

Effects of Feature Reduction on the Performance of Attack Recognition by Static and Dynamic Neural Networks

¹Mansour Sheikhan, ¹Zahra Jadidi and ²Maedeh Beheshti

¹Department of Electrical Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran

²Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

Abstract: One way of dealing with suspicious activities within a computer network is based on using intrusion detection system (IDS). Several soft computing paradigms have been applied to intrusion detection. In this way, feature selection and ranking is an important issue in intrusion detection, as well. In this paper, logistic regression is used to rank the features based on the Chi-square values for different selected subsets. To investigate the effects of feature reduction on classification rates and training time of neural attack recognizers, static and dynamic neural networks are employed and the performance of proposed systems are investigated in terms of detection rate (DR), false alarm rate (FAR) and cost per example (CPE). Empirical results show that the multi-layer perceptron (MLP) with 15 selected input features has higher classification rates for Normal, Probe, denial of service (DoS) and remote-to-local (R2L) attack classes, as compared to some other machine learning methods. This classifier performs better in terms of DR, FAR and CPE, as well.

Key words: Feature reduction • Attack recognition • Neural classifier

INTRODUCTION

The number of attacks in computer networks has grown extensively and many new intrusive methods have appeared. One way of dealing with suspicious activities within a network is based on using intrusion detection system (IDS) [1, 2]. An IDS monitors the activities of environment and decides on its anomaly [3, 4].

On the other hand, soft computing is an approach to develop an intelligent system which mimics the human mind and ability to reason and learn in an environment of uncertainty and imprecision. Soft computing consists of several computing paradigms that have been applied to the intrusion detection field, including neural networks [3-7], fuzzy sets [8], genetic algorithms [9] and hybrid systems [10-15].

In this way, feature selection and ranking is an important issue in intrusion detection, as well. In this paper, logistic regression is used to rank the features based on the Chi-square values for different selected subsets using best subset selection model [16]. The effects of feature reduction on classification rate and training time of neural attack recognizers are investigated in this paper by employing various optimized, by genetic

algorithm (GA), structures for multi-layer perceptron (MLP) and Elman neural networks.

The subsequent sections of this paper are organized as follows. In Section 2, the international knowledge discovery and data mining group (KDD) Cup 99 benchmark dataset [17], on which the experiments are conducted, is briefly reviewed. In Section 3, the dataset preprocessing procedure is reported. As part of the feature selection experiments, the statistical analysis is presented in Section 4. The simulated neural attack recognizers and empirical results are discussed in Section 5. Conclusions are also drawn in Section 6.

INTRUSION DATA

The KDD Cup 99 dataset includes a set of 41 features derived from each connection and a label which specifies the status of connection records (normal or specific attack type). These features fall in four categories [17]: basic features of a connection, content features, time-based traffic features and host-based traffic features. Likewise, attacks fall into four main categories [17]: denial-of-service (DoS), Probe, remote-to-local (R2L) and user-to-root (U2R).

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

E-mail: :msheikhn@azad.ac.ir Tel: +98 21 88605565 Fax: +98 21 88059690

Table 1: Number of selected samples of 10% KDD 99 dataset for training

Connection/attack type	Number of training samples	Distribution (%)
Normal	9727	19.69
Probe	411	0.83
DoS	39145	79.24
U2R	6	0.01
R2L	113	0.23
Total	49402	100.00

Table 2: Number of selected samples of corrected KDD 99 dataset for test

Connection/attack type	Number of test samples	Distribution (%)
Normal	6059	19.48
Probe	417	1.34
DoS	22985	73.90
U2R	7	0.02
R2L	1635	5.26
Total	31103	100.00

KDD dataset is divided into training and testing record sets. Total number of connection records in the training dataset is about 5 million records. In this paper, a subset of 10% KDD training dataset, with the same distribution, is employed (Table 1). As it can be seen in Table 1, sample distributions for different categories of attacks in training data differ significantly from each other. The test data has a different distribution. Moreover, the test data includes additional attack types, not present in the training data, which makes classification more complicated. Table 2 reports the distribution of Normal and attack classes in the test dataset of this research, as a subset of corrected KDD 99 dataset [17].

PREPROCESSING OF FEATURES

Features in the KDD datasets have different forms: discrete, continuous and symbolic, with significantly varying resolution and ranges. Most pattern classification methods are not able to process data in such a format. Hence, preprocessing is required.

Symbolic_valued features, such as *protocol_type* (3 different symbols), *service* (70 different symbols) and *flag* (11 different symbols) are mapped to integer values ranging from 0 to $N-1$, where N is the number of symbols. Continuous features having smaller integer value ranges like *wrong_fragment* [0,3], *urgent* [0,14], *hot* [0,101], *num_failed_logins* [0,5], *num_compromised* [0,9], *num_root* [0,7468], *num_file_creations* [0,100], *num_shells* [0,5], *num_access_files* [0,9], *count* [0,511], *srv_count* [0,511], *dst_host_count* [0,255], *dst_host_srv_count* [0,255] 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* [0,58329], *src_bytes* [0,1.3billion] and *dst_bytes* [0,1.3billion], to reduce the ranges to [0,4.77] and [0,9.11], respectively. 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].

STATISTICAL ANALYSIS FOR FEATURE RANKING

Chi-square Analysis: Chi-square is a non-parametric test of statistical significance for bivariate tabular analysis. In this way, consider a set of k measurements of size: $\{x_1, x_2, \dots, x_k\}$, where x_k is the size of k^{th} measurement. They are supposed to be "normally" distributed and their mean and standard deviation are assumed to be μ and σ , respectively. The Chi-square value is obtained as follows:

$$\chi^2 = \sum_{i=1}^k \frac{(x_i - \mu)^2}{\sigma^2} \quad (1)$$

Logistic Regression: Logistic regression is part of a category of statistical models, called generalized linear models. Logistic regression allows one to predict a discrete outcome, such as group membership, from a set of variables that may be continuous, discrete, or mix of them. Logistic regression method is used for bivariate analysis of data [16].

Feature Ranking: In this paper, logistic regression is used to rank the features based on the Chi-square values for different selected subsets using best subset selection model. Higher the Chi-square value, higher is the ranking. The 41 features are ranked for different subsets with the subset size ranging from 1 to 41. The subset selection model gives us a complete analysis for the ranking of features. The ranking results of the Chi-square test on KDD dataset are reported in Table 3.

SIMULATED ATTACK RECOGNIZERS AND EMPIRICAL RESULTS

In this paper, various optimized structures of MLP and Elman, as static and dynamic neural networks, are employed in different experiments to investigate the effects of feature reduction on the classification rate and training time of neural attack recognizers.

Four experiments are performed by using the 38, 25, 20 and 15 more important features as the input of neural classifiers, respectively (Table 3). The number of

Table 3: Chi-square values of input features with respect to the attack class

Feature	Probe	DoS	U2R	R2L
dst_host_diff_srv_rate	3686.276	1334.816	2531.961	1114.092
rerror_rate	2734.528	1016.257	613.386	1016.537
dst_host_srv_rerror_rate	2707.659	967.869	301.063	586.241
srv_rerror_rate	2515.679	805.548	244.916	583.339
dst_host_rerror_rate	2251.962	732.804	207.829	560.594
diff_srv_rate	1228.261	551.745	39.879	350.122
dst_host_same_srv_rate	793.251	449.233	39.160	311.149
service	588.682	438.755	36.736	249.510
dst_host_srv_count	546.117	433.031	32.613	239.163
logged in	427.171	363.640	25.147	141.752
dst_host_srv_diff_host_rate	422.292	353.490	25.008	141.309
srv_count	123.396	344.882	15.455	141.222
same_srv_rate	91.771	336.854	15.334	126.073
protocol type	84.579	328.688	10.681	125.016
num_compromised	70.402	308.367	10.256	116.023
wrong_fragment	68.621	275.583	6.350	99.830
dst_host_same_src_port_rate	65.364	247.046	6.251	78.342
hot	33.863	240.337	6.243	53.059
srv_serror_rate	20.310	188.873	6.206	46.793
dst_host_srv_serror_rate	19.602	129.101	6.193	45.480
is_guest_login	18.219	121.415	3.790	37.077
serror_rate	17.718	102.236	3.373	33.889
src_bytes	8.265	101.479	3.360	27.668
duration	7.639	52.407	2.928	26.065
dst_host_serror_rate	7.385	45.386	2.674	25.980
count	2.232	35.564	2.606	15.351
root_shell	2.077	27.393	2.534	11.075
num_failed_logins	2.059	27.166	2.372	10.737
dst_bytes	1.248	10.068	2.084	8.121
land	1.200	9.045	1.589	8.046
srv_diff_host_rate	0.815	7.675	1.508	7.879
num_access_files	0.763	7.564	0.892	5.662
num_file_creations	0.573	6.404	0.393	3.202
num_shells	0.486	6.095	0.059	1.370
num_root	0.462	2.624	0.012	0.561
urgent	0.196	2.586	0.011	0.047
su_attempted	0.109	1.437	0.011	0.014
dst_host_count	0.025	0.900	0.005	0.006
flag	0.000	0.000	0.000	0.000
is_host_login	0.000	0.000	0.000	0.000
num_outbound_cmds	0.000	0.000	0.000	0.000

Table 4: Number of hidden-layer neurons of MLP and Elman in different experiments

Number of selected features	38	25	20	15
Number of hidden-layer neurons of MLP	38	33	30	25
Number of hidden-layer neurons of Elman	38	20	18	15

Table 5: Cost matrix values for KDD 99

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

hidden-layer neurons for MLP and Elman network in each of these experiments is selected by using the genetic algorithm (GA) reported in [18] (Table 4).

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. Another metric that is used here is the classification rate. Classification rate for each class of data is defined as the ratio between the number of test instances correctly classified and the total number of test instances of this class.

For the purpose of classifier algorithm evaluation, another comparative measure is defined which is cost per example (CPE) [19]. CPE is calculated using the following formula:

$$CPE = \frac{1}{N_T} \sum_{i=1}^m \sum_{j=1}^m CM(i,j).C(i,j) \tag{2}$$

Where CM and C are confusion matrix and cost matrix, respectively. N_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 5 [17].

The confusion matrix and training time, when using MLP and Elman attack recognizers with different selected features as their input, are reported in Table 6 and Table 7, respectively. In this way, the detection rate for each case is calculated and shown in Figure 1. As shown in Figure 1, MLP performs better than Elman classifier in attack recognition. In addition, better detection rate is achieved when the 15 more important features are used as the inputs of MLP attack recognizer.

The training time of each neural attack recognizer is also shown in Figure 2. As shown in Figure 2, Elman classifiers have lower training time.

Table 6: Confusion matrix and training time for MLP attack recognizer with different selected input features

Number of input features (Abbreviated name of classifier)	Predicted					
	Actual	Normal	Probe	DoS	U2R	R2L
38 (ML-38)	Normal	6037	14	8	0	0
	Probe	48	315	54	0	0
	DoS	50	27	22908	0	0
	U2R	2	2	1	1	1
	R2L	1095	5	0	1	534
Training time = 10191 seconds						
25 (ML-25)	Normal	6045	13	1	0	0
	Probe	8	385	24	0	0
	DoS	53	33	22896	0	3
	U2R	2	3	1	0	1
	R2L	1114	3	0	0	518
Training time=6292 seconds						
20 (ML-20)	Normal	6045	12	2	0	0
	Probe	32	365	15	0	5
	DoS	49	38	22898	0	0
	U2R	2	2	2	0	1
	R2L	55	4	0	0	1576
Training time=4983 seconds						
15 (ML-15)	Normal	6047	11	1	0	0
	Probe	16	357	27	0	17
	DoS	19	10	22914	0	42
	U2R	3	1	2	0	1
	R2L	18	0	0	0	1617
Training time= 3790 seconds						

Table 7: Confusion matrix and training time for Elman attack recognizer with different selected input features

Number of input features (Abbreviated name of classifier)	Predicted					
	Actual	Normal	Probe	DoS	U2R	R2L
38 (EL-38)	Normal	5949	8	96	0	6
	Probe	138	221	44	13	1
	DoS	3975	0	19001	3	6
	U2R	3	1	0	1	2
	R2L	1473	86	0	0	76
Training time = 2743 seconds						
25 (EL-25)	Normal	5948	10	95	0	6
	Probe	68	251	96	1	1
	DoS	3460	9	19488	0	28
	U2R	2	1	3	0	1
	R2L	1572	3	0	0	60
Training time= 1740 seconds						
20 (EL-20)	Normal	5948	9	102	0	0
	Probe	89	249	78	1	0
	DoS	3568	8	19408	0	1
	U2R	4	1	1	0	1
	R2L	1568	0	0	0	67
Training time= 1583 seconds						
15 (EL-15)	Normal	5969	3	87	0	0
	Probe	56	274	87	0	0
	DoS	3485	3	19497	0	0
	U2R	2	0	4	0	1
	R2L	1502	1	0	0	132
Training time= 1313 seconds						

Table 8: Performance of proposed neural attack recognizers as compared to other machine learning models

Model	Classification rate						DR	FAR	CPE
	Normal	Probe	DoS	U2R	R2L				
Winner of KDD in 2000 [20]	99.5	83.3	97.1	13.2	8.4	91.8	0.6	0.2331	
PNrule [19]	99.5	73.2	96.9	6.6	10.7	91.1	0.4	0.2371	
Runner up of KDD in 2000 [21]	99.4	84.5	97.5	11.8	7.3	91.5	0.6	0.2356	
ESC-IDS [22]	98.2	84.1	99.5	14.1	31.5	95.3	1.9	0.1579	
ML-38	99.6	75.5	99.7	14.3	32.7	94.9	0.36	0.1517	
ML-15	99.8	85.6	99.7	0.0	98.9	99.4	0.02	0.0108	

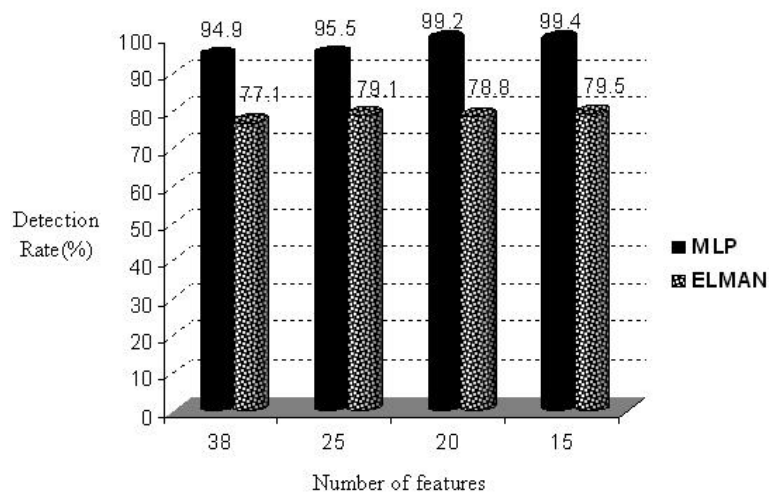


Fig. 1: Detection rate of neural attack recognizers with reduced-size features

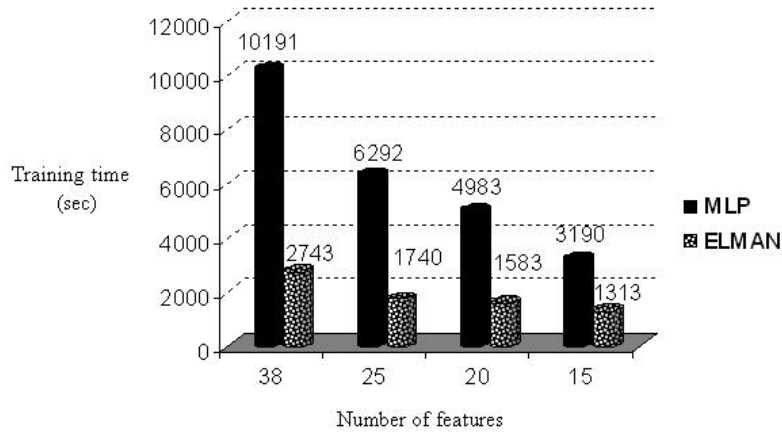


Fig. 2: Training time of neural attack recognizers with reduced-size features

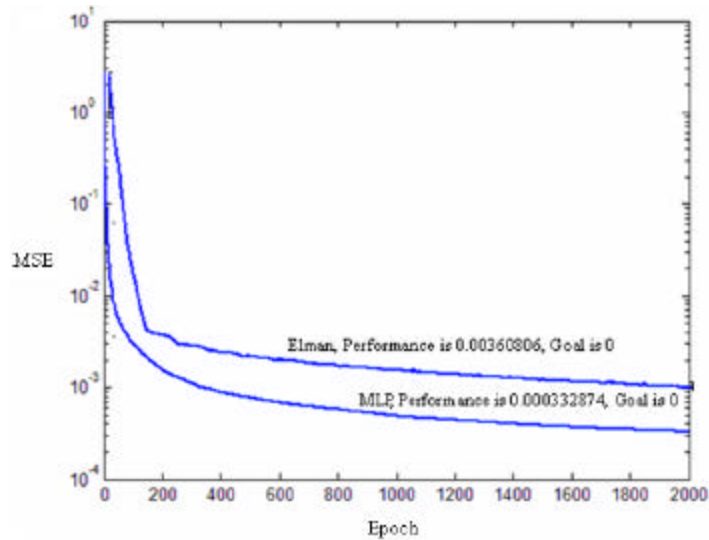


Fig. 3: Error performance of neural attack recognizers with 15 selected input features

The error performance of MLP and Elman attack recognizers, in terms of mean squared error (MSE), with the 15 more important selected features as their input is shown in Figure 3. As shown in Figure 3, MLP offers better error performance as compared to Elman neural network.

The performance of MLP attack recognizers with 38 and 15 selected input features, as sample cases, has been compared with some other machine learning methods in terms of classification rate, DR, FAR and CPE (Table 8). As shown in Table 8, the MLP attack recognizer with the 15 more important features as its input has higher classification rate for Normal, Probe, DoS and R2L classes, as compared to systems reported in [19-22]. To have a fast attack recognizer by using a reduced-size training dataset (Table 1), as compared to 10% KDD training dataset that was used in [19-22], we experimented poor performance in U2R attack recognition that had only 6 training samples in our dataset. Although, the MLP classifier with 38 selected input features has acceptable performance in U2R attack recognition, as compared to other systems in Table 8.

CONCLUSIONS

Due to importance of feature selection in intrusion detection field, logistic regression was used in this paper to rank the features based on the Chi-square values for different subsets. To have lightweight IDS [23] and also to reduce the false alarms [24] in the state-of-art systems, we investigate the effects of feature reduction on classification rate and training time of neural attack recognizers by employing various optimized structures of MLP and Elman neural networks.

Empirical results showed that the MLP with 15 selected input features had higher recognition rates for Normal, Probe, DoS and R2L attack classes, as compared to some other machine learning methods. This classifier performed better in terms of detection rate, false alarm rate and cost per example, as well.

REFERENCES

1. Wu, S.X. and W. Banzhaf, 2010. The Use of Computational Intelligence in Intrusion Detection Systems: A Review. *Applied Soft Computing*, 10: 1-35.
2. Tsai, C.F., Y.F. Hsu, C.Y. Lin and W.Y. Lin, 2009. Intrusion Detection by Machine Learning: A Review. *Expert Systems with Applications*, 36: 11994-12000.
3. Zhou, C.V., C. Leckie and S. Karunasekera, 2010. A Survey of Coordinated Attacks and Collaborative Intrusion Detection. *Computers and Security*, 29: 124-140.
4. Garcia-Teodoro, P., J. Diaz-Verdejo, G. Macia-Fernandez and E. Vazquez, 2009. Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges. *Computers and Security*, 28: 18-28.
5. Shon, T. and J. Moon, 2007. A Hybrid Machine Learning Approach to Network Anomaly Detection. *Information Sci.*, 177: 3799-3821.
6. Beghdad, R., 2008. Critical Study of Neural Networks in Detecting Intrusions. *Computers and Security*, 27: 168-175.
7. Sheikhan, M. and A.A. Sha'bani, 2009. Fast Neural Intrusion Detection System Based on Hidden Weight Optimization Algorithm and Feature Selection. *World Applied Sciences J.*, 7(Special Issue of Computer and IT): 45-53.
8. 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.
9. 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.
10. Mukkamala, S., G. Janoski and A.H. Sung, 2002. Intrusion Detection Using Neural Networks and Support Vector Machines. In the Proceedings of the International Joint Conference on Neural Networks, pp: 1702-1707.
11. Abadeh, M.S., J. Habibi and C. Lucas, 2005. Intrusion Detection Using a Fuzzy Genetic-Based Learning Algorithm. *Network and Computer Application*, 30: 414-428.
12. Tajbakhsh, A., M. Rahmati and A. Mirzaei, 2009. Intrusion Detection Using Fuzzy Association Rules. *Applied Soft Computing*, 9: 462-469.
13. Sheikhan, M. and Z. Jadidi, 2009. Misuse Detection Using Hybrid of Association Rule Mining and Connectionist Modeling. *World Applied Sciences J.*, 7(Special Issue of Computer and IT): 31-37.
14. Ali Aydin, M., A. Halim Zaim and K. Gokhan Ceylan, 2009. A Hybrid Intrusion Detection System Design for Computer Network Security. *Computers and Electrical Engineering*, 35: 517-526.

15. Tong, X., Z. Wang and H. Yu, 2009. A Research Using Hybrid RBF/Elman Neural Networks for Intrusion Detection System Secure Model. *Computer Physics Communications*, 180: 1795-1801.
16. 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.
17. K.D.D., 1999. Cup Competition (Available on <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>).
18. Sheikhan, M. and B. Movaghar, 2009. Exchange Rate Prediction Using an Evolutionary Connectionist Model. *World Applied Sciences J.*, 7(Special Issue of Computer and IT): 8-16.
19. 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.
20. Pfahringer, B., 2000. Winning the KDD 99 Classification Cup: Bagged Boosting. *J. SIGKDD Explorations*, 1: 65-66.
21. Levin, I., 2000. KDD Classifier Learning Contest: LLSoft's Results Overview. *J. SIGKDD Explorations*, 1: 67-75.
22. Nadjaran Toosi, A. and M. Kahani, 2007. A Novel Soft Computing Model Using Adaptive Neuro-Fuzzy Inference System for Intrusion Detection. In the Proceedings of the IEEE International Conference on Networking, Sensing and Control, pp: 834-839.
23. Chen, C.M., Y.L. Chen and H.C. Lin, 2010. An Efficient Network Intrusion Detection. *Computer Networks*, 33: 477-484.
24. Spathoulas, G.P. and S.K. Katsikas, 2010. Reducing False Positives in Intrusion Detection Systems. *Computers and Security*, 29: 35-44.