Application of Partial-Connected Dynamic and GA-Optimized Neural Networks to Misuse Detection Using Categorized and Ranked Input Features

M. Sheikhan[1], Z. Jadidi[2], A. Farrokhi[3]
1- Associate Professor, Department of Communication Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.
Email: msheikhn@azad.ac.ir (Corresponding author)
2- MSc., Department of Electronic Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.
Email: z_jadidi@azad.ac.ir
3- Assistant Professor, Department of Electronic Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.
Email: ali_farrokhi@azad.ac.ir

Received: June 2010 Revised: October 2010 Accepted: February 2011

ABSTRACT:
The number of attacks in computer networks has grown extensively, and many new intrusive methods have been appeared. Intrusion detection is known as an effective method to secure the information and communication systems. In this paper, the performance of Elman and partial-connected dynamic neural network (PCDNN) architectures are investigated for misuse detection in computer networks. To select the most significant features, logistic regression is also used to rank the input features of mentioned neural networks (NNs) based on the Chi-square values for different selected subsets in this work. In addition, genetic algorithm (GA) is used as an optimization search scheme to determine the sub-optimal architecture of investigated NNs with selected input features. International knowledge discovery and data mining group (KDD) dataset is used for training and test of the mentioned models in this study. The features of KDD data are categorized as basic, content, time-based traffic, and host-based traffic features. Empirical results show that PCDNN with selected input features and categorized input connections offers better detection rate (DR) among the investigated models. The mentioned NN also performs better in terms of cost per example (CPE) when compared to other proposed models in this study. False alarm rate (FAR) of the PCDNN with selected input features and categorized input connections is better than other proposed models, as well.

KEYWORDS: Feature categorization, feature ranking, misuse detection, dynamic neural networks.

1. INTRODUCTION

One way of dealing with suspicious activities within a network is based on using intrusion detection system (IDS). An IDS monitors the activities of environment and decides on its anomaly. Based on the information source, there are two kinds of IDS: network-based [1] and host-based [2]. Monitoring the data exchanged between computers is performed in network-based IDS, and host-based intrusion detection systems are served on host computer. Based on the type of analyzing events, two kinds of IDS have been developed: anomaly-based [3, 4] and misuse-based [5, 6]. In anomaly-based IDS, the activities that vary from established patterns for users are detected. On the other hand, in misuse-based IDS user's activities with the known behaviors of attackers are compared.

The classification of anomaly-based detection techniques are as: knowledge-based [3], statistical-based [7], and machine learning (e.g. artificial neural networks (ANNs) [1, 4], Bayesian networks [8], Markov models [2], genetic algorithms [9], clustering and outlier detection [10], fuzzy logic [11, 12] and hybrid systems [13]).

Similarly, the detection techniques in misuse-based IDS are as: knowledge-based [14, 15], statistical-based [16], and machine learning (e.g. ANNs [6, 16-22], Bayesian networks [23], genetic algorithms [24], fuzzy logic [11], decision trees [25, 26], clustering [27] and hybrid systems [6, 28-32]).

In this paper, the performance of different structures of dynamic neural models (such as Elman with one and two hidden layers and partial-connected dynamic neural network (PCDNN) architectures) are investigated for misuse detection in computer networks. Since, feature selection and ranking is an important issue in intrusion detection, logistic regression is used in this work to rank the features based on the Chi-square values for different selected subsets using best subset selection model [22, 33]. The effects of feature reduction on classification rate and training time of mentioned attack recognizers are investigated in this paper when employing genetic algorithm (GA)-optimized structures.

International knowledge discovery and data mining group (KDD) dataset [34] is used for training and test of the mentioned models in this study. Each connection in KDD is characterized by 41 features and a label which specifies the status of connection records (normal or a specific attack type). These features can be grouped into four categories: basic features, content features, time-based traffic features, and host-based traffic features. To reduce the size and computational complexity of Elman and PCDNN-based IDS, the connections to hidden layer are connected partially based on the mentioned four feature categories.

The remainder of this paper is organized as follows. Section 2 provides the KDD dataset details. The preprocessing procedure of features is discussed in Section 3. As part of the feature selection experiments, the statistical analysis is presented in Section 4. The application of GA in optimization of NN's architecture is reviewed in Section 5. The simulations and experimental results are reported in Section 6. Conclusions are also drawn in Section 7.

2. KDD DATASET

In 1999, recorded network traffic from the Defense Advanced Research Project Agency (DARPA) dataset was summarized into network connections with 41 features per connection. This formed the benchmark provided by the International knowledge discovery and data mining group (KDD). The KDD dataset consists of three components: “10% KDD”, “Corrected KDD” and “Whole KDD” [34]. There are four main categories of attacks given in the KDD 99: denial-of-service (DoS), probe, remote-to-local (R2L) and user-to-root (U2R). There are multiple attack types for each main attack category, as well (Table 1).

Table 1. 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)
U2R	buffer_overflow (30), rootkit (10), loadmodule (9), perl (3)
R2L	warezclient (1020), guess_passwd (53), warezmaster (20), imap (12), ftp_write (8), multihop (7), phf (4), spy (2)

The analysis in this paper is performed on the "10% KDD" dataset. It is reminded that each connection in KDD is characterized by 41 features. As mentioned earlier, these features are grouped into four categories: basic features, content features, time-based traffic features and host-based traffic features.

Table 2. Description of basic features in KDD dataset

Feature	Description
duration	Duration of the connection (in seconds)
protocol_type	Type of the connection protocol
service	Service on the destination
flag	Status flag of the connection
src_bytes	Number of bytes sent from source to destination
dst_bytes	Number of bytes sent from destination to source
land	1 if connection is from/to the same host/port; 0 otherwise
wrong_fragment	Number of wrong fragments
urgent	Number of urgent packets

Table 3. Description of content features in KDD dataset

Feature	Description
hot	Number of “hot” indicators
num_failed_logins	Number of failed logins
logged_in	1 if successfully logged in; 0 otherwise
num_compromised	Number of “compromised” conditions
root_shell	1 if root shell is obtained; 0 otherwise
su_attempted	1 if “su root” command attempted; 0 otherwise
num_root	Number of “root” accesses
num_file_creations	Number of file creation operations
num_shells	Number of shell prompts
num_access_files	Number of operations on access control files
num_outbound_cmds	Number of outbound commands in a FTP session
is_host_login	1 if the login belongs to the “hot” list; 0 otherwise
is_guest_login	1 if the login is a “guest” login; 0 otherwise

Basic features can be derived from packet headers without inspecting the payload (Table 2). In the content features, domain knowledge is used to assess the payload of the original transmission control protocol (TCP) packets (Table 3). Time-based traffic features are designed to capture properties that mature over a two-second temporal window (Table 4). Host-based traffic features utilize a historical window estimated over the number of connections, instead of time. Therefore, they are designed to assess attacks which span in intervals longer than 2 seconds (Table 5).

Table 4. Description of time-based traffic features in KDD dataset

Feature	Description
count	Number of connections to the same host as the current connection in the past two seconds
srv_count	Number of connections to the same service as the current connection in the past two seconds
serror_rate	Percent of connections that have “SYN” errors (same-host connections)
srv_serror_rate	Percent of connections that have “SYN” errors (same-service connections)
rerror_rate	Percent of connections that have “REJ” errors (same-host connections)
srv_rerror_rate	Percent of connections that have “REJ” errors (same-service connections)
same_srv_rate	Percent of connections to the same service
diff_srv_rate	Percent of connections to different services
srv_diff_host_rate	Percent of connections to different hosts

3. PREPROCESSING PROCEDURE

It is noted that 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 (with 3 different symbols), service (with 70 different symbols), and flag (with 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 [0,1] range.

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 [0,1] range.

Table 5. Description of host-based traffic features in KDD dataset

Feature	Description
dst_host_count	Number of connections having the same destination host
dst_host_srv_count	Number of connections having the same destination host and using the same service
dst_host_same_srv_rate	Percent of connections having the same destination host and using the same service
dst_host_diff_srv_rate	Percent of different services on the current host
dst_host_same_src_port_rate	Percent of connections to the current host having the same src port
dst_host_srv_diff_host_ rate	Percent of connections to the same service coming from different hosts
dst_host_serror_rate	Percent of connections to the current host that have an S0 error
dst_host_srv_serror_rate	Percent of connections to the current host and specified service that have an S0 error
dst_host_rerror_rate	Percent of connections to the current host that have an RST error
dst_host_srv_rerror_rate	Percent of connections to the current host and specified service that have an RST error

4. FEATURE RANKING BASED ON STATISTICAL ANALYSIS

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 [33].

It is noted that logistic regression is a generalized linear statistical model. 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 [33].

Also, Chi-square is a non-parametric test of statistical significance for bivariate tabular analysis. In this way, consider a set of k measurements $\{x_1,x_2,...,x_k\}$. If they are normally distributed and their mean and standard deviation are μ and σ, respectively then the Chi-square value is calculated as follows:

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

In this way, higher values of Chi-square results in higher 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. For example, the ranking results of the Chi-square test on KDD dataset are reported for the 15 most significant features in Table 6.

Table 6. Chi-square values of 15 most significant features with respect to the attack class

Feature	DoS	Probe	U2R	R2L
dst_host_diff_srv _rate	1334.8	3686.3	2532.0	1114.1
rerror_rate	1016.3	2734.5	613.4	1016.5
dst_host_srv_rerror_rate	967. 9	2707.7	301.1	586.2
srv_rerror_rate	805.5	2515.7	244.9	583.3
dst_host_rerror _rate	732.8	2251.0	207.8	560.6
diff_srv_rate	551.7	1228.3	39. 9	350.1
dst_host_same_srv_rate	449.2	793.3	39.2	311.1
service	438.8	588.7	36.7	249.5
dst_host_srv_count	433.0	546.1	32.6	239.2
logged in	363.6	427.2	25.1	141.8
dst_host_srv_diff _host_rate	353.5	422.3	25.0	141.3
srv_count	344.9	123.4	15. 5	141.2
same_srv_rate	336.9	91.8	15.3	126.1
protocol type	328.7	84.6	10.7	125.0
num_compromised	308.4	70.4	10.3	116.0

5. GENETIC ALGORITHM OPTIMIZATION PROCESS

Genetic algorithm can be used as an optimization search scheme to determine the optimal or sub-optimal architecture and parameters of a neural network [35].

Genetic algorithm improves the performance of NNs by selecting the best input features, optimization of network parameters (e.g. learning rate, momentum coefficient, number of hidden layers, number of nodes in hidden layer, and initial weights), modification of nodes' activation function, and determination of weights. In this work, GA is used for determining the optimum number of hidden layer nodes of Elman with selected input features [36].

The genetic algorithm optimization process is described in the following procedure:

1. Randomize population.
2. Evaluate the fitness function (1/(1+MSE)) for each individual in the population.
3. Select the first two individuals with the highest fitness values and copy directly to the next generation without any genetic operation.
4. Select the remaining individuals in the current generation and apply crossover and mutation genetic operations accordingly to reproduce the individuals in the next generation.
5. Repeat from the second step until all individuals in population meet the convergence criteria.
6. Decode the converged individuals in the final generation and obtain the optimized parameters.

6. SIMULATION AND EXPERIMENTAL RESULTS

As mentioned earlier, the performance of different structures of dynamic neural models (such as Elman with one and two hidden layers and partial-connected dynamic neural network (PCDNN) architectures) as misuse-based IDSs are investigated in this paper (Fig. 1 to Fig. 3). As shown in Figs. 1 and 2, the numbers of nodes at different layers of Elman models with one and two hidden layers are set to 41-10-5 and 41-20-10-5 arrangements, through various try and error experiments, respectively. Each model has five output neurons (representing four attack types and normal class).

As it can be seen in Fig. 3, the connections between 41 input nodes and hidden layer nodes in PCDNN are based on the categorization of features. It is noted that in our simulations the same categorization is applied to the inputs of Elman NNs, depicted in Figs. 1 and 2.

Also, the effects of feature reduction on the performance of Elman and PCDNN attack recognizers are investigated in this paper by applying only the 15 selected features, listed in Table 6, to the mentioned NNs. Genetic algorithm (GA) is employed to determine the optimum number of hidden layer nodes.

In this work, 49402 records from "10% KDD" dataset and 31104 records from "Corrected KDD" dataset are used as training and test datasets, respectively (Table 7). These sets have similar distribution, except of U2R test samples, for different categories of attacks as corresponding KDD datasets.

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.

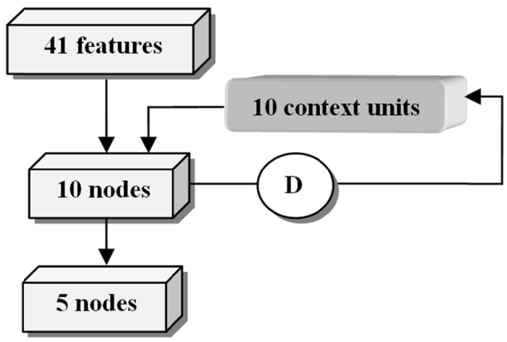

Fig. 1. Fully-connected Elman misuse detector with single hidden layer

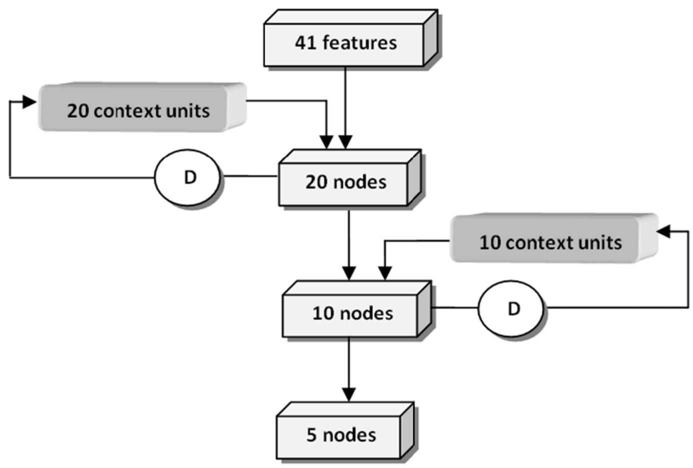

Fig. 2. Fully-connected Elman misuse detector with two hidden layers

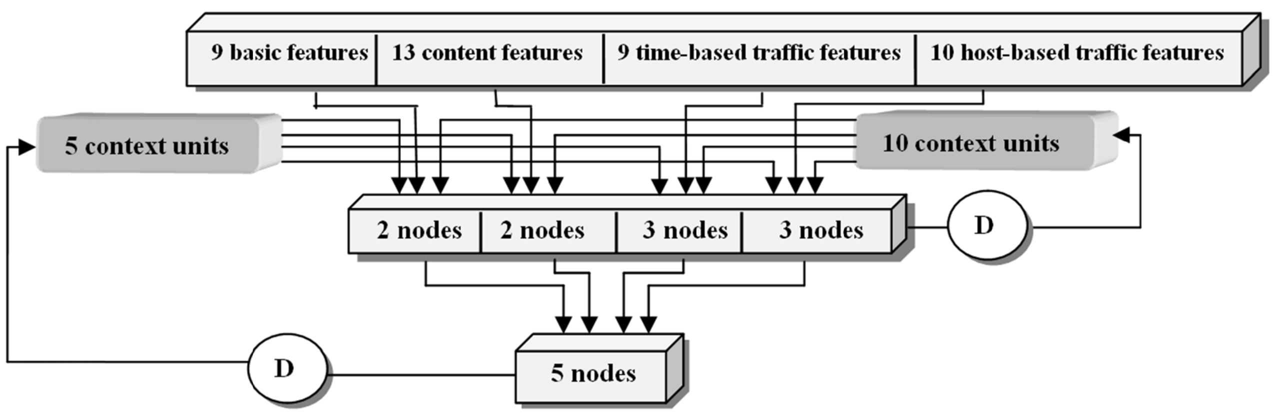

Fig. 3. PCDNN misuse detector

Table 7. Size of the training and test datasets

Class	Number of training samples	Number of test samples
Normal	9727	6059
DoS	39145	22985
Probe	411	417
U2R	6	24
R2L	113	1619

Another comparative measure is cost per example (CPE) [37]. CPE is calculated using the following formula:

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

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 8 [34].

The confusion matrix of the PCDNN model with categorized input features is reported in Table 9. The confusion matrices of single-hidden layer and two-hidden layers Elman-based neural classifiers with 41 categorized input features are reported in Tables 10 and 11, respectively. The confusion matrices of single-hidden layer and two-hidden layers Elman-based neural classifiers with the 15 most important selected input features are reported in Tables 12 and 13, as well. Finally, the confusion matrix of the PCDNN model with 15 selected features and categorized input connections is reported in Table 14.

The number of hidden-layer neurons for PCDNN model in the experiments with selected input features is determined by using the genetic algorithm (GA) and obtained as 10. The training times of the investigated classifiers are reported in Table 15.

For the selected input features experiments, the error performance in terms of mean squared error (MSE) is shown in Fig. 4. As shown in Fig. 4, PCDNN offers better error performance when compared to Elman neural classifier.

The performance of the proposed models has been compared with some other machine learning methods, in terms of detection rate (DR), false alarm rate (FAR) and cost per example (CPE), as well (Table 16). As shown in Table 16, PCDNN with 15 selected input features and categorized input connections offers better detection rate (DR) among the investigated models. The mentioned model also performs better in terms of cost per example (CPE) when compared to other proposed models in this study. False alarm rate (FAR) of the PCDNN with selected input features is better than other proposed models, as well.

Also, the classification rate (CR) of different attacks and FAR of the PCDNN model, as the superior one among the investigated models in this work, and some other IDS algorithms developed in recent years are reported in Table 17. As can be seen, the DoS and R2L classification rates of PCDNN with selected input features and categorized input connections have high ranks, as compared to other models. It is noted that R2L is a hard-detectable attack [48] and the mentioned ANN model offers this performance with a reduced-size of neural net connections and computational complexity.

7. CONCLUSION

In this paper, the performance of different structures of Elman and partial-connected dynamic neural network (PCDNN) models has been investigated for misuse detection. The most significant features have been selected by using logistic regression to rank the input features of NNs based on the Chi-square values for different selected subsets. In addition, genetic algorithm (GA) has been used to determine the sub-optimal architecture of Elman NN with selected input features. Empirical results have shown that PCDNN with selected input features and categorized input features offers better detection rate (DR) among the investigated models. The mentioned model also performs better in terms of cost per example (CPE) when compared to other proposed models in this study. False alarm rate (FAR) of PCDNN is better than other proposed models, as well.

Table 8. Cost matrix values for KDD

Actual \ Predicted	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

Table 9. Confusion matrix of PCDNN model with 41 categorized input features

Actual \ Predicted	DoS	Probe	R2L	U2R	Normal
DoS	19073	9	0	0	3903
Probe	43	250	0	0	124
R2L	0	4	10	0	1605
U2R	4	1	0	0	19
Normal	96	15	0	0	5948

Table 10. Confusion matrix of single-hidden layer Elman model with 41 categorized input features

Actual \ Predicted	DoS	Probe	R2L	U2R	Normal
DoS	18939	0	0	0	4046
Probe	63	200	0	0	154
R2L	0	0	4	0	1615
U2R	4	0	0	0	20
Normal	101	7	0	0	5951

Table 11. Confusion matrix of two-hidden layer Elman model with 41 categorized input features

Actual \ Predicted	DoS	Probe	R2L	U2R	Normal
DoS	21734	0	0	0	1251
Probe	18	240	2	0	157
R2L	0	3	45	0	1571
U2R	0	0	1	0	23
Normal	88	7	0	0	5964

Table 12. Confusion matrix of single-hidden layer Elman model with 15 selected input features

Actual \ Predicted	DoS	Probe	R2L	U2R	Normal
DoS	19615	0	0	0	3370
Probe	55	236	0	0	126
R2L	0	3	23	0	1593
U2R	11	4	0	0	9
Normal	101	6	0	0	5952

Table 13. Confusion matrix of two-hidden layer Elman model with 15 selected input features

Actual \ Predicted	DoS	Probe	R2L	U2R	Normal
DoS	19497	3	0	0	3458
Probe	87	274	0	0	56
R2L	0	7	129	0	1483
U2R	4	0	3	0	17
Normal	87	3	0	0	5969

Table 14. Confusion matrix of PCDNN model with 15 selected input features and categorized input connections

Actual \ Predicted	DoS	Probe	R2L	U2R	Normal
DoS	22973	12	0	0	0
Probe	141	267	0	0	9
R2L	0	11	1607	0	1
U2R	9	15	0	0	0
Normal	0	10	0	0	6049

Table 15. Training time of different simulated dynamic IDS models

Model	Training time (s)
PCDNN with 41 categorized input features	653
Single-hidden layer Elman with 41 categorized input features	283
Two-hidden layer Elman with 41 categorized input features	723
Single-hidden layer Elman with 15 selected input features	243
Two-hidden layer Elman with 15 selected input features	1303
PCDNN with 15 selected input features and categorized input connections	617

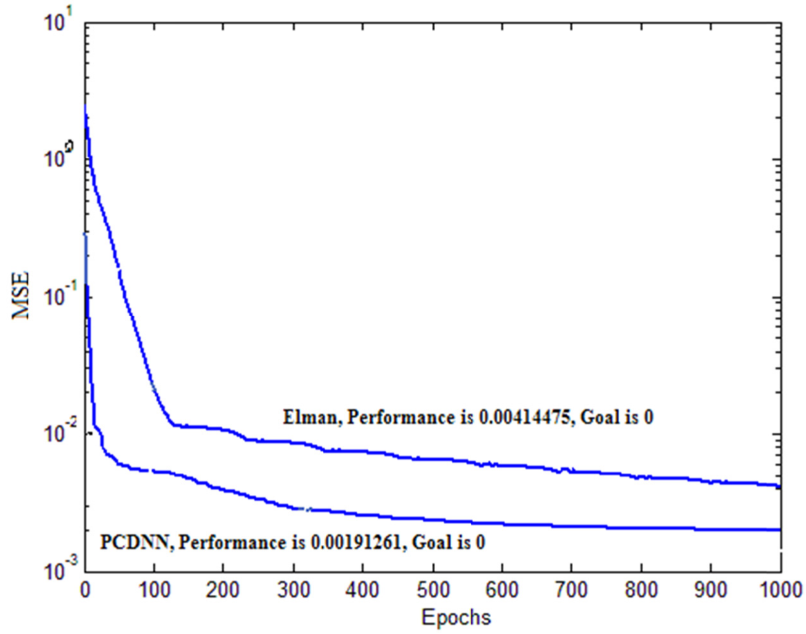

Fig. 4. Error performance of Elman and PCDNN IDSs with selected input features

Table 16. Performance comparison of models in intrusion detection

Model	CPE	DR	FAR
PCDNN with 15 selected input features and categorized input connections (proposed in this study)	0.0124	99.2	0.17
PCDNN with 41 categorized input features (proposed in this study)	0.4735	77.2	1.83
Single-hidden layer Elman with 41 categorized input features (simulated in this study)	0.4858	76.4	1.78
Two-hidden layer Elman with 41 categorized input features (simulated in this study)	0.2972	87.9	1.57
Single-hidden layer Elman with 15 selected input features (simulated in this study)	0.4379	79.4	1.77
Two-hidden layer Elman with 15 selected input features (simulated in this study)	0.4288	79.5	1.49
Winner of KDD in 2000 [25]	0.2331	91.8	0.6
Runner up of KDD in 2000 [26]	0.2356	91.5	0.6
PNrule [37]	0.2371	91.1	0.4
MLP [5]	NR*	73.0	27.03
Radial Basis Function (RBF) ANN [5]	NR*	96.1	3.85
Self Organizing Feature Map (SOFM) ANN [5]	NR*	71.6	28.37
Jordan ANN [5]	NR*	62.9	37.09
RNN [5]	NR*	73.1	26.85
Data mining [14]	NR*	70-90	2
Clustering [14]	NR*	93	10
K- Nearest Neighbor [14]	NR*	91	8
Support Vector Machine (SVM) [14]	NR*	98	10
Hierarchical Self Organizing Map (H-SOM) ANN [14]	NR*	90-91.5	7.6-14.5
Fuzzy Association Rules [30]	NR*	91	3.34

*: Not Reported

Table 17. Performance of proposed model and other machine learning methods in terms of CR and FAR

Method	Metric (%)	DoS	Probe	U2R	R2L
MLP ANN [15]	CR	97.2	88.7	13.2	5.6
	FAR	0.3	0.4	0.1	0.1
Gaussian Bayes decision algorithm [15]	CR	82.4	90.2	22.8	0.1
	FAR	0.9	11.3	0.5	0.1
K-means clustering algorithm [15]	CR	97.3	87.6	29.8	6.4
	FAR	0.4	2.6	0.4	0.1
Nearest clustering algorithm [15, 38]	CR	97.1	88.8	2.2	3.4
	FAR	0.3	0.5	0.1	0.1
Incremental RBF ANN [15, 39]	CR	73.0	93.2	6.1	5.9
	FAR	0.2	18.8	0.4	0.3
Leader algorithm [15, 40]	CR	97.2	83.8	8.3	1.0
	FAR	0.3	0.3	0.3	0.1
Hyper sphere algorithm [15, 41]	CR	97.2	83.8	8.3	1.0
	FAR	0.3	0.3	0.3	0.1
Fuzzy ARTMAP ANN [15, 42]	CR	97.0	77.2	6.1	3.7
	FAR	0.3	0.2	0.1	0.1
C4.5 decision tree algorithm [15]	CR	97.0	80.8	1.8	4.6
	FAR	0.3	0.7	0.1	0.1
Boosted modified probabilistic neural network (BMPNN) [43]	CR	96.8	96.1	38.6	48.5
	FAR	0.3	0.2	0.0	0.1
Hybrid flexible neural tree [44]	CR	98.8	98.4	99.7	99.1
	FAR	0.6	1.4	0.2	0.8
Back-propagation neural network (BPN) [45]	CR	99.6	91.1	0.0	0.0
	FAR	0.4	9.0	0.0	0.0
Gaussian mixture [46]	CR	88.2	93.0	22.8	9.6
	FAR	0.2	72.3	45.8	1.0
RBF ANN [46]	CR	75.1	91.3	7.0	5.6
	FAR	0.3	88.1	61.9	0.9
Binary tree algorithm [46]	CR	96.5	77.9	13.6	0.4
	FAR	3.6	52.4	35.4	7.7
LAMSTAR neural network [46]	CR	99.2	98.5	28.9	41.2
	FAR	0.5	25.0	10.0	0.1
Wavelet neural network [47]	CR	95.5	91.3	54.7	80.0
	FAR	4.5	8.7	45.3	20.0
PCDNN with selected input features and categorized input connections	CR	99.9	64.0	0.0	99.3
	FAR	0.0	0.2	0.0	0.0

REFERENCES

[1] Cansian A.M., Moreira E., Carvalho A., and Bonifacio J.M., **"Network intrusion detection using neural networks"**, *in Proc. Int. Conf. on Computational Intelligence and Multimedia Applications*, pp. 276-280, 1997.

[2] Yeung D.Y., Ding Y., **"Host-based intrusion detection using dynamic and static behavioral models"**, *Journal of Pattern Recognition*, Vol. 36, pp. 229-243, 2003.

[3] Garcia-Teodoro P., Diaz-Verdejo J., Macia-Fernandez G., and Vazquez E., **"Anomaly-base network intrusion detection: techniques, systems and challenges"**, *Journal of Computers & Security*, Vol. 28, pp. 18-28, 2009.

[4] Ramadas M., Ostermann S., and Tjaden B., **"Detecting anomalous network traffic with self-organizing maps"**, *Recent Advances in Intrusion Detection, RAID, Lecture Notes in Computer Science (LNCS)*, Vol. 2820, pp. 36-54, 2003.

[5] Beghdad R., **"Training all the KDD data set to classify and detect attacks"**, *Neural Network World*, Vol. 17, pp. 81-91, 2007.

[6] Sheikhan M., Jadidi Z., **"Misuse detection using hybrid of association rule mining and connectionist modeling"**, *World Applied Sciences Journal*, Vol. 7, Special Issue of Computer & IT, pp. 31-37, 2009.

[7] Ye N., Emran S.M., Chen Q., and Vilbert S., **"Multivariate statistical analysis of audit trials for host-based intrusion detection"**, *IEEE Transactions on Computers*, Vol. 51, pp. 810-820, 2002.

[8] Kruegel C., Mutz D., Robertson W., and Valeur F., **"Bayesian event classification for intrusion detection"**, *in Proc. Annual Computer Security Applications Conf.*, pp. 14-23, 2003.

[9] Song D., Heywood M.I., and Zincir-Heywood A.N., **"Training genetic programming on half a million patterns: an example from anomaly detection"**, *IEEE Transactions on Evolutionary Computation*, Vol. 9, pp. 225-239, 2005.

[10] Sequeira K., Zaki M., **"ADMIT: anomaly-based data mining for intrusions"**, *in Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 386-395, 2002.

[11] Dickerson J.E., **"Fuzzy network profiling for intrusion detection"**, *in Proc. North American Fuzzy*

Information Processing Society (NAFIPS) Int. Conf., pp. 301-306, 2000.

[12] Gomez J., Dasgupta D., **“Evolving fuzzy classifiers for intrusion detection”**, *in Proc. IEEE Workshop on Information Assurance*, pp. 68-75, 2002.

[13] Shon T., Moon J., **“A hybrid machine learning approach to network anomaly detection”**, *Journal of Information Sciences*, Vol. 177, pp. 3799-3821, 2007.

[14] Han S.J., Cho S.B., **“Detecting intrusion with rule-based integration of multiple models”**, *Journal of Computers & Security*, Vol. 22, pp. 613-623, 2003.

[15] Novikov D., Yampolskiy R.V., and Reznik L., **“Artificial intelligence approaches for intrusion detection”**, *in Proc. IEEE Conf. on Systems, Applications and Technology*, pp. 1-8, 2006.

[16] Biermann E., Cloeteand E., and Venter L.M., **“A comparison of intrusion detection systems”**, *Journal of Computers & Security*, Vol. 20, pp. 676-683, 2001.

[17] Debar H., Dorizzi B., **“An application of recurrent network to an intrusion detection system”**, *in Proc. Int. Joint Conf. on Neural Networks*, pp. 478-483, 1992.

[18] Kayacik G., Zincir-Heywood N., and Heywood M., **“On the capability of an SOM-based intrusion detection system”**, *in Proc. Int. Joint Conf. on Neural Networks*, pp. 1808-1813, 2003.

[19] Golovko V., Vaitsekhovich L., Kochurko P., and Rubanau U., **“Dimensionality reduction and attack recognition using neural network approaches”**, *in Proc. Int. Joint Conf. on Neural Networks*, pp. 2734-2739, 2007.

[20] Beghdad R., **“Critical study of neural networks in detecting intrusions”**, *Journal of Computers and Security*, Vol. 27, pp. 168-175, 2008.

[21] Sheikhan M., Sha'bani A.A., **“Fast neural intrusion detection system based on hidden weight optimization algorithm and feature selection”**, *World Applied Sciences Journal, Special Issue of Computer & IT, Vol. 7,* pp. 45-53, 2009.

[22] Sheikhan M., Jadidi Z., and Beheshti M., **“Effects of feature reduction on the performance of attack recognition by static and dynamic neural networks”**, *World Applied Sciences Journal*, Vol. 8, pp. 302-308, 2010.

[23] Joshi M.V., Agrawal R.C., and Kumar V., **“Mining needless in a haystack: classifying rare classes via two-phase rule induction”**, *in Proc. ACM SIGMOD Conf. on Management of Data*, pp. 91-102, 2001.

[24] Lin Y., Chen K., and Liao X., **“A genetic clustering method for intrusion detection”**, *Journal of Pattern Recognition*, Vol. 37, pp. 924-927, 2004.

[25] Pfahringer B., **“Winning the KDD 99 classification cup: bagged boosting”**, *Journal of SIGKDD Explorations*, Vol. 1, pp. 65-66, 2000.

[26] Levin I., **“KDD classifier learning contest: LLSoft's results overview”**, *Journal of SIGKDD Explorations*, Vol. 1, pp. 67-75, 2000.

[27] Denning D.E., **“An intrusion-detection model”**, *IEEE Transactions on Software Engineering*, Vol. 13, pp. 222-232, 1987.

[28] Mukkamala S., Janoski G., and Sung A.H., **“Intrusion detection using neural networks and support vector machines”**, *in Proc. Int. Joint Conf. on Neural Networks*, pp. 1702-1707, 2002.

[29] Abadeh M.S., Habibi J., and Lucas C., **“Intrusion detection using a fuzzy genetic–based learning algorithm”**, *Journal of Network and Computer Applications*, Vol. 30, pp. 414-428, 2005.

[30] Tajbakhsh A., Rahmati M., and Mirzaei A., **“Intrusion detection using fuzzy association rules”**, *Journal of Applied Soft Computing*, Vol. 9, pp. 462-469, 2009.

[31] Sheikhan M., Gharavian D., **“Combination of Elman neural network and classification-based predictive association rules to improve computer networks' security”**, *World Applied Sciences Journal*, Vol. 7, Special Issue of Computer & IT, pp. 80-86, 2009.

[32] Sheikhan M., Khalili A., **“Intrusion detection based on rule extraction from dynamic cell structure neural network”**, *Majlesi Journal of Electrical Engineering*, Vol. 4, No. 4, pp. 24-34, 2010.

[33] Tamilarasan A., Mukkamala S., Sung A.H., and Yendrapalli K., **“Feature ranking and selection for intrusion detection using artificial neural networks and statistical methods”**, *in Proc. Int. Joint Conf. on Neural Networks*, pp. 4754-4761, 2006.

[34] KDD Cup 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed July 2008.

[35] Hochman R., Khoshgoftaar T.M., Allen E.B., and Hudepohl J.P., **“Using the genetic algorithm to build optimal neural networks for fault-prone module detection”**, *in Proc. 7th Int. Symp. on Software Reliability Engineering*, pp. 152-162, 1996.

[36] Sheikhan M., Movaghar B., **“Exchange rate prediction using an evolutionary connectionist model”**, *World Applied Sciences Journal*, *Special Issue of Computer & IT*, Vol. 7, pp. 8-16, 2009.

[37] Agrawal R., Joshi M.V., **“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, 2000.

[38] Duda R.O., Hart P.E., **Pattern Classification and Scene Analysis**, Wiley, 1973.

[39] Han F.M., **Principles of Neurocomputing for Science and Engineering**, McGraw Hill, 1991.

[40] Hartigan J.A., **Clustering Algorithms**, John Wiley and Sons, 1975.

[41] Lee Y., **Classifiers: Adaptive Modules in Pattern Recognition Systems**, *Cambridge, MIT Press*, 1989.

[42] Carpenter G.A., Grossberg S., Markuzon N., Reynolds J.H., and Rosen D.B., **“Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps”**, *IEEE Transactions on Neural Networks*, Vol. 3, pp. 698-713, 1992.

[43] Tran T.P., Jan T., **“Boosted modified probabilistic neural network (BMPNN) for network intrusion detection”**, *in Proc. Int. Joint Conf. on Neural Networks*, pp. 2354-2361, 2006.

[44] Chen Y., Abraham A., and Yang B., **“Hybrid flexible neural-tree-based intrusion detection systems”**, *International Journal of Intelligent Systems*, Vol. 22,

pp. 337-352, 2007.

[45] Chang R-I., Lai L-B., Su W-D., Wang J-C., and Kouh J-S., **"Intrusion detection by backpropagation neural networks with sample-query and attribute-query"**, *International Journal of Computational Intelligence Research*, Vol. 3, pp. 6-10, 2007.

[46] Venkatachalam V., Selvan S., **"An approach for reducing the computational complexity of LAMSTAR intrusion detection system using principal component analysis"**, *International Journal of Computer Science*, Vol. 2, pp. 76-84, 2007.

[47] Yu L., Chen B., and Xiao J., **"An integrated system of intrusion detection based on rough set and wavelet neural network"**, *in Proc. IEEE Int. Conf. on Natural Computation*, pp. 194-199, 2007.

[48] Sabhnani M., Serpen G., **"Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set"**, *Journal of Intelligent Data Analysis*, Vol. 6, pp. 1-13, 2004.