World Applied Sciences Journal 7 (Special Issue of Computer & IT): 31-37, 2009
ISSN 1818-4952
© IDOSI Publications, 2009

Misuse Detection Using Hybrid of Association Rule Mining and Connectionist Modeling

Mansour Sheikhan and Zahra Jadidi

Department of Electrical Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran

Abstract: With the growing of computer networks, the number of attacks has grown extensively. Intrusion detection system (IDS) is known as a critical technology to help protection. In this paper, a hybrid misuse-based IDS, using combined structure of an association rule mining algorithm and a connectionist model, is presented. The key idea is to take advantage of different classification abilities of knowledge-based and machine learning approaches for different attacks. To lower the computational load of association rule mining, the inputs of rule mining algorithm are selected based on the results of a feature relevance analysis. Experimental results show that the proposed hybrid model, in which knowledge-based section of the system reports hard recognizable attack categories, can improve classification results, especially for remote-to-local (R2L) and user-to-root (U2R) attack classes. This hybrid system also offers better detection rate (DR) and cost per example (CPE) compared to neural-based IDS. False alarm rate (FAR) of the proposed model is comparable with other intrusion detection systems, as well.

Key words: Association rule mining . neural network . intrusion detection system

INTRODUCTION

With the growing of computer networks and Internet connectivity, the number of attacks has grown extensively. Preventing attacks only by passive security policies, e.g. firewalls, is difficult. Intrusion detection system (IDS) is known as a critical technology to help protection as an active way.

Based on the information source, IDS may be either host-based or network-based [1]. Host-based IDS examines data held on individual computers that serve as hosts and network-based IDS examines data exchanged between computers, such as traffic volume, Internet protocol (IP) addresses, service ports, protocol usage, etc.

Based on the type of processing modules for analyzing events and detecting potential hostile behavior, IDS may be anomaly-based [2] or misuse-based [3]. Anomaly-based IDS detects activities that vary from established patterns for users and misuse-based IDS compares user's activities with the known behaviors of attackers.

Many soft computing approaches have been applied to the intrusion detection field, e.g. neural networks [4], Markov models [5], fuzzy logic [6], genetic algorithms [7], decision trees [8] and hybrid systems [9].

Given the significance of the intrusion detection problem, a benchmark is provided by the international knowledge discovery and data mining group (KDD) [10]. There are four main categories of attacks given in the KDD. They are denial-of-service (DoS), probe, remote-to-local (R2L) and user-to-root (U2R). DoS attacks deny legitimate requests to a system. Probe involves scanning and probing for getting confidential data. R2L is unauthorized access from a remote user and U2R is unauthorized access to local super-user privileges.

The detection results reported by most of the researchers who employed machine learning algorithms on KDD indicate that DoS attacks and probes are detected accurately whereas attacks involving content (R2L and U2R) have substantially lower detection rates [11].

On the other hand, association rule induction is one of the most well-known approaches in data mining techniques [12]. Recently, association rules have been used in pattern recognition problems such as classification [13-15]. In [14, 15], a new concept called class association rule (CAR) is used to solve the classification problems. A CAR set is a subset of association rules with the specified classes as their consequences. Boolean and fuzzy CARs are used for building classifiers in [14, 15], respectively.

Currently, how to construct a new hybrid intelligent knowledge processing system is a hot research topic. So, in this paper a new model is proposed that combines classification-based association

Corresponding Author: Dr. Mansour Sheikhan, P.O. Box 11365/4435,
Post-Graduate Center, South Tehran Branch, Islamic Azad University, Iran

rule approach with a multilayer perceptron (MLP) neural network, as a hybrid misuse-based IDS. This combination provides improved classification results as compared to MLP-based classifier and also many of the previous systems. In addition, a feature relevance analysis is used to decrease the number of applied features to association rule part of the system.

The subsequent sections of this paper are organized as follows. The foundation of classification based on predictive association rules (CPAR) approach is described in Section 2. KDD 99 dataset is briefly reviewed in Section 2, too. The architecture of the proposed system, method of feature selection and preprocessing procedures are reported in Section 3. Experimental results and conclusions are also drawn in Section 5 and 6, respectively.

CLASSIFICATION BASED ON ASSOCIATION RULE MINING

A hybrid system that combines an association rule mining algorithm and a MLP neural network is proposed in this paper as misuse-based IDS. The foundation of rule-based part of the system is reviewed in this section.

In the field of data mining, a classification approach, called associative classification has been proposed that achieves higher classification accuracy than traditional approaches such as C4.5 [15]. However, this approach suffers from low efficiency due to the facts that it often generates a very large number of rules and also its confidence-based rule evaluation measure may lead to over-fitting [16].

In this paper, a better approach, called CPAR is used, that inherits the basic idea of first order inductive learner (FOIL) in rule generation and integrates the features of associative classification in predictive rule analysis [16].

Definitions: Let T be a set of tuples. Each tuple in T follows the scheme $(A_1, A_2, ..., A_k)$, where $A_1, A_2, ..., A_k$ are k attributes. Each continuous attribute is first turned discrete into a categorical attribute.

Definition 1: (literal) A literal p is an attribute-value pair, taking the form of (A_i, v), in which A_i is an attribute and v a value. A tuple t satisfies a literal $p=(A_i, v)$ if and only if $t_i=v$, where t_i is the value of the i^{th} attribute of t.

Definition 2: (rule) A rule, which takes the form of "$p_1 \wedge p_2 \wedge \ldots \wedge p_l \rightarrow c$", consists of a conjunction of literals $p_1, p_2, ..., p_l$, associated with a class label c. A tuple t satisfies rule r's body if and only if it satisfies every literal in the rule. If t satisfies r's body, r predicts that t is of class c. If a rule contains zero literal, its body is satisfied by any tuple.

```
Input: Training set $D = P \cup N$.
Output: A set of rules for predicting class labels for examples.
Procedure Predictive Rule Mining
    Set the weight of every example to 1
    Rule set $R \leftarrow F$
    $totalWeight \leftarrow TotalWeight(P)$
    $A \leftarrow$ Compute PNArray from $D$
    While $TotalWeight(P) > d.totalWeight$
        $N' \leftarrow N, P' \leftarrow P, A' \leftarrow A$
        Rule $r \leftarrow emptyrule$
        While true
            find best literal $p$ according to $A'$
            if $gain(p) < min\_gain$ then break
            append $p$ to $r$
            for each example $t$ in $P' \cup N'$ not satisfying $r$'s body
                remove $t$ from $P'$ or $N'$
                change $A'$ according to the removal of $t$
            end
        end
        $R \leftarrow R \cup \{r\}$
        for each example $t$ in $P$ satisfying $r$'s body
            $t.weight \leftarrow a. t.weight$
            change $A$ according to the weight decreased
        end
    end
return $R$
```

Fig. 1: Predictive rule mining algorithm

Definition 3: (PNArray) A PNArray stores the following information corresponding to rule r:

- P and N: the numbers of positive and negative examples satisfying r's body.
- P(p) and N(p): for each possible literal p, the numbers of positive and negative examples satisfying the body of rule r', the rule constructed by applying p to r.

Predictive rule mining: The most time-consuming part of FOIL is evaluating every literal when searching for the one with the highest gain [16]:

$$\text{gain}(p) = |P^*| \left(\log \frac{|P^*|}{|P^*| + |N^*|} - \log \frac{|P|}{|P| + |N|} \right) \quad (1)$$

where $|P|$ and $|N|$ are positive and negative examples satisfying the current rule r's body, respectively. After literal p is added to r, there are $|P^*|$ positive and $|N^*|$ negative examples satisfying the new rule's body.

By using PNArray, the predictive rule mining (PRM) algorithm (Fig. 1) achieves much higher efficiency than FOIL on large datasets [15], such as KDD 99. In PRM, after an example is correctly covered

Table 1: Number of samples in KDD 99 datasets

KDD dataset	DoS	Probe	R2L	U2R	Normal
10%	391458	4107	1126	52	97277
corrected	229853	4166	16347	70	60593
whole	3883370	41102	1126	52	972780

Table 2: Attack types and number of their samples in 10% KDD dataset

Category	Type (Number of samples)
DoS	smurf (280790), neptune (107201), back (2203), teardrop (979), pod (264), land (21)
Probe	satan (1589), ipsweep (1247), portsweep (1040), nmap (231)
R2L	warezclient (1020), guess_passwd (53), warezmaster (20), imap (12), ftp_write (8), multihop (7), phf (4), spy (2)
U2R	buffer_overflow (30), rootkit (10), loadmodule (9), perl (3)

by a rule, instead of removing it, its weight is decreased by multiplying a factor (a).

Rule generation in CPAR: In associative classification, association rule mining is used to generate candidate rules, which includes all conjunction of literals that meet the supported threshold. Then, a subset of rules is selected from the candidates. This subset is built by combining the best K rules for every example (K=1 in [15] and K=4 in [17]).

In PRM, every rule is generated from the remaining dataset. Suppose an example t in the remaining dataset is covered by a rule r that is just generated. We are not sure whether r is the best rule for t because r is generated by greedy algorithm and also from the remaining dataset instead of the whole dataset. PRM selects only the best literal and ignores all the others. The "best" rule among them may not be the best rule based on the whole dataset.

But CPAR stands in the middle between exhaustive and greedy algorithms and combines the advantages of both. CPAR builds rules by adding literals one by one, which is similar to PRM. However, instead of ignoring all literals except the best one, CPAR keep all close-to-the-best literals during the rule building process. So, at a certain step in the process of building a rule, after finding the best literal p, another literal q that has similar gain as p (differs by at most 1%) may be found. Besides continuing building the rule by appending p to r, q is also appended to the current rule r to create a new rule r', which is pushed into the queue [16].

Rule evaluation: To evaluate the prediction power of rule r (r="$p_1 \wedge p_2 \wedge \dots \wedge p_l \rightarrow c$"), the Laplace expected error estimate is used to estimate the accuracy of rules [16]:

$$\text{LaplaceAccuracy} = (n_c+1)/(n_{tot}+k) \qquad (2)$$

where k is the number of classes, n_{tot} is the total number of examples satisfying the rule's body, among which n_c examples belong to c, the predicted class of the rule.

Classification: Given a rule set containing rules for each class, the best K rules of each class is used and by comparing the *LaplaceAccuracy* of the best K rules of each class, the class with the highest accuracy is chosen as the predicted class.

KDD dataset: The KDD intrusion detection benchmark consists of three components (Table 1). Each of the four mentioned categories of attacks (DoS, Probe, R2L and U2R) represents generalizations of specific attack types. These main categories represent classifications of types of behavior that can be grouped logically together. So, for each category, there are multiple attack types. Table 2 lists the attack categories along with the attack types and number of samples in the 10% KDD dataset. As indicated in Table 2, "10% KDD" dataset contains 22 attack types. Each connection in KDD is characterized by 41 features. The descriptions, types and value ranges of 19 sample features are listed in Table 3.

PROPOSED IDS ARCHITECTURE

Combination of CPAR and MLP neural network: Benefits of the CPAR algorithm can be combined with those of the neural networks. The key idea is to take advantage of different abilities of neural networks, e.g. experienced by the author in another fields [18-22] and knowledge-based algorithms [23] for intrusion detection. The block diagram of proposed IDS is shown in Fig. 2. As shown in Fig. 2, at the output of proposed hybrid model, the attack types "neptun", "satan", "guess_passwd" and "buffer_overflow" are chosen as the detection objective from DoS, probe, R2L and U2R attack classes, respectively.

Table 3: Description and value ranges of 19 sample features in KDD dataset

Feature	Description	Type	Value ranges
duration	Duration of the connection (in seconds)	continuous	[0,58329]
protocol_type	Type of the connection protocol	discrete	3 different symbols
service	Service on the destination	discrete	70 different symbols
flag	Status flag of the connection	discrete	11 different symb ols
src_bytes	Number of bytes sent from source to destination	continuous	[0,1.3e+9]
dst_bytes	Number of bytes sent from destination to source	continuous	[0,1.3e+9]
wrong_fragment	Number of wrong fragments	continuous	[0,3]
urgent	Number of urgent packets	continuous	[0,14]
hot	Number of "hot" indicators	continuous	0,101]
num_failed_logins	Number of failed logins	continuous	[0,5]
num_compromised	Number of "compromised" conditions	continuous	[0,9]
num_root	Number of "root" accesses	continuous	[0,7468]
num_file_creations	Number of file creation operations	continuous	[0,100]
num_shells	Number of shell prompts	continuous	[0,5]
num_access_files	Number of operations on access control files	continuous	[0,9]
count	Number of connections to the same host as the current connection in the past two seconds	continuous	[0,511]
srv_count	Number of connections to the same service as the current connection in the past two seconds	continuous	[0,511]
dst_host_count	Number of connections having the same destination host	continuous	[0,255]
dst_host_srv_count	Number of connections having the same destination host and using the same service	continuous	[0,255]

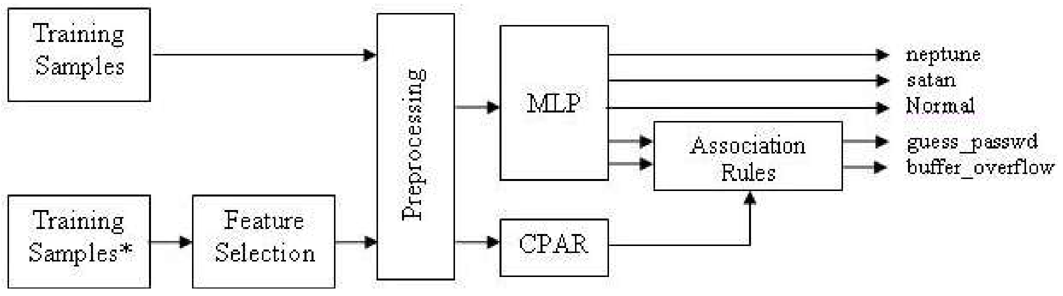

Fig. 2: Block diagram of proposed hybrid model for IDS
*R2L and U2R training samples are used for second experiment

There are many potential benefits of feature selection (e.g. reducing training and utilization times) [24]. So, in this paper the results of feature selection experiments reported in [25] are used in the feature selection box of Fig. 2. The brief overview of feature ranking and selection in KDD dataset are described in the next subsection.

On the other hand, one of the difficulties in applying association rules is involving datasets that contain continuous attributes. The preprocessing procedures in two branches of Fig. 2 are detailed in the next subsection.

Feature selection: Elimination of less significant features lowers the load of CPAR, enhancing the accuracy of detection and speeding up the computation, thus improving the overall performance of IDS.

Logistic regression has been used to rank the features based on the Chi-square values for different subsets selected using best subset selection model [25]. The higher the Chi-square value, the higher is the ranking. In Table 4 the ranking results of the Chi-square test on KDD dataset are listed for the 16 most significant features which are used as the inputs to CPAR algorithm, after preprocessing.

Table 4: Chi-square values of input features with respect to the attack class

	Attack type			
Feature	DoS	Probe	R2L	U2R
dst_host_diff_srv_rate	1334.82	3686.28	1114.09	2531.96
rerror_rate	1016.26	2734.53	1016.54	613.39
dst_host_srv_rerror_rate	967.87	2707.66	586.24	301.06
srv_rerror_rate	805.55	2515.68	583.34	244.92
dst_ host_rerror_rate	732.80	2251.96	560.59	207.83
diff_srv_rate	551.75	1228.26	350.12	39.88
dst_host_same_srv_rate	449.23	793.25	311.15	39.16
service	438.75	588.68	249.51	36.74
dst_host_srv_count	433.03	546.12	239.16	32.61
logged_in	363.64	427.17	141.75	25.15
dst_host_srv_diff_host_rate	353.49	422.29	141.31	25.01
srv_count	344.88	123.40	141.22	15.46
same_srv_rate	336.85	91.77	126.07	15.33
protocol_type	328.69	84.58	125.02	10.68
num_compromised	308.37	70.40	116.02	10.26
wrong_fragment	275.58	68.62	99.83	6.35

Preprocessing: Features in the KDD datasets have different forms: discrete, continuous and symbolic, with significantly varying resolution and ranges. Most

Table 5: Distribution of training and test samples

Attack type	Number of training samples	Number of test samples
neptune	10720	6294
satan	159	96
guess_passwd	53	45
buffer_overflow	30	25
Normal	9728	6059

Table 6: Cost matrix values for KDD 99

	Predicted				
Actual	DoS	Probe	R2L	U2R	Normal
DoS	0	1	2	2	2
Probe	2	0	2	2	1
R2L	2	2	0	2	4
U2R	2	2	2	0	3
Normal	2	1	2	2	0

pattern classification methods are not able to process data in such a format. Hence, preprocessing is required. First, we describe the preprocessing of 41 input features to MLP. Then, the preprocessing details of selected features, input to CPAR, are reported.

Symbolic-valued features, such as "protocol_type", "service" and "flag" are mapped to integer values ranging from *0* to *S-1*, where *S* is the number of symbols. Continuous features having smaller integer value ranges like "wrong_fragment", "urgent", "hot", "num_failed_logins", "num_compromised", "num_root", "num_file_creations", "num_shells", "num_access files", "count", "srv_count", "dst_host_count" and "dst_host_srv_count" are also scaled linearly to the range [0,1].

Logarithmic scaling (base 10) is applied to three features spanned over a very large integer range, namely "duration", "src_bytes" and "dst_bytes", to reduce the ranges. Other features are either Boolean, like "logged_in", having binary values, or continuous, like "diff_srv_rate", in the range of [0,1] and no scaling is needed for these features. So, each of the mapped features are linearly scaled to the range [0,1].

As seen in CPAR foundations, the usage of association rules requires a discrete set of items. These items are literal values of the dataset attributes. In order to use association rules on continuous attributes, the attributes must first be turned discrete. To this end, in the 16 selected features, the continuous ones are partitioned into equal-sized partitions by utilizing equal frequency intervals [26]. In equal frequency intervals method, the feature space is partitioned into an arbitrary number of partitions where each partition contains the same number of data points. In other words, the range of each partition is adjusted to contain *I* dataset instances. If a value occurs more than *I* times in a feature space, it is assigned a partition of its own. In "10% KDD" dataset, certain classes such as DoS attacks and normal connections occur in order of hundreds or thousands, whereas other classes such as U2R and R2L attacks occur in order of tens or hundreds. Therefore, to provide sufficient resolution for the minor classes, *I* is set to 10.

EXPERIMENTAL RESULTS

33209 records from KDD 99 dataset were chosen for our experiments. 20690 of them built the training set and the rest (12519 samples) built the test set (Table 5).

Two experiments were performed based on the structure shown in Fig. 2: using only the MLP classifier (upper branch) for all classes, and using the full structure (hybrid CPAR/MLP classifier).

Before discussing about the results of experiments, it seems necessary to mention the standard metrics that have been developed for evaluating IDS. Detection rate (DR) and false alarm rate (FAR) are the two most common metrics. DR is computed as the ratio between the number of correctly detected attacks and the total number of attacks, while FAR is computed as the ratio between the number of normal connections that is incorrectly misclassified as attacks and the total number of normal connections.

For the purpose of classifier algorithm evaluation, another comparative measure is defined which is cost per example (CPE) [27]. CPE is calculated using the following formula:

$$\mathrm{CPE} = \frac{1}{T}\sum_{i=1}^{m}\sum_{j=1}^{m} \mathrm{CM}(i,j).\mathrm{C}(i,j) \qquad (3)$$

where *CM* and *C* are confusion matrix and cost matrix, respectively. *T* represents the total number of test instances and *m* is the number of classes in classification. *CM* is a square matrix in which each column corresponds to the predicted class, while rows correspond to the actual classes. An entry at row *i* and column *j*, *CM(i,j)*, represents the number of misclassified instances that originally belong to class *i*, although incorrectly identified as a member of class *j*. The entries of the primary diagonal, *CM(i,i)*, stand for the number of properly detected instances. Cost matrix is similarly defined, as well and entry *C(i,j)* represents the cost penalty for misclassifying an instance belonging to class *i* into class *j*. Cost matrix values employed for the KDD 99 classifier learning contest are shown in Table 6 [10].

Table 7: Confusion matrix of MLP classifier

	Predicted				
Actual	neptune	satan	guess_passwd	buffer_overflow	Normal
neptune	6040	4	0	0	250
satan	10	75	0	0	11
guess_passwd	0	0	3	0	42
buffer_overflow	0	1	4	0	20
Normal	43	35	12	0	5969

Table 8: Confusion matrix of CPAR/MLP classifier

	Predicted				
Actual	neptune	satan	guess_passwd	buffer_overflow	Normal
neptune	6036	6	0	2	250
satan	8	77	0	0	11
guess_passwd	2	0	19	0	24
buffer_overflow	2	4	0	14	5
Normal	49	36	10	1	5963

Table 9: Performance of proposed models for intrusion detection

	Classification rate							
Model	neptune	satan	guess_ passwd	buffer_ overflow	Normal	DR	FAR	CPE
MLP	95.96	78.13	6.67	0.00	98.51	94.71	1.49	0.0733
CPAR/MLP	95.90	80.20	42.22	56.00	98.42	95.14	1.58	0.0655

In the first experiment, a MLP with a hidden layer of 40 neurons with tangent-sigmoid activation function, five linear output neurons (representing 4 attack types and 1 normal category) and 41 external input nodes, was used. Levenberg-Marquardt (LM) was used as the training function in this experiment, utilizing MATLAB Neural Network Toolbox. The confusion matrix of this classification method is shown in Table 7.

In the second experiment, the combined structure of CPAR/MLP, depicted in Fig. 2, was used for classification. In this experiment, the parameters of CPAR in rule generation algorithm were set as: δ=0.05, *min_gain*=0.7 and α=2/3. The best four rules were used in prediction, too. The MLP had 41 external input nodes, 35 neurons in hidden layer and five output neurons, as well. The confusion matrix of this hybrid classifier is shown in Table 8.

By applying 12519 test samples (Table 5), the performance of MLP and combined CPAR/MLP classifiers has been compared (Table 9).

CONCLUSIONS

In this paper, a hybrid structure was introduced for IDS that combined classification-based association rule approach with a connectionist model. Most of the machine learning algorithms offered an acceptable level of classification rate for DoS and probe attack categories and demonstrated poor performance on the R2L and U2R categories [11]. As shown in Table 9, the proposed hybrid CPAR/MLP classifier demonstrates better rates in mentioned categories. DR and CPE of the proposed hybrid system are better than MLP classifier, too. FAR of the proposed hybrid architecture is higher than MLP classifier. However FAR of the model is still better than many models (e.g. K-means with 6.21% FAR [28] and radial basis function (RBF) neural classifier with 3.85% FAR [29]).

REFERENCES

1. Kabiri, P. and A.A. Ghorbani, 2005. Research in Intrusion Detection and Response-A Survey. International Journal of Network Security, 1: 84-102.
2. Shon, T. and J. Moon, 2007. A Hybrid Machine Learning Approach to Network Anomaly Detection. Journal of Information Sciences, 177: 3799-3821.

3. Chen, Y., A. Abraham and B. Yang, 2007. Hybrid Flexible Neural-Tree-Based Intrusion Detection Systems. International Journal of Intelligent Systems, 22: 337-352.
4. Beghdad, R., 2008. Critical Study of Neural Networks in Detecting Intrusions. Journal of Computers and Security, 27: 168-175.
5. Yeung, D.Y. and Y. Ding, 2003. Host-Based Intrusion Detection Using Dynamic and Static Behavioral Models. Journal of Pattern Recognition, 36: 229-243.
6. Gomez, J. and D. Dasgupta, 2002. Evolving Fuzzy Classifiers for Intrusion Detection. In the Proceedings of the IEEE Workshop on Information Assurance, pp: 68-75.
7. Song, D., M.I. Heywood and A.N. Zincir-Heywood, 2005. Training Genetic Programming on Half a Million Patterns: An Example from Anomaly Detection. IEEE Transactions on Evolutionary Computation, 9: 225-239.
8. Pfahringer, B., 2000. Winning the KDD 99 Classification Cup: Bagged Boosting. Journal of SIGKDD Explorations, 1: 65-66.
9. Abadeh, M.S., J. Habibi and C. Lucas, 2005. Intrusion Detection Using a Fuzzy Genetic-Based Learning Algorithm. Journal of Network and Computer Application, 30: 414-428.
10. 1999 KDD Cup Competition (Available on http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html).
11. Sabhnani, M. and G. Serpen, 2004. Why Machine Learning Algorithms Fail in Misuse Detection on KDD Intrusion Detection Data Set. Journal of Intelligent Data Analysis, 6: 1-13.
12. Agrawal, R. and R. Srikant, 1994. Fast Algorithms for Mining Association Rules. In the Proceedings of the International Conference on Very Large Databases, pp: 487-499.
13. Florez, G., S.M. Bridges and R.B. Vaughn, 2002. An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection. In the Proceedings of the North American Fuzzy Information Processing Society Conference, pp: 27-29.
14. Li, J., H. Shen and R. Topr, 2002. Mining the Optimal Class Association Rule Set. Knowledge Based Systems, 15: 399-405.
15. Liu, B., W. Hsu and Y. Ma, 1998. Integrating Classification and Association Rule Mining. In the Proceedings of KDD'98, pp: 80-86.
16. Yin, X. and J. Han, 2003. CPAR: Classification Based on Predictive Association Rules. In the Proceedings of SIAM International Conference on Data Mining, pp: 331-335.
17. Li, W., J. Han and J. Pei, 2001. CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In the Proceedings of the International Conference on Data Mining, pp: 369-376.
18. Sheikhan, M., M. Tebyani and M. Lotfizad, 1996. Using Symbolic and Connectionist Approaches to Automate Editing Persian Sentences Syntacticly. In the Proceedings of the International Conference on Intelligent and Cognitive Systems, pp: 250-253.
19. Sheikhan, M., M. Tebyani and M. Lotfizad, 1997. Continuous Speech Recognition and Syntactic Processing in Iranian Farsi Language. International Journal of Speech Technology, 1: 135-141.
20. Sheikhan, M., 2003. Suboptimum Extracted Features and Classifier for Speaker-Independent Farsi Digit Recognizer. In the Proceedings of the International Symposium on Telecommunications, pp: 246-249.
21. Sheikhan, M., 2003. Prosody Generation in Farsi Language. In the Proceedings of the International Symposium on Telecommunications, pp: 250-253.
22. Sheikhan, M., M. Nasirzadeh and A. Daftarian, 2006. Text to Speech for Iraninan Dialect of Farsi Language. In the Proceedings of the Workshop on Farsi Computer Speech, University of Tehran, pp: 39-53.
23. Sanchez, D., M.A. Vila, L. Cerda and J.M. Serrano, 2009. Association Rules Applied to Credit Card Fraud Detection. Journal of Expert Systems with Applications, 36: 3630-3640.
24. Abdollah, M.F., A.H. Yaacob, S. Sahib, I. Mohamad and M.F. Iskandar, 2008. Revealing the Influence of Feature Selection for Fast Attack Detection. International Journal of Computer Science and Network Security, 8: 107-115.
25. Tamilarasan, A., S. Mukkamala, A.H. Sung and K. Yendrapalli, 2006. Feature Ranking and Selection for Intrusion Detection Using Artificial Neural Networks and Statistical Methods. In the Proceedings of the International Joint Conference on Neural Networks, pp: 4754-4761.
26. Wang, A.K.C. and D.K.Y. Chiu, 1987. Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9: 796-805.
27. Agrawal, R. and M.V. Joshi, 2000. PNrule: A New Framework for Learning Classifier Models in Data Mining (A Case-Study in Network Intrusion Detection). IBM Research Division, Report No. RC-21719.
28. Faraoun, K.M. and A. Boukelif, 2006. Neural Network Learning Improvement Using the K-Means Clustering Algorithm to Detect Network Intrusions. International Journal of Computational Intelligence, 3: 161-168.
29. Beghdad, R., 2007. Training All the KDD Dataset to Classify and Detect Attacks. Journal of Neural Network World, 17: 81-91.