Applied Soft Computing 9 (2009) 462–469

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Intrusion detection using fuzzy association rules

Arman Tajbakhsh, Mohammad Rahmati *, Abdolreza Mirzaei

Computer Engineering Department of Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 19 November 2006
Received in revised form 4 November 2007
Accepted 1 June 2008
Available online 7 June 2008

Keywords:
Association rule
Association hyper-edge
Classification
Intrusion detection
Matching measure

ABSTRACT

Vulnerabilities in common security components such as firewalls are inevitable. Intrusion Detection Systems (IDS) are used as another wall to protect computer systems and to identify corresponding vulnerabilities. In this paper, a novel framework based on data mining techniques is proposed for designing an IDS. In this framework, the classification engine, which is actually the core of the IDS, uses Association Based Classification (ABC). The proposed classification algorithm uses fuzzy association rules for building classifiers. Particularly, the fuzzy association rulesets are exploited as descriptive models of different classes. The compatibility of any new sample (which is to be classified) with different class rulesets is assessed by the use of some matching measures and the class corresponding to the best matched ruleset is declared as the label of the sample. A new method is also proposed to speed up the rule induction algorithm via reducing items that may be included in extracted rules.

KDD-99 dataset is used to evaluate the proposed framework. Although results on unseen attacks are not so promising, total detection rate and detection rate of known attacks is significant while false positive rate is kept low. Results are compared with some recent works in the literature using the same dataset. Generally, the proposed approach outperforms other methods, specially in terms of false positive rate.

1. Introduction

Reliance on Internet and world wide connectivity has increased the potential damage that can be inflicted by attacks launched over Internet against remote systems. Successful attacks inevitably occur despite the best security precautions. Therefore, intrusion detection has become an essential component of computer security to detect these attacks with the aim of preserving systems from widespread damages and identifying vulnerabilities of the intruded system.

Intrusion detection techniques can be categorized into *anomaly detection* and *misuse detection*. Anomaly detection systems, flag observed activities that deviate significantly from the established normal usage patterns as anomalies (i.e. possible intrusion). While misuse detection systems, use patterns of well-known attacks or weak spots of the system to match and identify known intrusion patterns or signatures [15]. Signature-based misuse detection techniques are currently most widely used in practice. However, there is a growing interest in intrusion detection community toward the application of machine learning techniques in this field. Considering this new trend and the extensive amount of data involved in such learning application [21], data mining approaches seem to be appropriate for this purpose [10,15].

Various data mining techniques have been applied to intrusion detection because it has the advantage of discovering useful knowledge that describes user's or program's behavior from large audit data sets. Statistics, artificial neural network, HMM (Hidden Markov Model), rule learning, and outlier detection scheme are some of the data mining techniques widely used for anomaly and misuse detections [11].

Statistics is the most widely used technique in intrusion detection [4]. NIDES (Next-generation Intrusion Detection Expert Systems) is the representative IDS based on statistics that measures the similarity between a subject's long-term behavior and short-term behavior for intrusion detection [2].

Hyperview of CS Telecom is a representative IDS using neural network [9]. It consists of two modules: neural network and expert system. The neural network in Hyperview uses temporal sequence of 60 types of audit data as inputs. R. Lippmann et al. have applied neural network to keyword-based detection system [17]. They have used keyword counts from transcripts for telnet session as inputs of neural network. While the artificial neural network has some similarity to statistical techniques, it has the advantage of easier representation of nonlinear relationship between input and

* Corresponding author. Tel.: +98 21 6454 2741; fax: +98 21 6649 5521.
E-mail address: rahmati@aut.ac.ir (M. Rahmati).

1568-4946/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2008.06.001

output. The defects of neural networks are that its computational load is very heavy and it is difficult to interpret the relationship between inputs and outputs.

HMM is also a useful tool to model the sequence of observed symbols of which the construction mechanism cannot be known. The representative one is the technique proposed by C. Warender of New Mexico University [25]. It uses system call audit trails to measure normal behaviors. Although HMM has a better performance in modeling system call events than other methods, it requires a very long time for modeling normal behaviors.

RIPPER [8], a rule learning tool, has been used for automatic construction of detection models. RIPPER is applied to labeled training data sets and automatically mines the patterns of intrusion in MADAM ID [15]. Though it is a good tool for discovering patterns, it cannot be easily applied to anomaly detection in order to detect novel intrusions.

The outlier detection scheme which is one of the data mining techniques attempts to identify a data point that is very different from the rest of the data. A. Lazarevic et al. have applied it to anomaly detection [14]. They have compared several variations of outlier detection algorithm and in their experiment, local outlier factor (LOF) approach shows the best performance against 1998 DARPA intrusion detection evaluation data.

One common disadvantage of most data mining techniques is the extensive amount of time required for training and learning the model being inspected. The purpose of this paper is to propose a relatively fast data mining based approach to intrusion detection, in which fuzzy association rules are utilized for learning monitored behaviors in a network. Standard Apriori algorithm described by Agrawal and Srikant [1] for mining association rules as modified by Kuok et al. [13] for fuzzy association rules, is used in this work to induce desired sets of rules. A key problem in Apriori algorithm is the execution time. Along with several implementation techniques used to speed up the algorithm, a novel method is proposed to reduce items involved in rule induction without resulting into any considerable information loss.

The rest of this paper is organized as follows. In Section 2 association rules, their extensions and induction algorithms are described in more details. Section 3 presents the proposed framework for intrusion detection. In Section 4, the results of the experiments carried out on KDD-99 dataset are presented and compared with some recent works in literature using the same dataset. Finally Section 5 concludes proposed work and summarizes its contributions.

2. Association rules

2.1. Definitions and applications

Association rule induction is one of the most well-known approaches in data mining techniques [1]. This technique was initially applied to the so-called market basket analysis, which aims at finding regularities in shopping behavior of customers of supermarkets. In particular, these kinds of rules, which are called Boolean association rules, try to identify sets of products (items) that are frequently bought together in a transaction. A discovered rule can tell, for example, people who buy butter and milk will also buy bread. Apriori [1] and Elact [5] are the best known basic algorithms for mining frequent item sets in a set of transactions. The most important issue in association rule induction is the execution time, because it increases exponentially with respect to the number of items. So far, there have been different implementations of primary association rule inductions, aiming at optimizing execution time. Among them, the Christian Borgelt's work [6] can be mentioned as a successful one. He has used prefix tree data structures in order to minimize the time needed to find frequent item sets.

In contrast to Boolean association rules which could only handle simple item-based transactions, the next generation of association rules faces quantitative (e.g. integer, categorical and numerical) attributes. In [22], mining quantitative association rules has been proposed. The algorithm starts by partitioning the attributes domains and combining adjacent partitions. Then, by considering each pair of attribute and its corresponding partition as an item, the problem is transformed into the Boolean one. For instance, considering *age* as an attribute and *10–15* as a partition in *age* domain (*age*, *10–15*) is considered as an item. In this way all transactions are transformed to item-based transactions. Although this method can solve the problems introduced by quantitative attributes, it causes the sharp boundary problem. It either ignores or over-emphasizes the elements near the boundaries in the mining process. Kuok et al. [13] proposed a fuzzy approach to handle both quantitative attributes and sharp boundary problems, while making the induced rules more comprehendible for humans. In particular, they suggested defining fuzzy linguistic values over attributes domains instead of using crisp partitions used in [22]. The concept of fuzzy set is better than partition method because fuzzy sets provide a smooth transition between members and non-members of a set, consequently handle the sharp boundary problem in an appropriate way.

Recently, association rules have been used in pattern recognition problems such as classification [10,16,18]. In [10], the authors have applied fuzzy association rule mining to intrusion detection. In this approach, a set of fuzzy association rules is extracted for each class, and is used as a model for that class. In order to determine the class of a set of transactions, they generate a set of fuzzy association rules from this transaction set and compute the similarity of the extracted rule set with sets mined from each class. The disadvantage of this method is that it cannot classify a single transaction, due to the fact that it requires a set of test transactions to induce a rule set. In [16,18], a new concept called *Class Association Rule* (CAR) is used to solve the classification problems. A Class Association Rule set is a subset of association rules with the specified classes as their consequences. Boolean and fuzzy CARs are used in [16] and [18], respectively, for building classifiers.

The classification algorithm, presented in this paper does not restrict the rules to CARs (as is done in [16,18]) and is also able to classify each single transaction (the property that is not available in [10]).

2.2. Apriori algorithm

Agrawal and Srikant [1] defined an association rule using the following notation: database T is a collection of n transactions, $\{T_1, T_2, \ldots, T_n\}$ and I is the set of all items, $\{i_1, i_2, \ldots, i_m\}$, where each of the transactions $T_j (1 \leq j \leq n)$ in the database T represents a set of items ($T_j \subseteq I$). An itemset is defined as a non-empty subset of I. An association rule can be represented as: $X \rightarrow Y(c, s)$, where $X \subseteq I$, $Y \subset I$ and $X \cap Y = \varnothing$. In this association rule, s is called *support* and c is *confidence* of the association rule. The *support* is the percentage of the transactions in which both X and Y appear in the same transaction and the *confidence* is the ratio of the number of transactions that contain both X and Y to the number of transactions that contain only X.

The main problem of association rule induction is that there are so many possible rules. Obviously such a vast amount of rules cannot be processed by inspecting each one in turn. The simplest way to restrict the search space and to select important rules is defining a minimum threshold for the support of the rules'

underlying itemsets. Itemsets satisfying this minimum support criterion are called *frequent itemsets*.

The basic Apriori algorithm [1] works in two steps. First, it finds frequent itemsets for Boolean association rules, receiving as input a database T of transactions and the minimum support for the rules. In the second step association rules are generated from the frequent itemsets found in the first step. Usually the first step is more important, because it accounts for the greater part of the processing time. It exploits the Apriori property which is based on the simple observation that no superset of an infrequent itemset (i.e., an itemset not having minimum support) can be frequent (can have enough support). A more detailed description of these two steps can be found in [1,6].

2.3. Selecting interesting rules

Confidence and support of the rules are not sufficient measures to select "interesting" rules. Consider the following case in the market basket problem: let us assume that minimum confidence limit is set to 60% and inspecting the transactions, it is observed that 60% of all customers buy some kind of bread, and 62% of the customers that buy cheese will also buy bread. Therefore, the two following rules will be induced, "(*no antecedent*) → bread" (confidence 60%) and "cheese → bread" (confidence 62%). Obviously the latter one cannot be considered as an interesting rule, since the fact that customer buys cheese does not have a significant influence on him/her buying bread. Formally speaking, adding an item to the antecedent is informative only if it significantly changes the confidence of the rule, otherwise the simpler rule suffices [7]. Using this principle to select a subset of induced association rules, not only keeps all informative rules, but also decreases the processing time in the next steps (classification in our case).

2.4. Fuzzy association rule induction

As stated before, Apriori algorithm can only be used to induce Boolean association rules, while usually databases contain quantitative values. Kuok et al. [13] proposed fuzzy association rule induction to handle this problem. A brief description of their approach is provided here.

Let $T = \{t_1, t_2, \ldots, t_n\}$ be the database and t_i represents the ith record in T. Moreover, $I = \{i_1, i_2, \ldots, i_m\}$ is used to represents all attributes appeared in T and i_j represents the jth attribute. Table 1 is a sample database with quantitative attributes. In this example $T = \{t_1, t_2, t_3\}$ and $I = \{retired, children, salary\}$. We can retrieve the value of attribute i_k in the jth record simply by $t_j[i_k]$. For example if we want to know the value of Salary of the second record, we can use $t_2[salary]$ and get the value 20,000.

Besides, each attribute i_k is associated with several fuzzy sets. $F_{i_k} = \{f_{i_k}^1, f_{i_k}^2, \ldots, f_{i_k}^l\}$ represents the set of fuzzy sets associated with i_k, and $f_{i_k}^j$ represents the jth fuzzy set in F_{i_k}. For example, if the attribute Salary has three fuzzy sets defined for it, e.g. high, medium, and low, we have $F_{\text{salary}} = \{high, medium, low\}$. The fuzzy sets and the corresponding membership functions are provided by domain experts or obtained through an automated approach.

Table 1
A sample database

Retired	Children	Salary
Yes	2	0
No	1	20,000
yes	2	15,000

Given a database T with attributes I and the definitions of fuzzy sets associated with attributes in I, the objective is to find out some interesting regularities between attribute values in a guided way. Any fuzzy association rule is in the following form:

$$\text{If } X \text{ is } A \text{ then } Y \text{ is } B. \tag{1}$$

In the above rule, $X = \{x_1, x_2, \ldots, x_p\}$ and $Y = \{y_1, y_2, \ldots, y_q\}$ are attribute sets. X and Y are disjoint subsets of I. $A = \{f_{x_1}, f_{x_2}, \ldots, f_{x_p}\}$ and $B = \{f_{y_1}, f_{y_2}, \ldots, f_{y_q}\}$ contain the fuzzy sets associated with the corresponding attributes in X and Y. For example $f_{x_k} \in F_{x_k}$ is a fuzzy set, defined on x_k domain. Each pair of $\langle x_k, f_{x_k}\rangle$ or $\langle y_k, f_{y_k}\rangle$ is called an item, and each pair of $\langle X, A\rangle$ or $\langle Y, B\rangle$ is called an itemset.

The first part of the rule 'X is A' is called the antecedent and 'Y is B' is called the consequent of the rule. The semantics of the rule is when 'X is A' is satisfied, we can imply that 'Y is B' is also satisfied. Here the word "satisfied" means there are sufficient amount of records which contribute their votes to the attribute/fuzzy set pairs and the sum of these votes is greater than a user specified threshold.

An appropriate rule should have enough *significance* and a high *certainty* factor. Significance and certainty factor are two concepts, equivalent to support and confidence that were introduced by Agrawal [1] in the domain of Boolean association rules.

Fuzzy association rule induction algorithm has two steps. As in Apriori algorithm, the first step is to find the so-called *large itemsets* which are itemsets with significance higher than a user specified threshold.

The significance factor of an itemset is calculated by first summing all votes of each record with respect to that itemset, then dividing it by the total number of records.

Let $X = \{x_1, x_2, \ldots, x_p\}$ be a set of attributes and $A = \{a_1, a_2, \ldots, a_p\}$ be a set of fuzzy sets, such that $a_j \in F_{x_j}$ and also let $\{m_{a_1}, m_{a_2}, \ldots, m_{a_p}\}$ be the set of membership functions of A, such that m_{a_j} represents the membership function of a_j. The following formula is used to calculate the significance factor of $\langle X, A\rangle$, i.e. $S_{\langle X, A\rangle}$.

$$\begin{aligned}\text{Significance} &= \frac{\text{Sum of votes satisfying } \langle X, A\rangle}{\text{Number of records in } T}\\ S_{\langle X,A\rangle} &= \frac{\sum_{t_i \in T}\prod_{x_j \in X}\{\alpha_{a_j}(t_i[x_j])\}}{||T||}\end{aligned} \tag{2}$$

where $||T||$ represents the number of records in T and

$$\alpha_{a_j}(t_i[x_j]) = \begin{cases} m_{a_j \in A}(t_i[x_j]) & \text{if } m_{a_j} \geq \omega,\\ 0 & \text{otherwise.}\end{cases} \tag{3}$$

in which, ω is a threshold used to avoid taking low membership values into account. As it can be observed in (2), operator $\prod(mul)$ is used to compute each record's vote. Other operators, such as *min*, can also be used instead of *mul*, however authors in [13] have concluded that *mul* achieves better results.

The second step in fuzzy association rule induction is generating rules using the *large itemsets* induced in the first step, considering *certainty factor*.

Let $\langle Z, C\rangle$ be a *large itemset*. Any fuzzy association rule induced from this *large itemset* will be in the form 'if X is A then Y is B', where $X \subset Z$, $Y = Z - X$, $A \subset C$ and $B = C - A$. *Certainty factor* for this rule is calculated as follows.

$$\begin{aligned}\text{Certainty} &= \frac{\text{Signinicance of } \langle Z, C\rangle}{\text{Significance of } \langle X, A\rangle}\\ C_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle} &= \frac{\sum_{t_i \in T}\prod_{z_k \in Z}\{\alpha_{c_k}(t_i[z_k])\}}{\sum_{t_i \in T}\prod_{x_j \in X}\{\alpha_{a_j}(t_i[x_j])\}}\end{aligned} \tag{4}$$

where

$$\alpha_{c_k}(t_i[z_k]) = \begin{cases} m_{c_k \in C}(t_i[z_k]) & \text{if } m_{c_k} \geq \omega, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

As it can be understood from (4) and (5), *certainty* reflects the ratio of votes supporting $\langle Z, C \rangle$ to those supporting $\langle X, A \rangle$.

3. Detection framework

The proposed framework for intrusion detection in distinctly composed of two phases; training and detection phases. Fig. 1 depicts a schematic view of different modules involved. *Rule induction* and *rule filtering* are described in Section 2 (for further details see [5,6,7,13]).

Before addressing other involved modules, it is better to exactly define what the term *item* means in dealing with fuzzy association rules (although it is formally defined in Section 2). Simply each attribute/value pair is called an *item*; e.g. let 'duration' and 'protocol' be two features, then duration/small or protocol/TCP are two instances of valid items.

Feature to item transformation, used in both training and detection phases (Fig. 1), is responsible for transforming an attribute-based record (a record which consists of several numerical or categorical features) into a set of valid items. As described in Section 2.1 this process is required for inducing association rules based on quantitative features. In the case of numerical attributes, this transformation is actually a fuzzification process, and for symbolic or categorical attributes means considering the corresponding attribute/value pair as an item with membership value 1.

Other modules (modules having bold borders in Fig. 1) are discussed in more details.

3.1. Defining fuzzy membership functions

Defining appropriate fuzzy membership functions on different attribute domains is another key issue we are engaged with. Number of fuzzy sets to be defined for each attribute, shape of the membership functions, and parameters defining each function are three questions that are to be answered. However, the problem that makes the membership function generation a non-trivial task, is that there is no guidelines or rules that can be used to choose the appropriate membership generation technique [19] and so experimental approaches seem to be the one solution to choose the best membership generation techniques. In the literature, one can also find plenty of techniques proposing tuning approaches for generated membership functions [3,20], which however we have not used in this paper. In this work, a fixed number of fuzzy sets are defined for every attribute (i.e. three), and all membership functions are considered to have a trapezoidal shape.

One approach to induce the parameters of trapezoids which present memberships is based on Fuzzy C-Means (FCM) clustering algorithm. Following the notations used so far, let $i_k \in I$ be the attribute whose corresponding fuzzy sets are to be induced, and $T = \{t_1, t_2, \ldots, t_n\}$ be the set of all transactions that are to be used for training, also let N be the desired number of fuzzy sets. At first step, the values $t_j[x_k]$(for each j such that $1 \leq j \leq n$) are clustered into N clusters, using FCM. For each generated cluster, we exploit the membership degrees of samples to fit a trapezoid. The fitted trapezoid will represent the membership function corresponding to that cluster. Fig. 2 part (a) shows the distribution of the values of a sample attribute, and part (b) illustrates the procedure of trapezoidal membership function fitting for that attribute. In Fig. 2 part (b) the membership degrees of the samples to the first, second and third clusters, are respectively plotted by circle, plus and dot symbols.

Experiments carried out using this method on some well-known machine learning databases, have achieved promising results. This membership generation method has remarkably outperformed the equally distributed membership functions (see [24] for more details).

3.2. Item reduction

The execution time of association rule induction algorithms exponentially increases with the increase in number of valid items. In the case of KDD-99 dataset there are totally 41 features used to describe each session. Among these features there are 10 symbolic features and 31 numerical features. Considering three valid items for each numerical feature and one valid item for each valid value of any of the symbolic features, there exists 189 valid items in KDD-99 dataset. Rule induction using this amount of items is practically impossible. Here, an algorithm is proposed to reduce items before rule induction.

A concept called *association hyper-edge* is used as the basis of item reduction. Association hyper-edges are sets of items that are strongly predictive with regard to each other [7]. For example the item set $\{a, b\}$ is considered as a hyper-edge if the average confidence of the rules: $a \rightarrow b$ and $b \rightarrow a$ is greater than a threshold. Considering a high value for this threshold (near 100%), this means

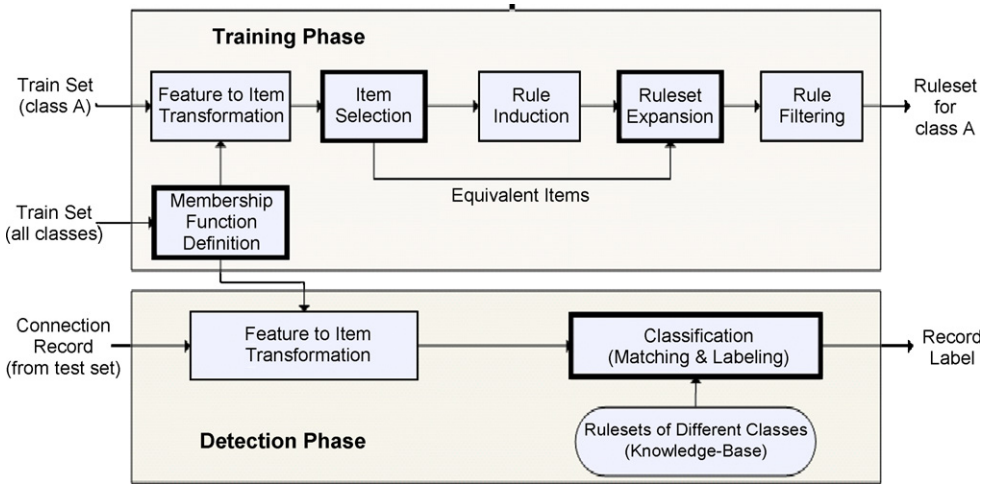

Fig. 1. Block diagram of both train and test phases in proposed intrusion detection framework.

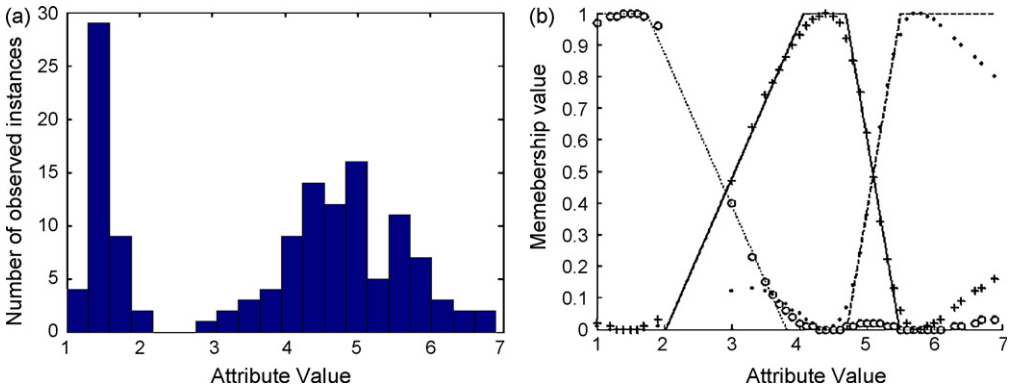

Fig. 2. Membership induction: (a) the histogram of attribute values; (b) clustering based membership function definition.

that, items *a* and *b* are observed together most of the times. In this case rule induction can be carried out, ignoring one of these two items, say '*b*', and when the ruleset is obtained each extracted rule that contains item '*a*' must be duplicated by substituting '*b*' for each occurrence of '*a*'. Of course the rules $a \rightarrow b$ and $b \rightarrow a$ should also be added to obtained ruleset. In the same way, a shared item between multiple hyper-edges can represent multiple items. Going one step further, by extracting all association hyper-edges having two members, the item reduction problem can be mapped to a graph problem. To exactly describe such mapping, graph nodes and edges are to be defined; consider each item as a graph node and suppose that an edge exists between two nodes if the items corresponding to those nodes form an association hyper-edge. Then the problem is to find the smallest subset of graph nodes such that every non-member node of the subset is reachable from the nodes in the subset by passing at most one edge.

In this work, this problem is handled through a greedy approach by giving the higher selection priority to the nodes with higher ranks. This method is proposed in [23]. Rule induction is then carried out using items corresponding to the selected nodes and after the induction phase, the obtained ruleset is expanded as described before.

3.3. Classification

Fuzzy association rules are used to build classifiers. In this approach, which will be called *Association Based Classification* (ABC), fuzzy association rulesets are exploited as descriptive models of different classes. We introduce some measures to assess how well any new sample is matched to a ruleset. Using these measures, we declare, the class corresponding to the best matched ruleset, as the label of the sample.

Let T be the set of all training samples in which each sample corresponds to one of L possible classes. We partition T to L disjoint subsets $(T_1, T_2, \ldots, T_L)$ such that $T_l (1 \leq l \leq L)$ contains all samples of class l. These subsets are independently used to induce L rulesets $(R_1, R_2, \ldots, R_L)$ such that for each l $(1 \leq l \leq L)$, $R_l = \{r_{l1}, r_{l2}, \ldots r_{lP_l}\}$ contains rules describing the patterns observed in class l.

The next step, i.e. the classification phase, is to assign a label to a new sample, say t. Fig. 3 illustrates our approach to this step.

In Fig. 3, $MM_{\langle t,R_l \rangle}$ provides us with a measure to estimate how well a sample, t, can be matched to ruleset R_l, and the function *FindBestMatch*, which is usually a simple minimum or maximum function, finds the class label whose ruleset is believed to be the best match for the sample t.

By applying ABC to an intrusion detection problem, what we have would be a misuse detection approach, because both normal and attack patterns are to be used in the classification process. In order to be able to implement anomaly-detection based IDSs, an *extension of ABC* is introduced.

ABC extension can be used to solve a two-class classification problem. Naming the involved classes as 'normal' and 'un-normal', the algorithm would be as follows: A Fuzzy association ruleset, R, is induced based on all normal training samples. To label a sample, say t, its compatibility to the ruleset R, i.e. $MM_{\langle t,R \rangle}$, is computed as described before. By the use of a threshold, it is determined to which class this sample belongs. Samples having compatibilities higher than the threshold are considered as normal, and those with lower compatibilities are considered as un-normal. An appropriate threshold value can be estimated by inspecting the compatibility values of training samples (it will be discussed in more details in Section 4).

3.4. Matching measures

Our main issue in classification is to define appropriate matching measures. Following the notations used so far, let $I = \{i_1, i_2, \ldots, i_m\}$ be the set of attributes and $F_{i_k} = \{f_{i_k}^1, f_{i_k}^2, \ldots, f_{i_k}^l\}$ be the set of fuzzy sets defined on the domain of i_k. Also, let t and $R = \{r_1, r_2, \ldots, r_P\}$ be, respectively, the sample and the ruleset that are to be matched.

R can be represented as follows.

$$R: \begin{cases} r_1: \text{if} \quad X_1 \text{ is } A_1 \text{ then } y_1 \text{ is } b_1. (s_1, c_1) \\ \vdots \\ r_p: \text{if} \quad X_p \text{ is } A_p \text{ then } y_p \text{ is } b_p. (s_p, c_p) \\ \vdots \\ r_P: \text{if} \quad X_P \text{ is } A_P \text{ then } y_P \text{ is } b_P. (s_P, c_P) \end{cases} \tag{6}$$

where for each $p (1 \leq p \leq P)$, $X_p = \{x_{p1}, x_{p2}, \ldots, x_{pn_p}\}$ such that $X_p \subset I$ and $A_p = \{a_{p1}, a_{p2}, \ldots, a_{pn_p}\}$ such that for each $j (1 \leq j \leq n_p)$ we have $a_{pj} \in F_{x_{pj}}$. In (6) $y_p \in I$ and $b_p \in F_{y_p}$, and finally s_p and c_p are significance and certainty factors of rule r_p, respectively. The rule format used here is the same as the one introduced in (1) except the fact that the consequent part consists of just one item. This is because rules having multiple items in their consequent can be

```
for l = 1 to L {
        MM⟨t,Rl⟩ = MatchingMeasure(t, Rl);
}
sample_label =
 FindBestMatch(MM⟨t,R1⟩, MM⟨t,R2⟩, ..., MM⟨t,RL⟩);
```

Fig. 3. Classification algorithm in our approach.

decomposed to multiple rules each one having one item in the consequent part.

In order to obtain a matching measure, a fuzzy concept called *firing strength* is used. Firing strength of a rule is the degree of satisfaction of antecedent of the rule by a transaction (or a sample). Considering the transaction t, the firing strength of the rule r_p, i.e. $fs_{\langle t,r_p\rangle}$ is formulated as follows.

$$fs_{\langle t,r_p\rangle} = \prod_{j=1}^{n_p} \alpha_{a_{pj}}(t[x_{pj}]) \tag{7}$$

Considering the above notations, a matching measure for $\langle t, R\rangle$, $\mathrm{MM}_{\langle t,R\rangle}$, is calculated as follows.

$$\mathrm{MM}_{\langle t,R\rangle} = \frac{\sum_{p=1}^{P} \mathrm{RI}_{\langle t,r_p\rangle} \times \mathrm{mm}_{\langle t,r_p\rangle}}{\sum_{p=1}^{P} \mathrm{RI}_{\langle t,r_p\rangle}} \tag{8}$$

where $\mathrm{mm}_{\langle t,r_p\rangle}$ is a measure estimating how well the transaction t is matched with the rule r_p, and $\mathrm{RI}_{\langle t,r_p\rangle}$ is a weight determining the influence ratio of the matching measure of the pth rule in computation of the final matching measure. We will call this quantity, *rule influence*. To obtain $\mathrm{mm}_{\langle t,r_p\rangle}$ and $\mathrm{RI}_{\langle t,r_p\rangle}$ we use the following formulas.

$$\mathrm{mm}_{\langle t,r_p\rangle} = 1 - \alpha_{b_p}(t[y_p]) \tag{9}$$

and also,

$$\mathrm{RI}_{\langle t,r_p\rangle} = \mathrm{fs}_{\langle t,r_p\rangle} \times c_p \tag{10}$$

As it can be seen in (9) $\mathrm{mm}_{\langle t,y_p\rangle}$ represents the deviation of t from the consequent of the rule, so it implies a matching error or a matching distance. But how important this error is, will be determined by the two factors forming $\mathrm{RI}_{\langle t,r_p\rangle}$. The parameter c_p gives us information on how much should we expect the consequent comes true, if the antecedent is satisfied, and $\mathrm{fs}_{\langle t,r_p\rangle}$, in (7), shows the degree to which the antecedent is satisfied. So multiplication of these two factors gives a good interpretation of the $\mathrm{mm}_{\langle t,y_p\rangle}$ importance.

Obviously, using (8) to compute $\mathrm{MM}_{\langle t,R_l\rangle}(1 \le l \le L)$, implies that the best class label for sample t is the class having the minimum matching measure, i.e.

$$\text{sample\_label} = \arg\min_{l=1}^{L} \mathrm{MM}_{\langle t,R_l\rangle} \tag{11}$$

Another matching measure is also introduced. To define this matching measure, each rule is treated as a *fuzzy implication*. Computation of the *truth degree* of this implication with respect to the sample to be classified, t, leads us to our objective.

Consider the implication "X is $A \Rightarrow Y$ is B" in dual logic, and let $v(.)$ be the truth degree operator (e.g. v (X is A) in dual logic can either be 1, if X is A, or 0, if X is not A). So v (X is $A \Rightarrow Y$ is B) can be computed based on dual logic implication truth table, considering v (X is A) and v (Y is B). Considering a fuzzy implication, there are several ways to compute its *truth degree* which is a number in the range 0–1 [26 pp. 144–148]. One of the most known methods in this domain is Mamdani implication [26]. He has suggested the use of *min* operator in this case, i.e.

$$v(X \text{ is } A \Rightarrow Y \text{ is } B) = \min[v(X \text{ is } A), v(Y \text{ is } B)] \tag{12}$$

Considering each fuzzy association rule, say r_p, as a fuzzy implication that is to be evaluated with respect to sample t, we have,

$$v(r_p) = \min[v(X_p \text{ is } A_p), v(y_p \text{ is } b_p)] \tag{13}$$

$$v(X_p \text{ is } A_p|t) = fs_{\langle t,r_p\rangle} \tag{14}$$

$$v(y_p \text{ is } b_p|t) = \alpha_{b_p}(t[y_p]) \tag{15}$$

So we define,

$$\mathrm{mm}'_{\langle t,r_p\rangle} = v(r_p) = \min[fs_{\langle t,r_p\rangle}, \alpha_{b_p}(t[y_p])] \tag{16}$$

And finally, to compute a matching measure for a ruleset (6), we can use the weighted average of $\mathrm{mm}'_{\langle t,r_p\rangle}$ for all $p(1 \le p \le P)$, using rule certainty factors as weights, i.e.

$$\mathrm{MM}'_{\langle t,R\rangle} = \frac{\sum_{p=1}^{P} c_p \times \mathrm{mm}'_{\langle t,r_p\rangle}}{\sum_{p=1}^{P} c_p} \tag{17}$$

And since $\mathrm{MM}'_{\langle t,R\rangle}$ is based on the truth degree of fuzzy implications, the best class label for sample t is the class having the maximum matching measure, i.e.

$$\text{sample\_label} = \arg\max_{l=1}^{L} \mathrm{MM}'_{\langle t,R_l\rangle} \tag{18}$$

4. Experimental results

Fuzzy rule induction and classifier modules are implemented using standard *C*. In the case of rule induction, Borgelt's implementation of Boolean association rule induction [7] is used as a start point. Borgelt, from Otto-von-Guericke University of Magdeburg, has implemented an efficient version of the Apriori algorithm [6]. The use of this software is under the terms of the GNU Public License. Prefix trees are used in this implementation for storing both itemset counters and transactions. This implementation is modified in order to induce fuzzy association rules as described in Section 2.4.

4.1. Data source

As mentioned before, KDD-99 dataset [21] is used to evaluate the proposed framework for intrusion detection. This dataset is a common benchmark for evaluation of intrusion detection techniques. KDD-99 consists of several components, that two of them are used in this work. In all experiments described below, '10%KDD' dataset is used for the purpose of training and the so-called 'Corrected' dataset is used as a test set. Several new and novel never-before-seen attacks have been used in 'Corrected' in order to assess the generalization ability of ID systems. Statistical details of the two KDD components used here are summarized in Table 2.

4.2. Evaluation

Performance of Intrusion Detection Systems are usually evaluated in terms of false positive and detection rates [12], which are estimated as follows:

$$\text{False positive rate} = \frac{\text{Number of false positive}}{\text{Total number of normal connections}} \tag{19}$$

$$\text{Detection rate} = 1 - \frac{\text{Number of false negatives}}{\text{Total number of attack connections}} \tag{20}$$

where false positive (negative) rate is the number of normal (attack) connections labeled as attack (normal).

Table 2
Characteristics of KDD-99 components used for train and test

Dataset	Total attack	Total normal	Total
10%KDD	396,743	97,278	494,021
Corrected	250,436	60,593	311,029

Table 3
Evaluation results of the proposed misuse detection approach

	All records	New attacks excluded	Old attacks excluded
Detection rate (%)	91	97.5	11
False positive rate (%)	3.34	3.34	3.34

Table 4
Detailed evaluation results of the proposed misuse detection approach

	All records	New attacks excluded	Old attacks excluded
DoS	78.9	81.2	2.0
Probe	88.5	88.6	88.5
U2R	68.6	66.7	71.0
R2L	6.2	16.5	0.2

Table 5
Some reported results on KDD dataset [11]

	Data mining	Clustering	K-NN	SVM	H-SOM
Detection rate (%)	70–90	93	91	98	90–91.5
False positive rate (%)	2	10	8	10	7.6–14.5

4.3. Experiments setup and results

Based on two classification approaches presented in Section 3.3 (i.e. ABC and ABC extension) two intrusion detection engines are implemented and tested in this work, one based on misuse detection and the other one based on anomaly detection, and the test results are compared and analyzed.

In all experiments, minimum confidence for association hyper-edge extraction is set to 98% and minimum confidence 50% is used for association rule induction.

Both methods proposed for calculating matching measures were tested. Results reported here are based on the first method introduced (results of the same tests using second method were slightly poorer).

For better assessment of the implemented scenarios, detection rate is reported in three cases; by using all 'Corrected' set records, by excluding unknown (unseen) attacks, and finally by excluding known attacks.

4.3.1. Misuse detection

Train data is categorized into 5 sets, 4 sets containing samples of four different attack categories (i.e. DoS, probe, U2R, R2L) and one set allocated to normal samples. Evaluation results are reported in Tables 3 and 4. Audit records which are best matched to one of the four possible attack categories are reported as attack, and the other records are considered normal. Table 4 details the same test results, reported in Table 3.

As observed in Table 3, except the last case (with exclusion of old attacks), the results are satisfying. Some classification results on the same train and test datasets using other techniques [12] are shown in Table 5. By comparing the results of the proposed approach with these results, it can be concluded that the obtained detection rate is nearly optimum while false detection rate is kept much lower in most cases.

Table 6
Evaluation results of the proposed anomaly detection approach

	All records	New attacks excluded	Old attacks excluded
Detection rate (%)	80.6	85.4	20.5
False positive rate (%)	2.95	2.95	2.95

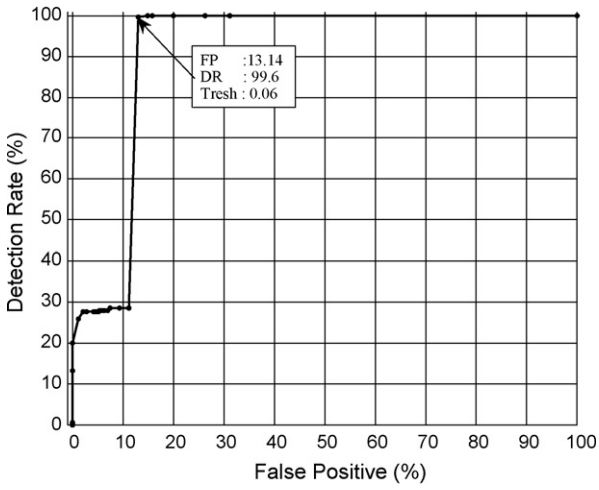

Fig. 4. ROC curve of the proposed anomaly detection system, on training data.

It is difficult to compare train and test time of different approaches, because many of the researches only use a part of train and test sets used in this work, and also the execution time is usually not reported. Total execution time of this work is about 500 s (including train and test) on P4/3GHz/R512MB station, which with regard to the number of train and test records used is completely reasonable.

4.3.2. Anomaly detection

The ABC extension is used in this scenario to classify normal and un-normal records. As mentioned before a key parameter in this classification approach is the compatibility threshold. In this work, the appropriate threshold is determined based on the system ROC curve obtained by carrying out several tests on train data, using different thresholds. Fig. 4 presents the above mentioned ROC curve, and the point selected as the one depicting an appropriate performance. In this way, 0.06 is selected as the compatibility threshold, to be used in anomaly detection scenario. Results obtained using this threshold, are reported in Table 6.

Comparing anomaly detection results with misuse detection method, it can be observed that,

- In contrary to the results reported from common anomaly detection methods, the false positive error rate in the proposed anomaly detection method is kept as low as in misuse detection scenario.
- There is a remarkable decrease in the detection rate of the known attacks in the anomaly detection scenario.
- In the case of unseen attacks the anomaly scenario remarkably outperforms the misuse approach. This is actually the most important advantage of anomaly base approaches.

5. Conclusion

An intrusion detection framework is proposed, implemented and evaluated. Significant results are obtained within a reasonable execution time. Contributions of this work are:

- New classification approach, that uses fuzzy association rulesets as descriptive models of different classes
- Definition of matching measures used in classification

- New item reduction approach to speed up rule induction, by the use of association hyper-edges

The proposed classification algorithm has several advantages in comparison to other classification algorithms that are as follows:

- Human comprehendible rules
- Handling symbolic (categorical) attributes
- Efficient classification on large datasets

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.asoc.2008.06.001.

References

[1] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile, (1994), pp. 487–499.
[2] D. Anderson, T.F. Lunt, H. Javits, A. Tamaru, A. Valdes, Detecting unusual program behavior using the statistical components of NIDES, NIDES Technical Report, SRI International, May 1995.
[3] A. Arslan, M. Kaya, Determination of fuzzy logic membership functions using genetic algorithms, Fuzzy Sets Syst. 118 (2) (2001) 297–306.
[4] E. Biermann, E. Cloete, L.M. Venter, A comparison of Intrusion Detection Systems, Comput. Security 20 (8) (December 2001) 676–683.
[5] C. Borgelt, Efficient implementation of Apriori and Elact, Presented at Workshop of Frequent Item Set Mining Implementations FIMI, USA, 2003.
[6] C. Borgelt, R. Kruse, Induction of association rules: Apriori implementation, Presented at 15th Conference on Computational Statistics (Germany, 2002). Available: http://fuzzy.cs.uni-magdeburg.de/∼borgelt/papers/cstat_02.pdf.
[7] C. Borgelt, 2005, Association Rule Induction, Available: http://fuzzy.cs.uni-magdeburg.de/∼borgelt.
[8] W.W. Cohen, Fast effective rule induction, in: Proceedings of the 12th International Conference on Machine Learning, July 1995, pp. 115–123.
[9] H. Debar, M. Becker, D. Siboni, A neural network component for an Intrusion Detection Systems, in: Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, (May 1992), pp. 240–250.
[10] G. Florez, S.M. Bridges, R.B. Vaughn, An improved algorithm for fuzzy data mining for intrusion detection, in: Proceedings of the North American Fuzzy Information Processing Society Conference NAFIPS, New Orleans, LA, (2002), pp. 27–29.
[11] S.J. Han, S.B. Cho, Detecting intrusion with rule-based integration of multiple models, Comput. Security 22 (7) (2003) 613–623.
[12] H.G. Kayacik, A.N. Zincir-Heywood, M.I. Heywood, On dataset biases in a learning system with minimum a priori information for intrusion detection, in: Proceedings of the 2nd Annual Conference on Communication Networks and Services Research, 2004, pp. 181–189.
[13] C. Kuok, A. Fu, M. Wong, Mining fuzzy association rules in databases, SIGMOD Record 27 (1) (1998) 41–46.
[14] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, J. Srivastava, A comparative study of anomaly detection schemes in network intrusion detection, in: Proceedings of the Third SIAM Conference on Data Mining, May 2003.
[15] W. Lee, S. Stolfo, K. Mok, A data mining framework for building intrusion detection models, in: Proceedings of the IEEE Symposium on Security and Privacy, 1999, pp. 120–132.
[16] J. Li, H. Shen, R. Topr, Mining the optimal Class Association Rule set, Knowledge Based Syst. 15 (2002) 399–405.
[17] R. Lippmann, S. Cunningham, Improving intrusion detection performance using keyword selection and neural networks, Comput. Netw. 34 (4) (2000) 594–603.
[18] B. Liu, W. Hsu, Y. Ma, Integrating classification and association rule mining, in: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, New York, (1998), pp. 80–86.
[19] S. Medasani, J. Kim, R. Krishnapuram, An overview of membership function generation techniques for pattern recognition, Int. J. Approx. Reason. 19 (3) (1998) 391–417.
[20] D. Simon, H∞ estimation for fuzzy membership function optimization, International Journal of Approximate Reasoning 40 (3) (2005) 224–242.
[21] S.J. Stolfo, et al., 1999, KDD-99 dataset, Available: http://www.kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[22] R. Srikant, R. Agrawal, Mining quantitative association rules in large relational tables, in: Proceedings of the International Conference on Management of Data, 1995, pp. 1–12.
[23] A. Tajbakhsh, Design and implementation of an Intrusion Detection Systems using data mining techniques, M.Sc. Thesis, Instructor M. Rahmati, Department of computer engineering, Amirkabir University of Technology, Iran, 2006.
[24] A. Tajbakhsh, M. rahmati, A. Mirzaei, A new classification approach using fuzzy association rules, in: Proceedings of the 11th International CSI Computer Conference, 2006, pp. 263–270.
[25] C. Warrender, S. Forrest, B. Pearlmutter, Detecting intrusion using calls: alternative data models, in: Proceeings of the IEEE Symposium on Security and Privacy, May 1999, 133–145.
[26] H.J. Zimmermann, third ed., Fuzzy Set Theory and its Applications, Kluwer, Boston,1996.