

ENTROPY BASED FUZZY RULE WEIGHTING FOR HIERARCHICAL INTRUSION DETECTION

M. R. MOOSAVI, M. FAZAELI JAVAN,
M. H. SADREDDINI AND M. ZOLGHADRI JAHROMI

ABSTRACT. Predicting different behaviors in computer networks is the subject of many data mining researches. Providing a balanced Intrusion Detection System (IDS) that directly addresses the trade-off between the ability to detect new attack types and providing low false detection rate is a fundamental challenge. Many of the proposed methods perform well in one of the two aspects, and concentrate on a subset of system requirements. There are many non-functional requirements for an applicable and practical IDS. The process should be online, incremental and adaptive to ever changing behaviors of normal users and attackers. Moreover providing comprehensive and interactive IDS could both, enhance the performance of the system and extend the knowledge of domain experts.

In this paper, we propose a fuzzy rule-based classification system using a hierarchical rule learning method. In each stage of the hierarchy, a set of rules with certain length of antecedent are investigated. A novel rule weighting method, based on the entropy measure, determines the appropriateness of each rule. The experimental results on KDD99 intrusion detection dataset show the effectiveness of the proposed method in tackling the tradeoff between accuracy and comprehensibility of fuzzy rule-based systems. Although the dimension of antecedents is not limited, the resultant rule-base contains a small number of complex rules, which are essential to reach the desired accuracy.

1. Introduction

Intrusion Detection Systems (IDS) are effective security tools that look for known or potential threats in network traffic and/or audit data recorded by hosts [16]. Basically, an IDS analyzes information about users behaviors from various sources such as audit trail, system table, and network usage data [3]. The problem of intrusion detection has been studied extensively in computer security, and has received a lot of attention in machine learning and data mining [7, 32].

There are two major paradigms for intrusion detection: misuse detection and anomaly detection [8, 43]. The former method, also called signature based detection, is based on known patterns for malicious activities. This method can establish a rule-base, by analysing the signature of the known attack types [5, 39]. The latter, identifies novel attacks that deviate from established statistical patterns of users,

Received: August 2012; Revised: February 2013; Accepted: February 2014

Key words and phrases: Intrusion detection, Hierarchical classification, Iterative fuzzy rule-based system, Rule weighting, Entropy measure, KDD99.

systems or networks. In spite of their capability to detect unknown attacks, systems designed for anomaly detection, usually generate high volumes of false alarms (i.e., matching normal traffic events with attack signatures, or permitting malicious connections) [47]. Therefore, the tradeoff between the ability to detect new attacks and the ability to generate a low rate of false alarms is the key point to develop an effective IDS.

According to the latest research literature, many different classification techniques have been applied to the intrusion detection problem [38]. Using data mining approaches, the patterns of network users activities could be extracted efficiently. Various techniques have been applied to discover useful knowledge that describes the users' behaviours from large audit data sets. Artificial neural networks, inductive and associative rule-based systems, genetic algorithms, clustering and outlier detection schemes are among widely used techniques for anomaly and misuse detections [41, 33, 20, 44]. Many of these works are proposed only to optimize the classification accuracy (or overall classification cost) and omits the necessity of interpretability optimization [57].

In order to increase the intrusion detection rate, a multiple-level tree classifier was proposed in [59] which contains three-levels of decision tree classification. A serious shortcoming of this approach, and its further improvements [60], is its high false alarm rate as well as low detection rate for unknown attacks.

To tackle high false alarm rate, some of the researches suggested to combine different techniques in hybrid systems [60]. The KDD cup 99 winner fused 5×10 C5 decision trees using cost-sensitive bagged boosting algorithm [46]. This method has a low false alarm rate but does not perform well in detecting new attack types. To deal with this problem, Pan et al. [42] proposed a misuse detection method incorporating different classification abilities of neural networks and the C4.5 decision trees algorithm for different attack types. Another hybrid system is proposed by Hwang [24] in which, he tried to combine the advantages of signature based and anomaly detection systems (i.e., low false positive rate and detecting unknown attacks).

Each of the above-mentioned works increased the accuracy of the system but provided a more complex model. In fact, the complexity of the model is a common drawback for most of the proposed methods. A complex model could not be used along with domain expert knowledge, which is a major disadvantage in the field of intrusion detection.

Fuzzy systems based on if-then rules have been successfully used in many applications areas. Particularly, in the area of network security, fuzzy logic techniques have been used since 1990's [44, 23]. Precision and complexity are inversely related. Fuzzy logic can deal with imprecision and vagueness. Therefore, it is appropriate for the development of security systems, since many security elements are fuzzy [34, 50]. Patcha stated that several quantitative parameters that are used in the context of intrusion detection can potentially be viewed as fuzzy variables [44]. Bridges argued that the security concept is fuzzy: the concept of fuzziness helps to smooth out the abrupt separation of normal behaviors from abnormal behaviours [6].

Many fuzzy rule based classifiers are proposed for intrusion detection. While usual rule based techniques fail in the case of the KDD 99, major rule based IDS, namely UCS [51], XCS [13] and XCSR [4], use genetic algorithms for introducing new rules into the population or tuning rule weights, which is too much time consuming and result in a huge rule base. Another state of the art fuzzy rule-based system is MOGFIDS that is evolved from an agent based evolutionary framework and can act as a genetic feature selection wrapper [57].

Ozyer et. al. proposed a method based on iterative rule learning using a fuzzy rule-based genetic classifier [41]. Their approach is mainly composed of two phases. First, a large number of candidate fuzzy rules (having at most three itemsets as the length of antecedent) are generated for each class. A genetic algorithm will try to extract one individual (i.e., rule) for one label iteratively. The antecedent of each rule is coded as a chromosome and a function of the confidence is used as the fitness function. During the next stage, boosting mechanism evaluates the weight of each data item to help the rule extraction mechanism focus more on data having relatively more weight, i.e., uncovered less by the rules extracted until the current iteration. The idea behind using the boosting mechanism is to aggregate multiple hypotheses generated by the same learning algorithm invoked over different distributions of the training data into a single composite classifier.

In [56] a neuro-fuzzy classifier proposed. Different ANFIS networks are used for different intrusion classes. They have also used subtractive clustering to determine the number of rules and initial locations for membership functions. At last a genetic algorithm is used to optimize the system. Obviously, tuning the membership functions and using a complex decision making engine decrease the interpretability of the system.

Many learning algorithms use fuzzy model to represent the knowledge obtained. SLAVE is a GA based learning algorithm to extract a set of fuzzy rules from a set of examples [17]. This process is developed through an iterative approach in which a rule is selected each time. The authors further discussed the idea of iterative generation and selection of rules in [18]. In order to obtain new and different rules, the rule previously obtained is penalized by eliminating the examples covered by this rule. This iterative scheme is repeated until the set of rules obtained adequately represents the examples in the training set, returning the set of rules as the solution to the problem.

In this paper we propose a fuzzy rule-based classification system to tackle the tradeoff between accuracy and comprehensibility of intrusion detection systems. This method is a relatively fast approach to intrusion detection, in which fuzzy rules are utilized for learning monitored behaviours in a network. A hierarchical rule generation used in this work to induce desired sets of rules. In each stage of the hierarchy, a set of rules with certain length of antecedent are investigated. A simple method is used to reduce total number of rules involved in each stage of rule generation without resulting in any information loss. Afterwards, a novel rule weighting scheme is incorporated to adjust decision boundaries resulting in an appropriate number of high accurate rules. At the end of each stage, we eliminate correctly classified training instances covered by the selected rules.

The rest of this paper is organized as follows. In section 2, fuzzy rule-based classification systems are described to introduce the notation. The proposed algorithm is applied to KDD99 intrusion detection dataset. The dataset is explained in section 3. Entropy based rule weighting is elaborated in section 4. Section 5, discusses the framework of hierarchical rule generation and selection, called Entropy Based Hierarchical Fuzzy Rule-Based System (EBHFRBS). In section 6, the experimental results are presented, and finally conclusions are remarked in section 7.

2. Fuzzy Rule-based Classification Systems

2.1. Fuzzy Logic. A fuzzy rule-based classification system is composed of different conceptual components [29]. Each rule in the rule-base specifies a subspace of pattern space using the fuzzy sets in the antecedent part of the rule. Different rule types have been used for pattern classification problems [11, 62].

Assume that fuzzy linguistic values are defined in the set $A = \{A_{fi} | 1 \leq f \leq d, 0 \leq i < l_f\}$ where d denotes the number of dimensions or features, l_f is the number of fuzzy sets for the feature f and A_{f0} is the don't care value (i.e. a rectangular membership function that covers the entire domain of each feature and can be used as the antecedent fuzzy set corresponding to the "don't care" condition [25]).

For a c -class d -dimensional problem, we use the following notation for each rule R_r ($1 \leq r \leq m$) in the rule-base of size m :

$$R_r : (x_1, a_{r1}) \wedge \dots \wedge (x_n, a_{rd}) \xrightarrow{CF_r} k_r \quad (1)$$

where $X = [x_1, x_2, \dots, x_d]^T$ is the input feature vector, $1 \leq k_r \leq c$ is the consequent class label, CF_r is the certainty grade of the fuzzy rule and the antecedent $a_{rf} \in A$ is one of the linguistic values of the feature f .

In order to classify a normalized feature vector $X_p = [x_{p1}, x_{p2}, \dots, x_{pd}]^T$, the degree of compatibility of X_p with each rule is calculated (i.e., using a T -norm to model the *and* connectives in the rule antecedent). In the case of using product as T -norm, the compatibility degree of rule R_r with X_p can be calculated as

$$\mu_r(X_p) = \prod_{i=1}^d \mu_{r_i}(x_{pi}) \quad (2)$$

where $\mu_{r_i}(\cdot)$ is the membership function of the antecedent fuzzy set a_{ri} .

In the case of using single winner reasoning method, the pattern is classified according to consequent class of winner rule R_w . With the rules of the form (1), the certainty grade of each rule is also used in finding the winner rule:

$$w = \arg \max_{i \in \{1, 2, \dots, m\}} \{\mu_i(X_p) \cdot CF_i\} \quad (3)$$

Fuzzy rule-based system comprehensibility has been taken into account. In our work it concerns:

- 1- Linguistic interpretability of fuzzy sets: Usually fuzzy sets represent linguistic meanings. While only a forward learning procedure is used to assign consequent part of the rules and the rule weights, no tuning occurs on simple homogeneous fuzzy sets.

- 2- Simplicity of the fuzzy rule-base: A very limited number of rules are selected from candidate rule-pool for final rule-base. In our five class problem, the rule-base includes a few hundreds of rules.
- 3- Complexity of fuzzy rules: While it is easy for human users to understand short rules with only a few number of antecedent conditions, minimizing rule length is considered. In our method, rules with antecedent length of more than three are rarely generated.
- 4- Simplicity of reasoning: The single winner method has tangible benefit of interpretability.

2.2. Space Partitioning. Fuzzy rule generation methods can be categorized into two approaches according to their strategies for dividing the input space into fuzzy subspaces: multidimensional antecedent fuzzy sets and grid-type fuzzy partitioning [26]. In grid-type fuzzy partitioning, the antecedent part of each fuzzy rule is a combination of linguistic values, which results in an interpretable fuzzy rule-based system.

When attribute i has L_i linguistic values (including don't care), the total number of possible combinations (i.e. total number of rules) is $\prod_{i=1}^d l_i - 1$ for a d -dimensional problem. Thus it is impracticable to test all combinations [27]. Usually the domain interval of all features are discretized into equal number of fuzzy sets (i.e. $\forall i : l_i = l$). The number of fuzzy if-then rules of maximum antecedent length q is calculated as:

$$\sum_{i=1}^q {}_d C_i \times l^i \quad (4)$$

where ${}_d C_i$ is the number of i -combinations from d elements. This equation implies that investigating the total number of rules with small number of antecedent conditions is feasible even with large number of features and fuzzy sets.

Usually it is assumed that the set of linguistic values is provided by the domain experts. This means that although the antecedent tuning may enhance the overall performance, but from the comprehensibility aspect of view, it is not desirable. The modification of antecedent fuzzy sets is likely to cause a gap between resulted membership functions and the understanding of linguistic values.

For continuous features, we have used 6 fuzzy partitions for each dimension of input vector as illustrated in Figure 1(a). For each symbolic feature, like protocol types, services and flags, we have defined P fuzzy singleton, where, P is the number of values that the feature can assume. For example, *protocol type* feature with three different symbols TCP, UDP and ICMP is shown in Figure1(b).

According to the normalization method discussed in experiments, for all continuous features the majority of values are mapped to a small interval around 0.5. As shown in Figure 2, some features such as 9 have similar values for most of the patterns which is mapped to 0.5. Therefore a fuzzy set with membership degree of 1.0 in this point will dominant in the reasoning procedure. On the other hand some features, such as 6, although spanned over a very large integer range, have few great values and their value for most of the patterns is mapped to 0.0.

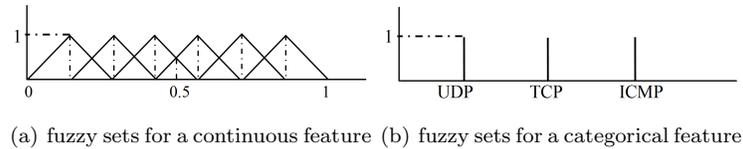


FIGURE 1. Input Vector Fuzzy Partitioning

3. KDD 99 Dataset

The KDD Cup 99 dataset has been widely used to evaluate performance of intrusion detection systems. The KDD99 intrusion detection dataset is based on the 1998 DARPA initiative, which provides designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies [22, 14, 36]. Although a noteworthy classifier should have enough flexibility to be well customized for datasets with various characteristics, but there are many clues about KDD99 intrusion detection dataset that should be very early considered in classifier system construction, such as fuzzy set partitioning just described in previous section.

The data set has 41 attributes for each connection record and a label indicating the status of the records as either normal or a specific attack type. These features had different forms of continuous, discrete, and symbolic, with significantly varying ranges and class various separability. Some features are derived features, which are useful in distinguishing normal connection from attacks. There are four groups of features: Basic Features, Content Features, Time-based Traffic Features, Host-based Traffic Features [30].

In the International Knowledge Discovery and Data Mining Competition [31], a subset (10%) of originally recorded data was prepared for the purpose of training the classifiers. A test subset was also prepared to evaluate different classification methods. Some statistics of this dataset is given in Table 1. The training set contains a total of 22 training attack types. Also there are 17 additional types in the testing set only. Each attack falls into one of the following categories:

- Denial of Service (DOS): Attacker tries to prevent legitimate users from using a service.
- Remote to Local (R2L): Attacker tries to gain access to the victim machine without having an account.
- User to Root (U2R): Attacker has local access to the victim machine and tries to gain super user privileges.
- Probe: Attacker tries to gain information about the target host.

The signatures of *DOS* and *Probe* attacks in the test subset provided by *KDD99* are very similar to those present in the training set. However, the types of U2R and R2L attacks differ significantly between the training and testing data sets [10]. In the testing set, over 80% U2R attacks and 60% R2L attacks are new to the training set. The lack of correlation makes these two attacks harder to be identified. Literature survey indicates that many intrusion detection systems have very low detection rates in identifying U2R and R2L attacks [37, 49].

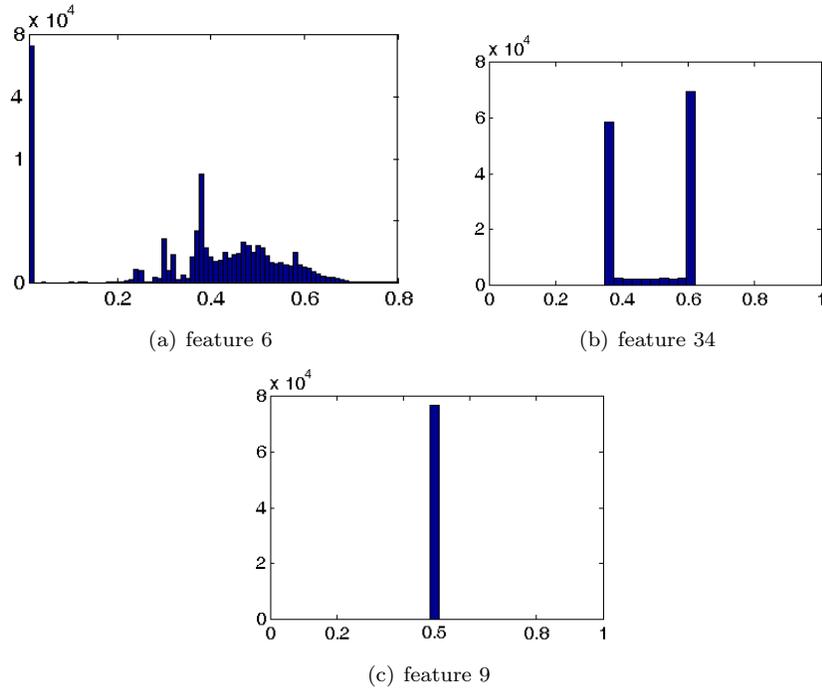


FIGURE 2. Distribution of Feature Values in the Training Set

4. Rule Weighting Using Entropy Measure

4.1. Entropy Measure. Entropy measure is one of the most commonly used discretization measures in the preprocessing phase. Entropy based discretization is a supervised, top-down splitting technique. It explores class distribution information in its calculation and determination of split-points. Using this measure, a numerical value could be determined as the split point for which the entropy of the two resultant intervals is minimized. The procedure could be repeated to arrive to a hierarchical discretization.

Dataset	Normal	DOS	U2R	R2L	Probe	Total
Original train set (10 % KDD)	97277	391458	52	1126	4107	494020
the training set used in the experiments	2556	372	41	103	242	3314
Test set (Corrected KDD)	60593	229853	70	16347	4166	311029
Whole KDD	972780	3883370	52	1126	41102	4898430

TABLE 1. Some Statistics of the KDD 99 Dataset

The entropy and information gain measures are also used for attribute selection in decision tree induction. It is also known as the expected information needed to classify a tuple in data set. The selected attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or “impurity” in these partitions. Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple tree is found.

C4.5 is the well-known decision tree induction algorithm that uses entropy based criterion to select best attribute in each node of the tree [48, 54].

Subspace clustering is an extension to attribute subset selection that has shown its strength at high-dimensional clustering [2, 45]. It can be performed by an unsupervised process, such as entropy analysis, which is based on the property that entropy tends to be low for data that contain tight clusters [9]. Yao et al proposed an entropy based fuzzy clustering method. In their work, an entropy measure is defined for identifying the number of clusters and their centers [61].

In our proposed method, each fuzzy rule could be considered as a separate classifier that would provide a degree of belongingness for each query point to the class of its consequent. The information gain theory is applicable to determine the effective influence or the certainty factor of each rule. In [40] a similar approach is used to determine the weight of prototypes in a weighted Nearest Neighbor classifier.

The weight of a rule should be degraded if instances in its covering area belong to various classes. This way, the rule will just decide about a portion of instances and other rules will be responsible for classifying some of these instances. We would like this partitioning to result in exact classification of instances. The idea is to select the weight parameter that will minimize entropy of the instances inside the decision area of the rule.

The information gain measure proposed by Shannon in a pioneering work on information theory [53]. For each split point (between two adjacent samples) in a list of instances, L , the measure is defined as:

$$Info(L) = \frac{|L_1|}{|L|} \times entropy(L_1) + \frac{|L_2|}{|L|} \times entropy(L_2) \quad (5)$$

where L_1 is the first and L_2 is the second part of the list L .

Choosing the split point with minimum information gain, will result in minimum amount of expected information still required to correctly classify samples in each part of the list [21].

Given c classes, the entropy of a list L is defined as:

$$entropy(L) = - \sum_{l=1}^c p_l \log_2(p_l) \quad (6)$$

where p_l is the probability of class l in L , and could be defined as follows:

$$p_l(L) = \frac{|\eta_l(L)|}{|L|} \quad (7)$$

where $\eta_l(L) = \{(X_i, c_i) \in L | c_i = l\}$ is the set of all instances of class l in a list L and $1 \leq l \leq c$. Therefore p_l is simply determined by dividing the number of instances of class l in L by $|L|$.

The traditional qualitative description of entropy is that it refers to changes in the status quo of the system and could be interpreted as a measure of pattern disorder. So it could be used to prefer conditions that cover a large number of instances of a single class and few instances of other classes. Therefore the splitting criterion is determined so that the resulting lists are as pure as possible.

Some important points should be noticed about the information gain measure. First of all, we do not need to consider the entropy of the second part of the list

(i.e. L_{t2}). The classification of instances in second part (i.e., instances after the split point) will later be decided in investigation of other rules. These instances will probably fall into decision boundary of various rules and their effect will be considered in other localities. This means that the second term of the information gain measure could be omitted. Though, considering the entropy of L_{t1} is not adequate. This measure tends to prefer unbalanced split in which one partition is smaller than the other. Therefore the normalization factor of $|L_{t1}|/|L|$ is essential.

Based on the list L_{t1} , not only the consequent label but also the certainty grade of the rule should be determined. Obviously, one class is in the majority of the list, which could be used as the consequent label of the rule. To compute the certainty grade, this class should be considered in relation with the average compatibility of instances. The main clue is that the probability of observing an instance with a certain label (i.e., the confidence measure) could be incorporated in the weight factor. Here we simply define the confidence of a list L as:

$$confidence_l(L) = \frac{\sum_{X \in L, X \in Class l} \mu_r(X)}{\sum_{X \in L} \mu_r(X)} \quad (8)$$

where l is a class label.

4.2. Rule Learning Procedure. The rule learning procedure includes: selection of the consequent class for an antecedent combination (i.e., candidate rules, in term of confidence), determination of the certainty degree (based on the entropy measure in the decision area) and choosing a set of rules for the final rule-base (according to an evaluation heuristic).

First of all, a set of candidate rules is generated based on available training data and the algorithm starts with an initial solution: $\{CF_i = 1 | i = 1, 2, \dots, m\}$. The algorithm attempts to improve this solution by adjusting the weight of the rules, based on the entropy measure.

After rule generation, to tune a fuzzy rule R_r , for each training instance X , a dissimilarity measure, called score, is computed:

$$score_r(X) = \frac{1}{\mu_r(X) \times CF_r} \quad (9)$$

All training instances in the covering area are sorted in ascending order of their scores in a list denoted as L . Then, the best split point in the list is found in order to boost classification result. For each split point in the sorted list (i.e., between each two consecutive instances that splits the list into two lists L_1 and L_2), the entropy measure is calculated, and then, the split point which results in the minimum entropy value, is selected as the best split point.

The consequent class of the rule R_r , denoted as K_r , is determined based on this split point and the list L_1 . We use a common approach for identifying the K_r , which is expressed as:

$$k_r = \arg \max_{1 \leq i \leq c} \{Confidence_i(L_1)\} \quad (10)$$

Also, the rule's weight would be updated based on the score of the best split point and the confidence of the list L_1 . Assuming X_i and X_j as two consecutive

instances in the best split point (i.e., X_i as the last instance in L_1 and X_j as the first instance in L_2), the weight of rule R_r is:

$$CF_r = \frac{score_r(X_i) + score_r(X_j)}{2} \times confidence_B(L_1) \quad (11)$$

The rule pruning procedure is based on a rule evaluation measure. Among many heuristic rule evaluation measures in the literature [28], our measure is based on soft consistency degree [19] and the rule evaluation proposed in [29] and can be expressed as:

$$\Gamma_r = \sum_{X \in Class\ k_r} CF_r \cdot \mu_r(X) - 2 \times \sum_{X \notin Class\ k_r} CF_r \cdot \mu_r(X) \quad (12)$$

The idea is to obtain rules covering the maximum number of examples (completeness degree) with the minimum number of negative examples (consistency degree) [19]. It must be noted that the confidence (and consequently the certainty grade) equation is based on the patterns in L_1 and the incorporation of rules is not considered in the rule weighting step. But the evaluation criterion is based on classification of patterns in the decision area of each rule.

5. The Proposed Framework

Major fuzzy rule-based systems initially generate all simple rules, i.e. rules with certain and limited number of antecedents. The important advantage of these systems is their comprehensibility. To guarantee the interpretability of a fuzzy system, the inference should be based on simple rules as discussed in section 2.1.

Providing an acceptable performance for KDD IDS dataset, and also in many real world datasets, is not possible by means of rules with only a few attributes in the antecedents. On the other hand, considering high dimensional rules is tedious. The method proposed here, provides a chance of having limited number of rules with various antecedent lengths.

The proposed architecture, called EBHFRBS, starts with investigation of single antecedent rules. All rules that cover at least one training instance will be generated. Then the entropy based procedure will adjust rule weights and consequents. Afterwards, a rule pruning procedure will remove inappropriate rules. Finally, the training instances most compatible will remained rules, will be removed, and the retained subset of instances will be incorporated in the next stage in which rules with two antecedents are examined.

This way, the rule-base is built in an iterative approach. At each stage, the most compatible rules with the current distribution of retained instances are selected for the fuzzy rule-base. The block diagram of the proposed method is depicted in Figure 3.

Figure 4 shows the overall algorithm in pseudo-code. To describe this algorithm, assume that for a c -class problem, a set of training examples of the form $T = \{X_i | 1 \leq i \leq n\}$ is given, where $X_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T$ is a d -dimensional feature vector. The proposed algorithm starts with one antecedent rules (i.e., stage=1 in line 5 of the pseudo-code). In each stage, first of all, a set of candidate rules is generated. Then, for a number of iterations, the rules are investigated to determine

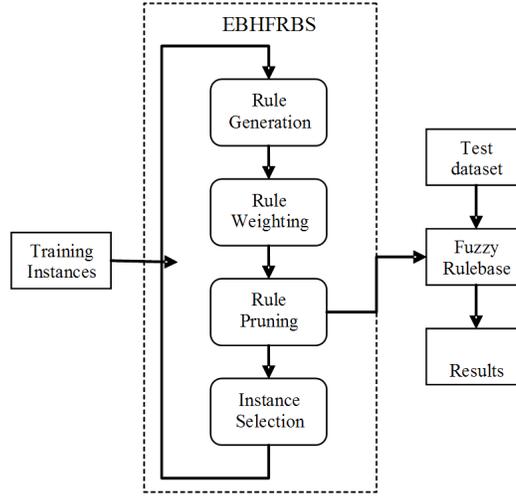


FIGURE 3. Context Diagram of the Proposed Method

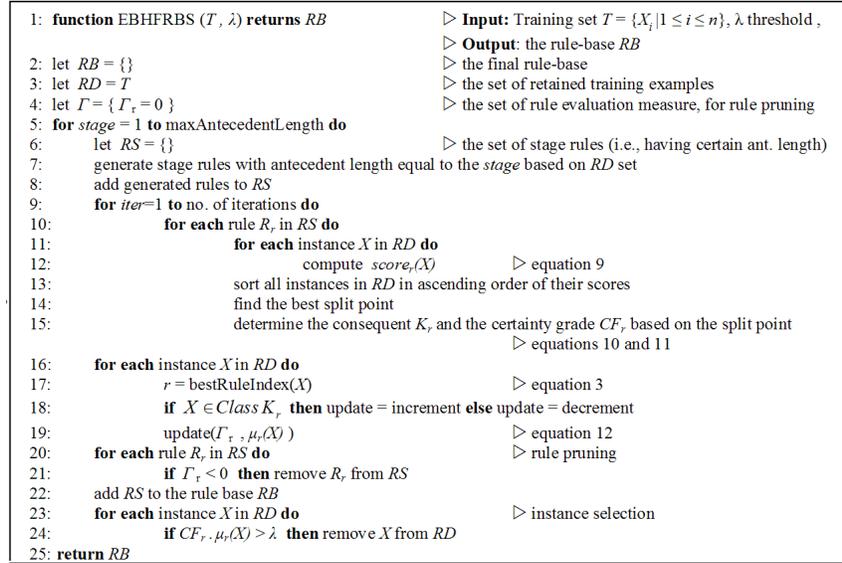


FIGURE 4. The Proposed Learning Algorithm

the consequent class and the certainty grade of stage rules (lines 9-15, based on equations (10) and (11)).

In lines 16-19, the rule evaluation measure (equation (12)) is computed and the stage rules with Γ_r less than zero are removed. Finally, in lines 23 and 24, most compatible training instances are removed from the training set.

In this learning framework, the Instance Selection module eliminates those examples from the training set that are λ covered by the last stage rules obtained in

the rule-base. Here we extend the Concept of λ Covering [19] to establish a condition for determining when a set of rules is sufficient to represent a system. Let RB be a set of rules and RD a set of examples. The subset of RD representing the examples λ covered by RB is the λ -cut of RD and called the covering parameter [18].

Induction procedure uses a similar approach (hierarchical based on the same thresholds) to visit learned rules and conclude the decision making. In order to classify an input query pattern Q , the degree of compatibility of the pattern with each rule of the first stage is calculated. If the pattern is λ covered by the winner, the winner rule classifies the pattern. Otherwise, we try to classify the pattern using rules of the next stage.

Note that the classification of a pattern not covered by any rule in the rule-base is rejected. The classification of a pattern X_p requires conflict resolution if two rules with different consequent classes have the same and maximum value of firing degree, $\mu(X_p).CF$, in equation (3) [18]. The conflict is solved by selecting the rule having higher certainty grade. If both rules have the same CF , then we use the one generated and learned first.

6. Experiments

6.1. Experiment Setup. In the pre-processing phase, First of all, all untrustworthy and redundant patterns (i.e., having missing values and duplicate patterns) are removed from training set [55]. Then for continuous features, to correct the bias in favour of large value features, we used mean and standard deviation of each feature. Each feature value of data point X_p is normalized as:

$$\bar{x}_{pi} = (x_{pi} - \text{mean}_q(x_{qi})) / (8 \times \text{stdev}_q(x_{qi})) + 0.5 \quad (13)$$

Various normalization methods may fit different feature vectors. Here we normalized features 5 and 6 with a logarithmic measure. These two features are spanned over a very large integer range (src-bytes in range [0, 1.3 billion] and dst-bytes in range [0, 1.3 billion]), but most of their values are mapped to a small range which is comparable to the scale reckoned for other features. Therefore these two features are normalized as:

$$\bar{x}_{pi} = \log(x_{pi} - \text{min}_q(x_{qi}) + 1) / \log(\text{max}_q(x_{qi}) - \text{min}_q(x_{qi}) + 1) \quad (14)$$

After data normalization, an instance selection technique, explained in [39], was used to reduce the size of training set to 3314 samples. The details of data distribution in the selected subset are provided in Table 1. KDD CUP 99 dataset is extremely imbalanced in the size of classes. Most of instances are in two major classes, namely *normal* and *DOS*, while instances of *U2R* and *R2L* attacks are only 5.27 percent of the dataset. A simple fuzzy rule-based classifier (versus our proposed hierarchical classifier) will generate strong rules (i.e. rules with high firing degrees for most of training instances) for normal class. These rules decide about most of the samples in this major class and provide appropriate overall accuracy, but these rules have an important impact on low detection rate of minor classes. Our proposed system overcomes this problem by means of hierarchical rule generation, selection and weighting.

The fuzzy reasoning is also done in a hierarchical approach. For each query pattern, first of all, we find the degree of fulfilment (i.e., the firing strength) of the fuzzy rules with single antecedent. Then we use the λ threshold to stop the reasoning procedure. The fuzzy rules with two antecedents are involved just if the firing strength of the winner rule is less than a certain threshold.

6.2. Performance Evaluation. Table 2 gives some statistical details of our experiments with KDD99. The table shows that a small subset of generated rules is selected in each stage and a subset of instances most compatible with these rules is removed. Finally, a compact rule-base with 431 rules is generated and used in the inference which is appropriate for a very complex feature space.

The classification rate of different classes, obtained in each stage, is shown in Table 3 and Figure 5. As illustrated in Figure 5, the classification of *Normal* and *DOS* classes reached a satisfactory accuracy faster than minority classes. Although the detection of *U2R* and *R2L* instances is difficult, but the detection accuracy of these attacks in the fourth stage outperforms major intrusion detection methods, which are reported in Table 4. This table shows that the proposed method outperforms all major methods in detection rate of *U2R* and *R2L* methods.

Stage	1	2	3	4	Total
total number of rules(antecedent combinations)	246	29,520	2,302,560	131,245,920	133,578,246
number of generated rules	123	6,774	206,517	3,565,691	3,565,691
number of rules in the final rule-base	16	167	183	65	431
number of removed training instances	593	1,496	827	398	3,314

TABLE 2. The Statistical Details of Learning Stages

	Class label	Stage 1	Stage 2	Stage 3	Stage 4
Train	Normal	31	84	94	99
	DOS	0	33	45	95
	U2R	0	0	24	57
	R2L	15	19	19	30
	Probe	0	20	45	83
Test	Normal	23	69	98	99
	DOS	0	28	35	87
	U2R	0	0	3	25
	R2L	0	0	0	18
	Probe	0	14	38	76

TABLE 3. The Accuracy Results

In Table 4, we compared the performance of the proposed method with various methods: simple 1-NN classifier; winner of the KDD99 contest, which used a decision tree classification algorithm [46]; kernel-miner which is a decision forest [35]; the fuzzy rule-based genetic classifier proposed by Ozyer [41]; PNrul which is a rule-based system [1]; XCS [13] and UCS [51] which are the state-of-the-art evolutionary classifier, and use GA for introducing new rules into the population [12, 52, 58]; MOGFIDS fuzzy rule-based system that is evolved from an agent based evolutionary framework and can act as a genetic feature selection wrapper [57].

It should be mentioned that the UCS selects 7779 rules which is in the best case improved to 510 rules in UCSSE method [51]. Tables 2 and Table 4 show that we could reach both accuracy and interpretability. Although we need some rules with four antecedents but they are only 15 percent of the rule-base.

In Table 5, the resultant confusion matrix for test data set is presented. There is a trade off between detection of major classes and minor classes. Increasing the

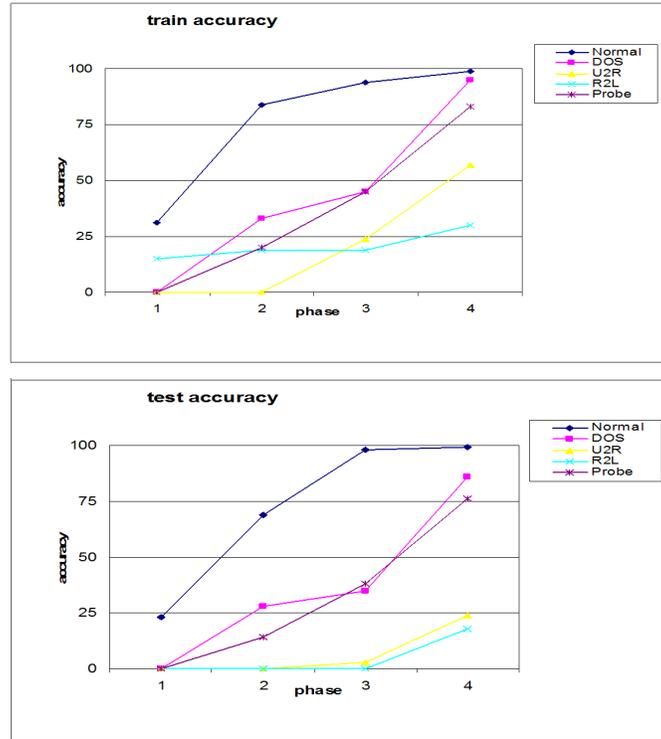


FIGURE 5. Changes of Accuracy for Each Class in Each Phase

detection rate of minor classes usually cause higher false alarms. In the other words, increasing the overall accuracy leads to lower error for *Normal* and *DOS* classes, but increasing the detection rate of *U2R* and *R2L* leads to lower detection rate for *Normal* and *DOS* classes.

Class Label \ Method	Method								
	1-NN	KDD contest winner	kernel miner	Ozyer	PN rule	XCS	UCS SE	MOG FIDS	EBHFRBS
Normal	99.6	99.4	99.4	95.8	99.5	95.7	99.1	98.4	99.1
DOS	97.3	97.1	97.5	97.4	96.9	49.1	96.7	97.2	86.5
U2R	03.5	13.2	11.8	10.9	06.6	08.5	21.3	15.8	24.6
R2L	00.6	08.4	7.3	06.9	10.7	03.9	2.59	11.0	18.3
Probe	75.0	83.3	84.5	54.1	73.2	93.0	75.4	88.6	76.4

TABLE 4. Comparison of major methods

Actual Class	Predicted Class						%Correct
	Normal	DOS	U2R	R2L	Probe		
Normal	60048	349	3	6	187		99.1
DOS	29326	198823	0	100	1604		86.5
U2R	147	0	56	4	21		24.6
R2L	13167	2	45	2963	12		18.3
Probe	728	247	0	7	3184		76.4
F.P.	0.42	0.00	0.46	0.04	0.36		

Accuracy =85.23% , FP=0.89%

TABLE 5. Final Results of EBHFRBS on KDD Test Data

The results show that our method will implicitly consider the higher importance of instances of minor classes.

IDS is one of the fields that misclassification costs are not the same. Obviously, failing to detect an intruder is more costly than misclassifying a normal user as intruder [15]. Indeed, if a normal user's logon fails due the false-positive prediction of the IDS, the imposed cost is not more than a further try by the user. On the other hand, granting permission to an intruder may cause illegal access to information that can be quite costly [39].

The hierarchical structure of the proposed method provides the chance of generating and selecting proper rules for various classes in different stages. This way, implicitly different misclassification costs are considered in the learning process.

7. Conclusion

We proposed a Hierarchical Fuzzy Rule-Based Classification System to detect intrusions in computer networks. In our method, a new type of fuzzy rule-based systems is proposed which uses a hierarchical rule generation and an entropy measure to determine the effective influence of each rule.

This approach generates rules with specific antecedent length, in each iteration, until no suitable rules will be attained. Therefore, rules with appropriate antecedent length will be generated. In general, it was observed that this approach could correct the bias toward the majority concepts and also increases the use of more relevant data samples in each stage, providing for a more robust form of classification in imbalanced problems.

The experimental results on KDD99 show the effectiveness of the proposed method in tackling the tradeoff between accuracy and comprehensibility of the fuzzy-ruled base systems, which is essential for classification of many real world problems.

REFERENCES

- [1] R. Agarwal and M. V. Joshi, *PNrule: a new framework for learning classifier models in data mining (a case-study in network intrusion detection)*, 2000.
- [2] A. Ahmad and L. Dey, *A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets*, Pattern Recognition Letters, **32(7)** (2011), 1062-1069.
- [3] R. Bace and P. Mell, *Intrusion detection systems. US Dept. of Commerce, Technology Administration*, National Institute of Standards and Technology, 2001.
- [4] M. Behdad, L. Barone, T. French and M. Bennamoun, *On XCSR for electronic fraud detection*, (2012), 139-150.
- [5] E. Biermann, E. Cloete and L. M. Venter, *A comparison of Intrusion detection systems*, Computers & Security, **20(8)** (2001), 676-683.
- [6] S. M. Bridges and R. B. Vaughn, *Fuzzy data mining and genetic algorithms applied to intrusion detection*, The 23rd National Information Systems Security Conference, Baltimore, MA, (2000), 13-31.
- [7] D. J. Brown, B. Suckow and T. Wang, *A survey of intrusion detection systems*, Department of Computer Science, University of California, San Diego, 2002.
- [8] V. Chandola, A. Banerjee and V. Kumar, *Anomaly detection for discrete sequences: a survey*, IEEE Transactions on Knowledge and Data Engineering, **24(5)** (2012), 823-839.

- [9] C. H. Cheng, A. W. Fu, Y. Zhang and Y. Chen *Entropy-based subspace clustering for mining numerical data*, The fifth ACM SIGKDD international conference on Knowledge discovery and data mining, **1(4)** (1999), 84-93.
- [10] T. S. Chou, K. K. Yen and J. Luo, *Network intrusion detection design using feature selection of soft computing paradigms*, International Journal of Computational Intelligence, **4(3)** (2008), 196-208.
- [11] O. Cordón, M. J. del Jesus and F. Herrera, *A proposal on reasoning methods in fuzzy rule-based classification systems*, International Journal of Approximate Reasoning, **20 (1)** (1999), 21-45.
- [12] H. Dam, K. Shafi and H. Abbass, *Can evolutionary computation handle large datasets? A study into network intrusion detection*, AI 2005: Advances in Artificial Intelligence, Springer Berlin Heidelberg, (2005), 1092-1095.
- [13] P. Dixon, D. Corne and M. Oates, *A rule set reduction algorithm for the XCS learning classifier system*, Springer Berlin Heidelberg, (2003), 20-29.
- [14] C. Elkan, *Results of the KDD'99 classifier learning*, ACM SIGKDD Explorations Newsletter, **1(2)** (2000), 63-64.
- [15] S. Ghodratinama, M. R. Moosavi, M. Taheri and M. Zolghadri Jahromi, *A cost sensitive learning algorithm for intrusion detection*, The 18th Iranian Conference on Electrical Engineering (ICEE), (2010), 559-565.
- [16] G. Giacinto, F. Roli and L. Didaci, *A modular multiple classifier system for the detection of intrusions in computer networks*, Multiple Classifier Systems, (2003), 346-355.
- [17] A. Gonzalez, R. Perez and J. L. Verdegay, *Learning the structure of a fuzzy rule: a genetic approach*, Fuzzy Systems and Artificial Intelligence, **3(1)** (1994), 57-70.
- [18] A. Gonzalez and R. Perez, *SLAVE: a genetic learning system based on an iterative approach*, Fuzzy Systems, IEEE Transactions on, **7(2)** (1999), 176-191.
- [19] A. Gonzalez and R. Perez, *Completeness and consistency conditions for learning fuzzy rules*, Fuzzy Sets and Systems, **96(1)** (1998), 37-51.
- [20] S. J. Han and S. B. Cho, *Detecting intrusion with rule-based integration of multiple models*, Computers & Security, **22(7)** (2003), 613-623.
- [21] J. Han, M. Kamber and J. Pei, *Data mining: concepts and techniques, second edition (The Morgan Kaufmann series in data management systems)*, Morgan Kaufmann, (2005), 800.
- [22] S. J. Horng, M. Y. Su, Y. H. Chen, T. W. Kao, R. J. Chen, J. L. Lai and C. D. Perkasa, *A novel intrusion detection system based on hierarchical clustering and support vector machines*, Expert Systems with Applications, **38(1)** (2011).
- [23] H. H. Hosmer, *Security is fuzzy!: applying the fuzzy logic paradigm to the multipolicy paradigm*, In Proceedings of the 1992-1993 workshop on New security paradigms, (1993), 175-184.
- [24] K. Hwang, M. Cai, Y. Chen and M. Qin, *Hybrid intrusion detection with weighted signature generation over anomalous internet episodes*, IEEE Transactions on Dependable and Secure Computing, **4** (2007), 41-55.
- [25] H. Ishibuchi and T. Nakashima, *Effect of rule weights in fuzzy rule-based*, IEEE Transactions on Fuzzy Systems, **9(4)** (2001), 506-515.
- [26] H. Ishibuchi, T. Nakashima and T. Murata, *Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems*, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, **29(5)** (1999), 601-618.
- [27] H. Ishibuchi and T. Yamamoto, *Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining*, Fuzzy Sets and Systems, **141** (2004), 59 - 88.
- [28] H. Ishibuchi and T. Yamamoto, *Comparison of heuristic criteria for fuzzy rule selection in classification problems*, Fuzzy Optimization and Decision Making, **3(2)** (2004), 119-139.
- [29] M. Z. Jahromi and M. R. Moosavi, *Designing cost-sensitive fuzzy classification systems using rule-weight*, The First International Conference on Advances in Information Mining and Management (IMMM), (2011), 168-173.

- [30] H. G. Kayacik, A. Nur Zincir-Heywood and M. I. Heywood, *Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets*, The Third Annual Conference on Privacy, Security and Trust, 2005.
- [31] KDD Cup 1999 Intrusion detection dataset, <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>, 2007.
- [32] T. D. Lane, *Machine learning techniques for the computer security domain of anomaly detection*, Department of Electrical and Computer Engineering, Purdue University, 2000.
- [33] H. Lee, J. Song and D. Park, *Intrusion Detection System Based on Multi-class SVM*, Springer-Verlag Berlin Heidelberg, (2005), 511-519.
- [34] K. C. Lee and L. Mikhailov, *Intelligent Intrusion Detection System*, Intelligent Systems, 2nd International IEEE Conference, **2** (2004), 497-502.
- [35] I. Levin and H. marganit Street, *KDD-99 classifier learning contest: LLSOFT's results overview*, SIGKDD explorations, **1(2)** (2000), 67-75.
- [36] R. P. Lippmann, J. W. Haines, D. J. Fried, J. Korba and K. Das, *The 1999 DARPA off-line intrusion detection evaluation*, Computer Networks, **34(4)** (2000), 579-595.
- [37] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyszogrod, R. K. Cunningham, others and M. A. Zissman, *Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation*, DARPA Information Survivability Conference and Exposition (DISCEX '00), **2** (2000), 12-26.
- [38] A. Mitrokotsa and C. Dimitrakakis, *Ad Hoc Networks Intrusion detection in MANET using classification algorithms : the effects of cost and model selection*, AD HOC Networks, 2012.
- [39] M. R. Moosavi, M. Zolghadri Jahromi, S. Ghodrathnama, M. Taheri and M. H. Sadreddini, *A Cost sensitive learning method to tune the nearest neighbour for intrusion detection*, The Iranian Journal of Science and Technology, Transaction of Electrical & Computer Engineering, **36(E2)** (2012).
- [40] M. R. Moosavi, M. Fazaeli Javan, M. Zolghadri Jahromi and M. H. Sadreddini, *An adaptive nearest neighbor classifier for noisy environments*, The 18th Iranian Conference on Electrical Engineering, (2010), 576-580.
- [41] T. Ozyer, R. Alhajj and K. 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.
- [42] Z. S. Pan, S. Chen, G. B. Hu and D. Q. Zhang, *Hybrid neural network and C4.5 for misuse detection*, Machine Learning and Cybernetics, International Conference on, **4** (2003), 2463-2467.
- [43] M. Panda, A. Abraham, S. Das and M. R. Patra, *Network intrusion detection system: a machine learning approach*, Intelligent Decision Technologies, **5(4)** (2011), 347-356.
- [44] A. Patcha and J. min Park, *An overview of anomaly detection techniques : existing solutions and latest technological trends*, Computer Networks, **51** (2007), 3448-3470.
- [45] L. Peng and J. Zhang, *An entropy weighting mixture model for subspace clustering of high-dimensional data*, Pattern Recognition Letters, **32(8)** (2011), 1154-1161.
- [46] B. Pfahringer, *Winning the KDD99 classification cup: bagged boosting*, ACM SIGKDD Explorations Newsletter, **1(2)** (2000), 65-66.
- [47] P. E. Proctor, *Practical intrusion detection handbook*, Prentice Hall PTR, (2001), 392.
- [48] J. R. Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann, (1993), 280.
- [49] M. Sabhnani and G. Serpen, *Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set*, Intelligent Data Analysis, **8(4)** (2004), 403-415.
- [50] H. Schumacher and S. Ghosh, *A fundamental framework for network security*, Journal of Network and Computer Applications, **20(3)** (1997), 305-322.
- [51] K. Shafi and H. A. Abbass, *An adaptive genetic-based signature learning system for intrusion detection*, Expert Systems with Applications, **36(10)** (2009), 12036-12043.
- [52] K. Shafi, T. Kovacs, H. Abbass and W. Zhu, *Intrusion detection with evolutionary learning classifier systems*, Natural Computing, **8(1)** (2009), 3-27.
- [53] C. E. Shannon and W. Weaver, *The mathematical theory of communication*, Urbana, IL: Univ. of Illinois Press, 1949.

- [54] S. S. Sivatha Sindhu, S. Geetha and A. Kannan, *Decision tree based light weight intrusion detection using a wrapper approach*, Expert Systems with Applications, **39(1)** (2012), 129-141.
- [55] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, *A detailed analysis of the KDD CUP 99 data set*, The Second IEEE International Conference on Computational intelligence for Security and Defense Applications, (2009), 53-58.
- [56] A. N. Toosi and M. Kahani, *A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers*, Computer Communications, **30(10)** (2007), 2201-2212.
- [57] 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, **40(9)** (2007), 2373-2391.
- [58] S. X. Wu and W. Banzhaf, *The use of computational intelligence in intrusion detection systems: A review*, Applied Soft Computing, **10(1)** (2010), 1-35.
- [59] C. Xiang, M. Y. Chong and H. L. Zhu, *Design of multiple-level tree classifiers for intrusion detection system*, IEEE Conference on Cybernetics and Intelligent Systems 2004, **2** (2004), 873-878.
- [60] C. Xiang, P. C. P. C. Yong and L. S. L. S. Meng, *Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees*, Pattern Recognition Letters, **29(7)** (2008), 918-924.
- [61] J. Yao, M. Dash, S. S. Tan and H. Liu, *Entropy-based fuzzy clustering and fuzzy modeling*, Fuzzy Sets and Systems, **113(3)** (2000), 381-388.
- [62] M. J. Zolghadri and E. G. Mansoori, *Weighting fuzzy classification rules using receiver operating characteristics (ROC) analysis*, Information Sciences, **177(11)** (2007), 2296-2307.

MOHAMMAD REZA MOOSAVI*, DEPARTMENT OF COMPUTER SCIENCE AND ENG. AND IT, SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN

E-mail address: smmosavi@shirazu.ac.ir, mrmoosavi@gmail.com

MAHSA FAZAEI JAVAN, DEPARTMENT OF COMPUTER SCIENCE AND ENG. AND IT, SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN

E-mail address: mfjavan@gmail.com

MOHAMMAD HADI SADREDDINI, DEPARTMENT OF COMPUTER SCIENCE AND ENG. AND IT, SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN

E-mail address: sadredin@shirazu.ac.ir

MANSOOR ZOLGHADRI JAHROMI, DEPARTMENT OF COMPUTER SCIENCE AND ENG. AND IT, SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN

E-mail address: zjahromi@shirazu.ac.ir

*CORRESPONDING AUTHOR