterns by their matching degrees with the antecedent part of the rule, $n''(A \cdot C_j)$ is the sum of the products of the weights of all correctly covered patterns by their matching degrees with the antecedent part of the rules, and n' (C_j) is the sum of the weights of patterns of class C_j.

C. Genetic Rule Selection and Lateral Tuning

An excess number of rules may not produce good performance and it makes difficult to understand the model behaviour. To select and tune a compact set of fuzzy association rules with high classification accuracy from the rule base, a GA model is used, where rules are based on the linguistic two tuple representation [19]. The symbolic translation parameter of a linguistic term is a number within the interval [−0.5, 0.5) that expresses the domain of a label when it is moving between its two lateral labels. If S is set of labels representing a fuzzy partition, then there is a pair (S_i, α_i), $S_i \in S$, $\alpha_i \in [-0.5, 0.5)$. Figure 2 shows the symbolic translation of a linguistic term.



F c translation of a linguistic term

A specific GA, the CHC algorithm [20] is used. The CHC algorithm is a GA that presents a good trade-off between exploration and exploitation, making it a good choice for problems with complex search spaces. This genetic model makes use of a mechanism of selection of populations in order to perform an adequate global search. The CHC approach makes use of an incest prevention mechanism and a restarting process to encourage diversity in the population, instead of the well-known mutation operator. This incest prevention mechanism will be considered in order to apply the crossover operator, i.e., two parents are crossed if their hamming distance divided by 2 is more than a predetermined threshold L. This threshold value is initialized as the maximum possible distance between two individuals (the number of non-matching genes in the chromosome) divided by 4. Following the original CHC scheme, L is decremented by 1 when there are no new individuals in the population in one generation. In order to make this procedure independent of the number of genes in the chromosome, in this case, L will be decremented by φ% of its initial value (where φ determined by the user, usually 10%). When L is below zero, the algorithm restarts the population.

Scheme of this GA is as follows:

1) *Codification and initial gene pool:* To combine the rule selection with the global lateral tuning, a double coding scheme for both rule selection C_S and lateral tuning C_T is used. For the C_S part, each chromosome is a binary vector that determines when a rule is selected or not (alleles '1' and '0' respectively). Considering the M rules that are contained in the candidate rule set, the corresponding part, i.e., $C_S = \{c_1, .. . , c_M\}$, represents a subset of rules composing the final RB so that IF $c_i = 1$ THEN ($R_i \in$ RB) *else* ($R_i \notin$ RB), with R_i being the corresponding i^{th} rule in the candidate rule set and RB being the final RB. For the C_T part, a real coding is considered. This part is the joint of the α parameters of each fuzzy partition. If the no. of labels per variable is $(m_1, m_2, . . . , m_n)$ with n being the number of system variables, then, this part has the following form, where each gene is associated with the tuning value of the corresponding label: $C_T = (c_{11}, . . . , c_{1m1} , c_{21}, . . . , c_{2m2} , . . . , c_{n1}, . . . , c_{nmn})$. Finally, a chromosome C is coded in the following way: $C = C_S C_T$. To make use of the available information, all the candidate rules are included in the population as an initial solution. To do this, the initial pool is obtained with the first individual having all genes with value '1' in the C_S part and all genes with value '0.0' in the C_T part. The remaining individuals are generated at random.

2) *Chromosome evaluation:* To evaluate a determined chromosome penalizing a large number of rules, classification rate is computed and the fitness function is maximized. This function must be in the accordance with the framework of imbalanced datasets. Therefore the average of sum of correctly classified training patterns by the rules in the chromosome part C_S is used as fitness function.

$$\text{Fitness}(C) = \frac{\sum_{i=1}^{N_{rs}} NCP(R_i)}{N_{rs}} \quad (8)$$

where, N_{rs} is the number of rules in the rule set and $NCP(R_i)$ is number of correctly classified training patterns. If there is at least one class without selected rules or if there are no covered patterns, the fitness value of a chromosome will be penalized with the number of classes without selected rules and the number of uncovered patterns.

3) *Crossover operator:* The crossover operator will depend on the chromosome part where it is applied. In the C_S part, the half-uniform crossover scheme (HUX) is employed. The HUX crossover exactly interchanges the mid of the alleles that are different in the parents (the genes to be crossed are randomly selected from among those that are different in the parents). This operator ensures the maximum distance of the offspring to their parents (exploration). For the C_T part, the Parent Centric BLX (PCBLX) operator (an operator that is based on BLX-α) is considered. This operator is based on the concept of neighbourhood, which allows the offspring genes to be around the genes of one parent or around a wide zone that is determined by both parent genes. After crossover operation, four offspring are generated by the combination of the two from the part C_T with the two from the part C_S. The two best offspring obtained in this way are considered as the two corresponding descendants.

4) *Restarting approach*: To get away from local optima, a restarting approach has been used. In this case, the best chromosome is maintained, and the remaining are generated at random. The restart procedure is applied when the threshold value L is below zero, which means that all the individuals coexisting in the population are very similar.

IV. EXPERIMENTATION

Experiments were carried out on a subset of KDD-cup 99 dataset and the results were compared with some well-known IDS classifiers.

A. KDD-cup 99 Dataset

KDD-cup 99 dataset is made up of a large number of network connections related to normal and malicious traffic. KDD-cup 99 dataset is a version of the 1998 DARPA Intrusion Detection Evaluation Program, prepared and managed by MIT Lincoln Labs [21]. The 10% of KDD-cup 99 dataset consists of 494021 single connection vectors. Each connection vector in KDD-cup 99 dataset has 41 features and is labelled as either normal or an attack, with exactly one specific attack type. The 10% of the dataset contain a total number of 22 attack types as shown in table I.

TABLE I
INTRUSION CLASSES AND ATTACK TYPES

Sl. No.	Intrusion Class	Attack Types
1	Denial of Service	back, land, neptune, pod, smurf, teardrop
2	Probe	ipsweep, nmap, portsweep, satan
3	Remote to Local	ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster
4	User to Root	buffer_overflow, loadmodule, multihop, perl, rootkit

Since, the number of records in the 10% data set is very large (494,021) therefore; a subset of this large dataset has been used as train and test datasets. We have randomly selected 100 normal, 91 DoS, 80 Probe, 80 R2L and 59 U2R records, total of 410 records. After selecting them, we randomized their orders of records. Training and test sets were created using 10-fold cross validation. The performance of IDS classifier is calculated as the average of these ten sets.

B. Evaluation Metric

The performance of the classifier is measured in the terms of precision, recall, F-measure and accuracy. They can be calculated using the confusion matrix [22]. A confusion matrix contains information about actual and predicted classification done by a classification system. Table II shows the confusion matrix for a two class classifier.

TABLE II
CONFUSION MATRIX FOR TWO CLASSES

		Predicted class	
		Positive	Negative
Actual Class	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Measures of performance defined as follows:

1) *Precision:* Precision is the proportion of the predicted positive cases that were correct.

$$\text{Precision} = TP/(TP+FP)$$

2) *Recall:* Recall is the proportion of the positive cases that were correctly identified.

$$\text{Recall} = TP/(TP+FN)$$

3) *F-measure:* The F-measure is the harmonic mean of recall and precision.

$$\text{F-measure} = 2*\text{Precision}*\text{Recall}/(\text{Precision}+\text{Recall})$$

4) *Accuracy:* Accuracy is the proportion of total number of predictions that were correct.

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

C. Results

Table III is the confusion matrix of the proposed IDS. The confusion matrix shows that 95% of the actual normal test data were detected to be normal; 90% of the actual probe test data were detected to be probe; 94.5% of DoS test data were detected to be DoS; 92.5% of the actual R2L test data were detected to be R2L and 86.4% of the actual U2R test data were detected to be U2R. Precision of the normal class is 87%, for probe 97.3%, for DoS 98.9%, for R2L 88% and for U2R is 92.7%. The overall accuracy of the IDS is 92.2%.

The result of IDS proposed in this work is compared with some well-known IDSs. Table IV shows the comparison. Although the Proposed IDS used smaller percent of the original intrusion dataset, the performance measures are comparable with other IDSs.

TABLE III
CONFUSION MATRIX OF PROPOSED IDS

	Probe	Normal	DoS	R2L	U2R	Recall%	F-measure%
Probe	72	6	0	2	0	90	93.5
Normal	1	95	1	3	0	95	90.82
DoS	1	4	86	0	0	94.5	96.6
R2L	0	2	0	74	4	92.5	89.2
U2R	0	3	0	5	51	86.4	89.4
Precision%	97.3	87	98.9	88	92.7		
Accuracy%	92.2						

TABLE IV
COMPARISON WITH OTHER WELL-KNOWN CLASSIFIERS

	Metric	Proposed IDS	MOGFIDS [9]	KDD-cup 99 winner [23]	EFRID [8]	RIPPER [9]
Normal	Recall	95	98.36	99.5	92.78	
	Precision	87	74.73	74.61		
	F-measure	90.82	84.93	85.28		
Probe	Recall	90	88.59	83.3	50.35	81.16
	Precision	97.3	74.4	64.81		77.92
	F-measure	93.5	80.88	72.9		79.51
DoS	Recall	94.5	97.2	97.1	98.91	22.06
	Precision	98.9	99.89	99.88		95.75
	F-measure	96.6	98.53	98.47		35.86
R2L	Recall	92.5	11.01	8.4	7.41	8.33
	Precision	88	68.93	98.84		81.85
	F-measure	89.2	18.97	15.48		15.2
U2R	Recall	86.4	15.78	13.2	88.13	11.84
	Precision	92.7	61.01	71.43		55.10
	F-measure	89.4	25.08	22.28		19.49

The recall value of normal class of our IDS is 95%, which is better than EFRID (92.78%) but less than other classifiers. The less value of recall for normal class is due to the other classifiers were tested with dataset having large no. of normal connections. Precision (87%) and F-measure (90.82) value for normal class is better than other classifiers. For probe class, all three performance measures (recall 90.25%, precision 97.3% and F-measure 93.5%) are highest among

these IDS classifiers. For DoS class Recall and F-measure values (94.5%, 96.6% respectively) of our IDS are better than RIPPER (22.06%, 35.86% respectively). Precision value for DoS class (98.9%) is better than RIPPER but less than the other classifiers. Again, the less values of measures for DoS class, is due to the other classifiers are tested with dataset having large no. of DoS connections. For R2L recall and F-measure values (82.5%, 89.2%) are highest among all, and precision value is only less than the KDD-cup 99 winner. For U2R all three measures (recall 86.4%, precision 92.7% and F-measure 89.4%) are highest among all.

From this comparison, we see that the result of the classification of our IDS is comparable and better in some cases. This is due to the fitness function used in the rule selection and lateral tuning stage in the algorithm. So, this classifier is suited for intrusion detection problem.

V. CONCLUSION

The result of the experiment demonstrated that fuzzy-genetic based classification is an effective approach for intrusion detection. Comparison with other IDS classifiers has shown us that the proposed classification based IDS has potential to detect normal connection and attack types with good accuracy. The effectiveness in detection is due to the fitness function introduced for this intrusion detection model. Therefore, we can say that this classification model is effective in intrusion detection system.

REFERENCES

[1] J. P. Anderson, "Computer security threat monitoring and surveillance," James P. Anderson Co., Fort Washington, Pennsylvania, Technical Report, 1980.

[2] Y. Zhang, X. Wu, Z. Xing, and W. Hu, "On generating interpretable and precise fuzzy systems based on Pareto multi-objective cooperative co- evolutionary algorithm," *Appl. Soft Computing*, vol. 11, pp. 1284– 1294, 2011.

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed., San Fransisco, CA: Morgan Kaufmann, 2006.

[4] R. Rak and L. K. M. Reformat, "A tree-projection-based algorithm for multi-label recurrent-item associative-classification rule generation," *Data Knowledge Engineering*, vol. 64, no. 1, pp. 171–197, 2008.

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

[6] W. Lee, "A data mining framework for building intrusion detection models," in *IEEE Symposium on Security and Privacy*, Berkeley, California, 1999, pp. 120–132.

[7] S. M. Bridges and R. B. Vaughn, "Fuzzy data mining and genetic algorithms applied to intrusion detection," in *proc. National Information Systems Security Conference (NISSC)*, Baltimore, MD, 2000, pp.16-19.

[8] J. Gomez and D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection," in *proc. IEEE Workshop on Information Assurance*, United States Military Academy, West Point, NY, 2002, pp. 68-75.

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

[10] J. Jabez and Dr. G. S. A. Mala, "A study on genetic-fuzzy based automatic intrusion detection on network datasets," in *proc. International Conference on Software Engineering and Mobile Application Modelling and Development*, Chennai, India, 2012, pp. 1-8.

[11] Z. Lei, M. Lingrui, and H. Chunjie, "Intrusion detection based on immune principles and fuzzy association rules," *Intelligence Computing and Evolutionary Computation Advances in Intelligent Systems and Computing*, vol. 180, pp. 31-35, Springer, 2013.

[12] K. Chadha and S. Jain, "Hybrid genetic fuzzy rule based inference engine to detect intrusion in networks," *Intelligent Distributed Computing Advances in Intelligent System and Computing*, vol. 321, pp. 185-198, Springer, 2015.

[13] KDD-cup data set:http://kdd.ics.uci.edu/databases/kddcup99/kddcup99

[14] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 506–515, Aug. 2001.

[15] O. Cordon, M. Del Jesus, and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning*, vol. 20, no. 1, pp. 21–45, 1999.

[16] H. Ishibuchi and T. Yamamoto "Rule weight specification in fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 428-435, 2005.

[17] D. Dubois, E. Hullermeier, and H. Prade, "A systematic approach to the assessment of fuzzy association rules," *Data Mining and Knowledge Discovery*, vol. 13, no. 2, pp. 167–192, 2006.

[18] B. Kavsek and N. Lavrac, "APRIORI-SD: Adapting association rule learning to subgroup discovery," *Application of Artificial Intelligence*, vol. 20, no. 7, pp. 543–583, 2006.

[19] F. Herrera and L. Martinez, "A 2-tuple fuzzy linguistic representation model for computing with words," *IEEE Transactions on Fuzzy Systems.*, vol. 8, pp. 746–752, 2000.

[20] L. Eshelman, "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *proc. Foundations of Genetic Algorithms*, vol. 1, G. Rawlin, Ed. San Mateo, CA: Morgan kaufmann, 1991, pp. 265–283.

[21] Lincoln Laboratory MIT: http://www.ll.mit.edu

[22] R. Kohavi and F. Provost, "Glossary of terms," *Machine Learning*, vol. 30, pp. 271-274, 1998.

[23] C. Elkan, "results of the KDD'99 classifier learning, *ACM SIGKDD Int. conf. Knowledge Discovery and Data Mining,*" Boston, 2000, pp. 63-64.