Journal of mathematics and computer Science 6 (2013), 85-96

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

A New and Quick Method to Detect DoS Attacks by Neural Networks

Mohammad Masoud Javidi, Mohammad Hassan Nattaj

Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

javidi@uk.ac.ir, m_nataj127@yahoo.com

Article history:
Received December 2012
Accepted January 2013
Available online January 2013

Abstract

Since it is technically impossible to create computer systems (Hardware & Software) without any defect or security failure, intrusion detection in computer systems' researches is specifically regarded as important.IDS is a protective system that can detect disorders occurring on the network. The procedure goes as intrusion detection can report and control occurred disorders through steps including collecting data, seeking ports, controlling computers, and finally hacking. So, intrusion detection can report control intrusion sabotage that composed of phases collecting data, probing port, gaining computer's control and finally hacking. In this paper, we consider some different agents, each of which can detect one or two DOS attacks. These agents interact in a way not to interfere each other. Parallelization Technology is used to increase system speed. Since the designed agents act separately and the result of each agent has no impact on the others, you can run each system on discrete CPUs (depending on how many CPUs are used in IDS computers) to speed up the performance.

Keywords: Multi-Layer Protection (MLP), Neural Network, Intrusion Detection System, Misuse-based IDS.

1. Introduction

The purpose of an Intrusion Detection system is not to prevent an attack, but only to discover and possibly detect the attacks and to recognize security problems in system or computer networks and also to report it to the system administrator [1]. Intrusion Detection systems are generally used with Firewalls as their security complements [2]. Detection of anomaly outside-in traffics of the network and reporting it to the administrator, or preventing suspected contacts is the other feature of IDS [3]. IDS is capable of detecting attacks by both internal and external users.

Wang et al. [4] have declared their ideas about intrusion detection and have explored different methods of neural network. There are many different intelligent techniques for designing intrusion detection systems, such as Machine learning, data mining, and fuzzy sets which are divided into two groups of Fuzzy set and Fuzzy anomaly detection, to mention some. Neural network algorithms are also divided into two groups of Supervised Learning and Unsupervised Learning.

Vasiliadis et al. [5] have also presented their idea about parallelization of IDSs, in which they have applied all system CPUs to speed up the detection process of all intrusions of the network.

In this paper, we have applied neural network using supervised learning that apply system CPUs as parallel for intrusion detection. We used NSLKDD database [6] for training and neural network testing. Creating some discrete IDS layers, each of which is considered as a single agent, we start precise detection of intrusions. In our proposed method, we have classified DoS attacks into 7 categories and now we try to detect each category by one agent which is equipped with IDS. These agents act independently and report intrusion or no-intrusion into the network. The agents' achievements will be analyzed by the final analyst and at last the analyst will report whether the intrusion is a DoS one or not.

The rest of paper is organized as follows. In section 2, we have presented an introduction to Intrusion Detection Systems (IDS) and Neural Networks. Section 3 describes related works. Proposed method will be presented in section 5. Our proposed method evaluation will be discussed in section 6. Finally, the general conclusion of the paper will be presented in section 7.

2. Background

In this section, a brief description of intrusion detection systems and neural networks is presented.

2.1. Intrusion Detection System (IDS)

Nowadays, Intrusion detection systems are most original and complete parts of a network monitoring system. Intrusion detection system technologies relatively are new and promise us that we will do in order to detect network intrusion that will help. Intrusion detection is the process in which events and incidents on a system or network monitoring and monitoring of the network or system intrusion is detected [7].

The goal of intrusion detection is screening, evaluating and reporting of network activity. This system acts on the data packets that have passed access control tool. Due to the reliability limitations, internal threats, and the presence of required doubt and hesitation, the intrusion prevention system should allow some cases of suspected attacks to pass in order to decrease the probability of false detections (false positive). On the other hand, most of IDS methods are intellectual and use different techniques to detect potential attacks, intrusions and abuses. Usually, an IDS uses the bandwidth in a way that can keep on acting without making any effect on accounting and network architecture. Intrusion detection systems (IDS), are responsible for identifying and detecting any unauthorized use of the system, abuse or any damage caused by both internal and external users [4].

Intrusion detection systems try to detect anomaly intrusions to the network by special algorithms which can be divided into 3 categories of misuse-based, anomaly-based, and specification-based. Analyzing the user's behavior in the network, the anomaly-based system can find out the intrusions. In anomaly-based method, an index of normal behavior is created [5]. An abnormality may be an indication of an intrusion. Indexes of normal behavior are created based on approaches like Neural Networks, Machine Learning methods, and even life style safety systems [6]. To detect anomalous behaviors, normal behaviors should be identified and some specific patterns and rules should be designed for them. The behaviors which follow these patterns are considered as normal and events which show any deviation beyond the normal statistics of these patterns are detected as abnormal. It's extremely difficult to detect abnormal intrusions, because there is no consistent pattern to monitor them. Usually an event which shows more than two deviations from the normal behavior is assumed to be normal.

Unfortunately, unusual anomaly-based intrusion detections and IDSs of this kind cause many false alarms (false positive) due to the fact that the behavior patterns of the users and the system are very

different. Unlike signature-based detection methods (which must be consistent with the patterns of previous attacks), abnormal behavior detection methods can detect any kind of new attacks.

In misuse-based technique, usually known as signature-based detection, pre-designed intrusion templates (signatures) are stored as law, in a way that each template contains different types of a specific intrusion and once a template of this kind appears in the system, the intrusion occurring is alarmed. Usually, in these methods, the detector has data bases of attack signatures or templates and tries to detect patterns that are similar to those stored in its own data base.

This kind of methods are able to detect known intrusions, and if new attacks appear anywhere in the network, they are not able to detect them. The administrator should continuously add the templates (patterns) of new attacks to the intrusion detection system. One of the advantages of this method is the high accuracy applied in detecting intrusions the templates of which have precisely given to the system [2].

2.2. Neural Networks

Neural networks are derived from human brain. Human brain has 10^{10} (ten billion) nerve cells on which 13^x6^{10} connections can be imagined. It means that we have an extremely complex connective system which can help the human brain to act as a parallel processor. In each neuron of a processor i.e. a human brain 10^{10} operation processes are taken place. Modifying or eliminating some processes may create new connections. A nerve cell processing is 10^{16} times less than a silicon processing. Some of the capabilities of Neural Network are: generalization capability, the ability to inflict, the ability to repair, the possibility to be used as a shared or society memory, addressable and stockpiling memory, ability to train, high speed due to parallel processing and so on [8].

A biological neuron generates outputs by collecting its inputs which apply to a neuron through definite Dendrites with a special Synapse weight and reaching to a definite extent. The definite extent which is the threshold is, in fact, the factor of activity or inactivity of neuron. Therefore, we can consider 3 factors which should be considered when artificially modeling biological neurons[9]:

- Neuron is either active or inactive.
- Output only depends on the neuron inputs
- Inputs should reach a certain extent to generate output.

2.2.1.Simple Perceptron Neural Network

Perceptron is the simplest type of neuron modeling in neural networks. Perceptron's input can be any type of data, but its output is necessarily Boolean type. The learning in perceptron is supervised. That means we need to have a number of input and appropriate output in order for perceptron to be able to imitate it.

The learning process of Perceptron is as follows:

- It generates an output.
- It compares the output with the expected output.
- It adjusts itself in order to be closer to the required output, if necessary.

After repeating the above steps, perceptron will converge to the correct behavior.

Frank Rosen Blatt [10] created and invented perceptron simply by connecting this neurons, and for the first time, he simulated this model in digital computers and formally analyzed them.

2.2.2. Multilayer Perceptron Neural Network (MLP)

In most complex mathematical problems which lead to solve crucial nonlinear equations, a Multilayer Perceptron network can be used simply by defining appropriate weights and functions. Varieties of functions are used in neurons according to the problem method. In this type of networks, an input layer is used to apply the problem input of a hidden layer, and an output layer to ultimately provide the answer of the problem. The nodes on the input layer are sensory neurons and those on the output layer are responding ones respectively. There are also hidden neurons on the hidden layer.

These networks are often trained with error back propagation. An example of a Multilayer Perceptron network is shown in Figure (1).

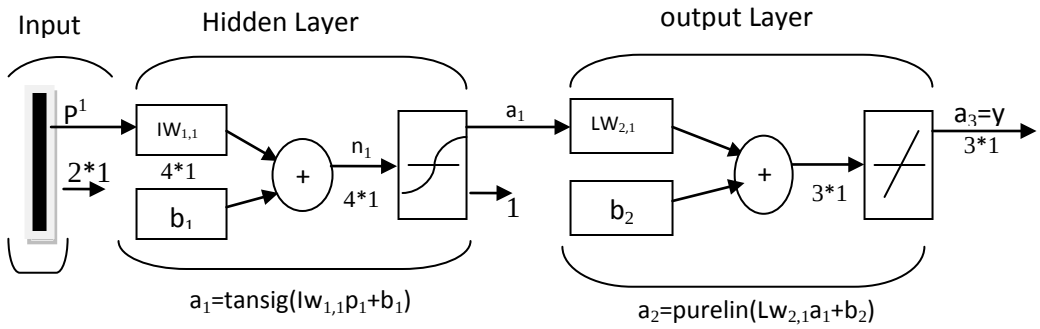

Fig. 1. The structure of the multilayer perceptron with *tansig* hidden neurons and output neurons with linear function [11].

Multilayer perceptron networks are built and applied with every number of layers, but the proposition we can accept without being proved declares that a three-layer perceptron network can divide any kind of space. This proposition, known as the Kolmogorov case, represents a very important concept that can be used in the construction of neural networks.

A special type of Multi-Layer neural networks is the Single Layer Perceptron (SLP). This network consists of one input and one output layer.

However, it is self-organizing neural network (SOM) that is very common way of Unsupervised Learning. The initial usage of SOM is data clustering and segmenting. Unlike MFNN that needs learning data including examples of both inputs and outputs, SOM only needs those data which contain explanatory inputs that explain specialties of variants or fields. Then SOM learns how to segment and cluster the data just on the basis of similarities or differences of input variants.

Therefore MFNNs are supervised neural network models which can be used for foresight and categorization; where SOMs are unsupervised neural network models that can be used for clustering [8,12].

3. Related works

Because of the capabilities of neural networks, they are able to act with deficient and incomplete data. Furthermore, there are Machine Learning (ML) techniques that can learn the templates they have not learnt during training phase. Most ML algorithms have been proposed to recognize DoS attacks, regardless of the minimum cost of error. These errors can lead to false alarms. The cost of false alarm is more expensive than non-detection [13].

Tan [8] used information such as instructions to CPU application to log in to the host address in order to detect and differentiate between normal and abnormal behaviors. While Rayan et al. [14] considered some templates of instructions and their frequencies.

Network traffic is another necessary data source. Cannady et al. in [15] executed Multi Learning Feed Forward Back Propagation (MLFF-BP) on 10,000 network packets collected from a simulated network for the purpose of misuse detection. Although the number of training and testing iterations needed 26.13 hours to get complete, but their experiments showed the MLFF-BP potentiality as a binary classifier for accurate detection of any attacks embedded in the experimental data.

Also, neural networks have multiple output neurons [16] to gather or classify multiple binary neural networks [11]. Usually, when faced with a new class, the latter is more flexible than the former for the network.

Ghosh et al. [17], compared the Recurrent Neural Networks (RNN) network with the MLFF-BP for predicting the sequence of system calling. The results showed a better performance for the RNN compared to MLFF-BP, in terms of accuracy of detection and false positive rate.

Cheng et al. [18], have applied RNN to detect network's anomaly intrusions in KDD99 base, while network's traffic data has a temporary local trait. Truncated-back-propagation-through-time learning algorithm was selected to accelerate the training. The information is kept in header but throwing away the payload leads us toward the loss of information.

4. The Proposed Method

In this section, the implementation of the proposed intrusion detection system, implementation steps and evaluation criteria are described.

4.1. Implementation

The network attacks can be divided in four groups of DoS, R2L, U2R, Probe. In the designed IDS, the system can detect DoS -type attacks, in very high detection rate. In fact, this kind of IDS is responsible for the detection attacks, which can be included in DoS category.

In order to design this type of IDS, we identified DoS attacks and designed a separate IDS for each one to detect that specific attack. In general, considering the designed IDS, the system will detect DoS attacks in the network (if there is any).

The whole process of the system is shown in Figure 2.

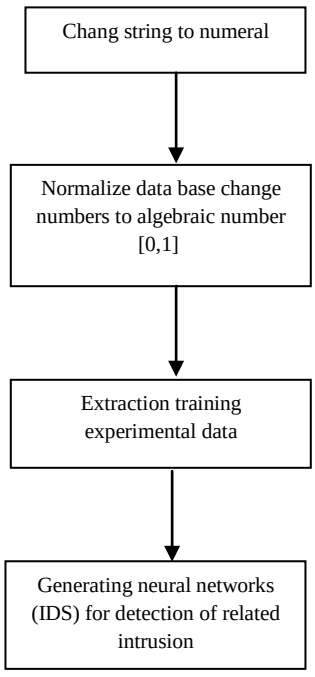

Fig. 2. The Process of generating system of intrusion detection

As you see in figure 2, in order to train neural network and have a more qualified process, we made some changes in database which as you see didn't affect on the totality of data base. It is just done for improving the function of neural network.

4.2. Implementation steps

The first step in MLP Method procedure is to standardize the data. The following equation can be used to standardize the data:

- In the used database, per 41 stated characteristics in each connection, a record (feature) is stated either numerically or in the order of strings. In other word, the type of connection is the order of strings, and in one of three possible modes of TCP, UDP and ICMP, which we have turned them into numeric values and given them numbers of 1, 2, 3 respectively. Table 1 shows all the changes.

Table1.the changes done in data base

String features of data base	Change to number
Connection is normal or in one of 39 available modes of attacks in data base	Assigning zero to the normal connection and numbers 1 to 39 to each attack
The kind of flag for each connection. There are 11 kinds of different defined flag in data base	Assigning numbers 1 to 11 to each flag
Three kinds of connection: ICMP, UDP, TCP	Assigning numbers 1 to 3 to them
The kind of presented service in each connection, which there are 64 kinds of different services in this data base	Assigning numbers 1 to 64, to each service

- The following formula is used for database normalization:

$$x_i = \frac{x_i - x_{i\,min}}{x_{i\,max} - x_{i\,min}} \qquad (1)$$

Where, X_i is a standard value of data, x_i is the real value of the data, x_{imin} is the smallest amount of data in the learning data, and x_{imax} is the biggest amount of data in the learning data. Thus, the data are put in the interval [0, 1].

- In this phase, the training and testing data are selected individually, which should have been done in the KDD99 [19] randomly (i.e., some of the records (approximately 25% of the whole database) should be considered as testing data and some of the records (approximately 75% of the database) should be considered as the training data). But for NSLKDD database, training and testing databases are used individually, i.e. the database can be used as standard.

Since, it is impossible to use the entire database is not possible (due to the high volume of data), we can choose 10% of the entire database. To do so, we should simply select the desired value of database in a way that there is information from all parts of the database.

- After selecting training and testing database, we should find a suitable algorithm for neural network. In this way, we could find the best algorithm for training through a variety of

experiments and a number of trial and errors. This algorithm can detect a high percent of attacks. General chart of the detection process is shown in Figure 3.

The designed IDS acts in binary mode i.e. the output of this IDS is either zero or one, in which Zero indicates *the attack* and One indicates the absence of that attack, that is if the designed neural network returns the number Zero, it means that an especial attack has taken place. For example, if IDS1 returns the number Zero, it means that IDS has detected the attack of Neptune (Neptune attack is a DoS attack which happen much more frequently than the others in the category) and its intrusion to the network and if it returns One it means that no Neptune attack has been detected in the entered data.

Since the obtained output of the neural network is not One and Zero (numbers are in the middle of zero and one), we defined a threshold to solve the problem, in a way that the output obtained from the network is divided into two groups of Zero and One. The mount 0.3 is the best criterion for the threshold, from which the highest perception of detection is obtained.

$$\begin{cases} a=0 & \text{if} \quad x<0.3 \\ a=1 & \text{if} \quad x>=0.3 \end{cases} \tag{2}$$

where, *a* is our favorite output, and x is the output obtained from the neural network.

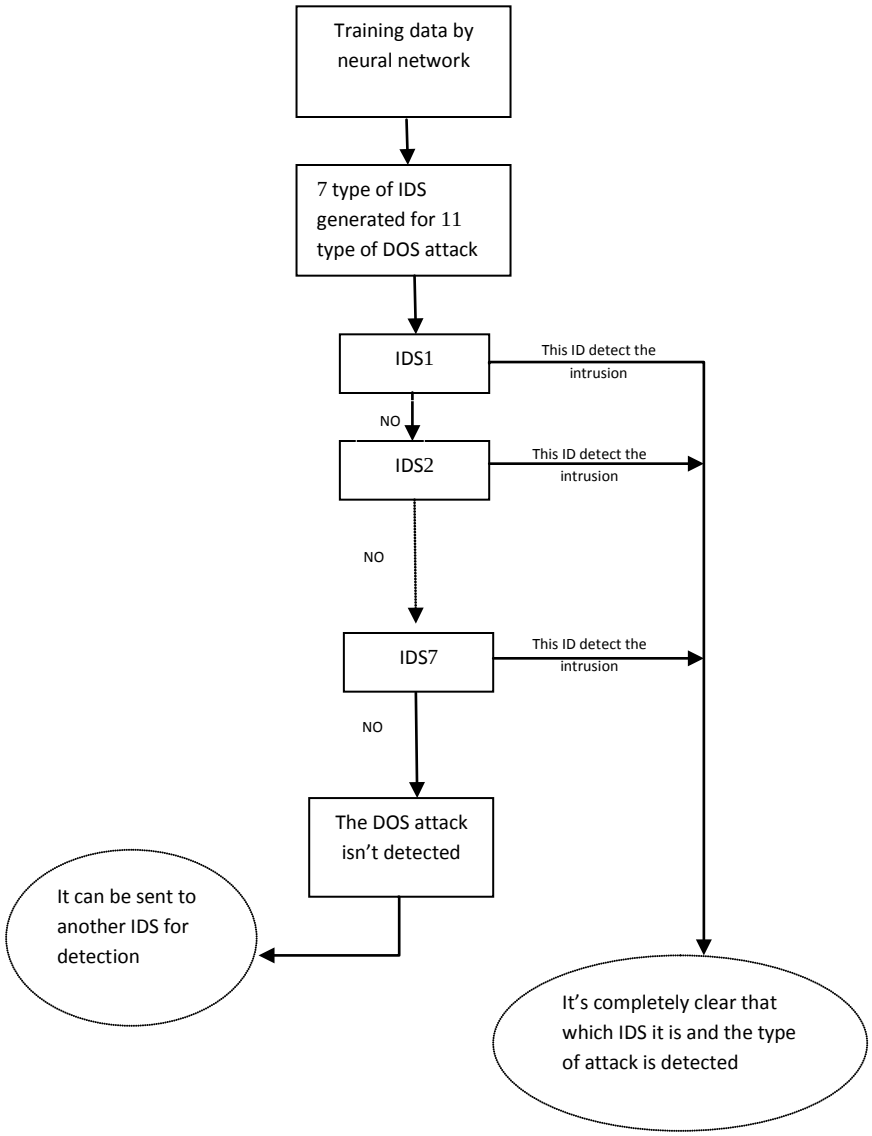

Fig. 3. The manner of intrusion detection by proposed system

4.3. Evaluation Criteria

To measure and detect the efficiency of the designed IDSs or the exact degree of their assurance and correctness the following criteria can be used:

True negative = correctly detect the normal data

True positive = correctly detect the attack

False Positive = distinguish normal events as attacks

False negative = distinguish the incidents of attack as normal

TNR = TN / (TN + FP) = the total number of normal incidents that are correctly detected / the total number of normal incidents that are detected as normal

TPR = TP / (TP + FN) = the number of incidents of attack that are correctly detected / the total number of incidents that are detected as attacks

FNR = FN / (FN + TP) = the number of attack incidents that are detected as normal / the total number of incidents that are detected as normal

FPR = FP / (FP + TN) = the number of normal incidents which are detected as attack / the total number of incidents which are detected as attack

In this implementation, we used the TPR criterion and as you see in the chart above, the efficiency of this implementation is approximately more than 98%.

5. Evaluation of results

To implement the neural network algorithm, we applied the MATLAB software (version 7.12.0.635, 32bit). In order to implement this algorithm, we should firstly train the designed neural network using training data, and then we should analyze the efficiency of the network using the experiment data. Note that the algorithm used in the proposed intrusion detection system is a neural network of MLP type. The computer used for implementation was a dual-core 1GHz CPU with 2G RAM.

5.1. NSLKDD database

The used database consists of information about standard connection records which in their own turn consist of a set of simulated attacks and intrusions in a military network.

A connection is a series of packets with protocols TCP, UDP or ICMP which start and finish at specific times and run under a defined contract between data from the origin IP address to the destination IP address. Each connection is labeled as normal or an attack, and in the case that it is an attack, its type is exactly defined. The record of each connection consists of about one hundred bytes.

The attacks shown in this data set fall in four principle groups of DoS, R2L, U2R and Probe that are shown in Table 2 [7]. As shown in the table, approximately 74 percent of the whole database is composed of DoSattacks which indicates the importance of creating an IDS to detect this group of attacks.

Each record from NSLKDD dataset [6], describes one connection in the form of 41 features.

To do so, The NSLKDD database which is the optimal version of KDD99, should be divided into two parts of Test and Training. We used NSLKDD database to design the proposed intrusion detection system. Before using this database, it is necessary to mention that this database has a high capacity and we used only 10% of the records for testing and training the designed network. Of course, this 10% is chosen in such a way that contains different types of attacks includes different modes of the network [7].

Table 2. Classification and dispersion percentage of attacks in database

Connection type	KDD Testing Set
Normal	19.48%
DoS	73.90%
U2R	1.34%
R2L	5.2%
Probe	0.07%

5.2. Configuration of MLP

To design the proposed intrusion detection system, we used 7 discrete IDSs, each of which has been configured using neural network. The configuration of these IDSs has been obtained with trial and errors and after performing the configuration process for 50 times, we have achieved some good results on the rate of intrusion detection. Figure 4 shows the detection percentage without considering the parallelization operations.

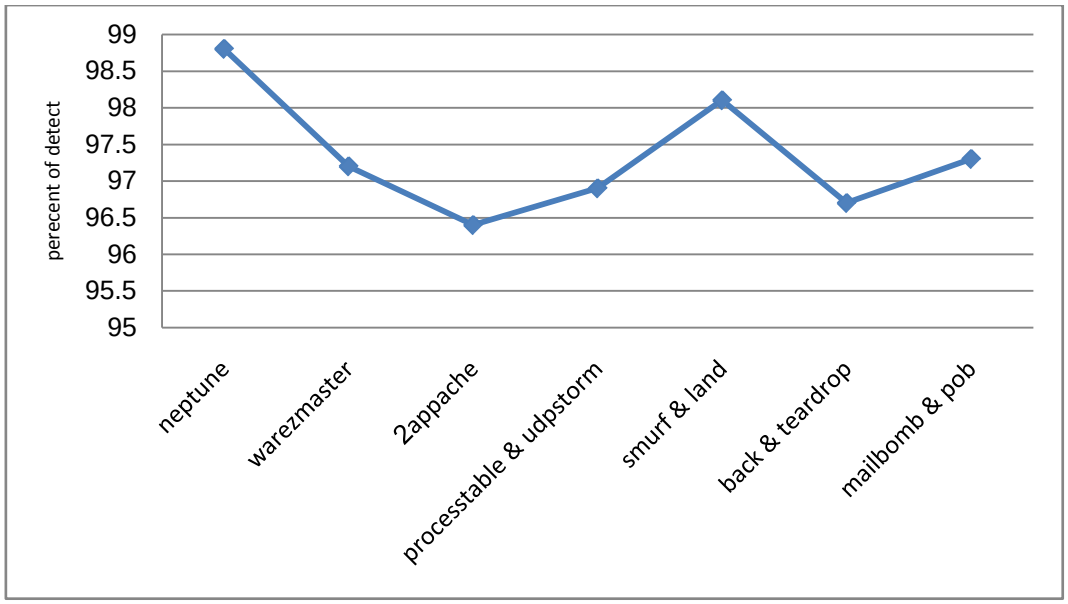

Fig. 4. The percent of detection of DoS attack

For example, we consider the IDS designed for the detection of Neptune attack which is one of the IDS attacks. In this method, the Learning Function of Trainrp, Transfer Functions of *'tansig'* for the output layer and Transfer Function of *'purelin'* in the hidden layer has been used. In this implementation, the average of moods in 50 tests with a middle layer with 11 neurons and learning rate of 0.01 and the *rp* of 0.4, has been equaled 98.8%.

5.3. Comparison with Other Intrusion Detection Systems

Several Multi Layout (ML) algorithms have been presented for the detection of DoS attacks that are available in articles [20,21,22,23,24]. Table 3 shows a comparison between several IDSs which can detect DoS attacks.

Table 3. Comparison between designed Intrusion Detection Systems

False positive rate	True positive rate	Classifier
3.1	96.6	Multi layer perceptron(MLP)_Resilient Back Propagation[20]
4.2	95.8	Decision Tree(C4.5)[21]
10.3	89.7	Support Vector Machine (SVM)[22]
6.5	93.5	K_Nearest Neighbor(K_NN)[23]
7.0	93.5	K_Meanse Clustering[24]
3.0	98.94	Proposed algorithm (with neural network resilient as paralleling)

In addition to the improvement of the intrusion detection percentage, and reduction of the false positive rate in this method, the procedure speed has also been enhanced. The rate of efficiency in this algorithm depends on the number of the processors available on the system that is arranged to perform IDS operation. Since the algorithm of the proposed intrusion detection is designed to detect each attack individually, and each designed IDS is located on one separate agent, therefore we apply these agents in a parallel mode on the available processors on the desired system which leads to improving the speed of the procedure.

5.4. Parallelization

The fact is that, the proposed algorithm has an accepting speed, even without parallelization, but it will reach a much higher speed when using parallelization on the processors. Figure 5 shows the improvement rate of the algorithm.

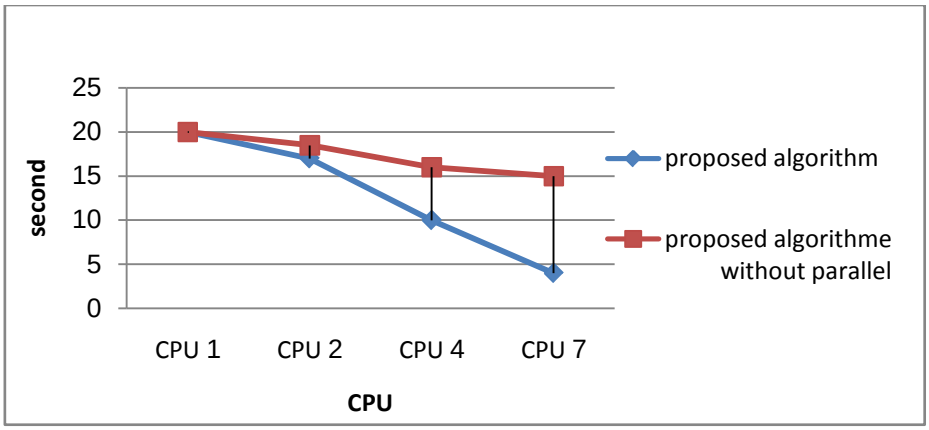

Fig. 5. Comparison between time performances

Table 4 shows the percentage of the intrusion detection speed in the proposed method.

Table 4. Speed improvement

Total CPU usage	**2 CPU**	**4 CPU**	**7 CPU**
The percent of speed improvement	8%	37%	60%

6. Conclusions

There are different algorithms for implementation and networks intrusion detection in computer networks. Depending on what kind of attack and in what degree of accuracy is going to be detected, we choose the appropriate algorithm and the best method of implementation for it. In the proposed IDS, we used supervised neural network to detect DOS intrusions in NSLKDD database and we also improved the speed of implementation using Parallelization Technology.In the proposed IDS, we used misuse-based, also known as signature-based technique. In the used data base, it is defined for each record that it is a special kind of attack or a normal connection is done. Therefore, we designed IDSs using the neural network that can detect a specific attack, and an agent is considered for each attack, while the appropriate IDS of the detection of the attack are located on that agent. In other words, we can detect the DoS attacks using multi-agents that can work in parallel.

References

[1] S. A. Hofmeyr, S. Forrest, A. Somayaji, J. Computer Security - JCS, 6(3), p151 **(1998)**.

[2] E. Lundin, E. Jonsson, , J. Computer Networks, 3(4), p623 **(2000)**.

[3] X. D. Hoang, J. Hu, P. Bertok, "A multi-layer model for anomaly intrusion detection using program sequences of system calls",Proceeding of the IEEE International Conference on Networks (ICON), 531-536**(2003)** .

[4] G. Wang, J. Hao, J. Ma, L. Huang, J.Expert Systems with Applications, 37(9), p6225 **(2010)**.

[5] G. Vasiliadis, M. Polychronakis, S. Ioannidis, "MIDeA: a multi-parallel intrusion detection architecture", Proceeding of the 18th ACM conference on Computer and Communications Security **(2011)** 297-308;New York.

[6] http://nsl.cs.unb.ca/NSL-KDD

[7] S.P. Shieh, V.D. Gligor, , J. IEEE Trans. on Knowledge and Data Engineering, 9(4),p 661 **(1997)**.

[8] J. Cannady, "Artificial neural networks for misuse detection", Proceedings of the 21st National Information Systems Security Conference, Berlin, **(1998)** 368–381.

[9]A. Fakharzadeh, Z. Alamdar, M. Hosseinipour, J. Math. Computer Sci. 5(3), p160 **(2012).**

[10] D. Bolzoni, S. Etalle, P. Hartel, "POSEIDON: A 2-tier anomaly based network intrusion detection system", Proceeding of the 4th IEEE International Workshop on Information Assurance **(2006)** 156-166.

[11]H. Demuth, M. Beale, Neural Network Toolbox User's Guide for use with MATLAB, TheMathWorks Inc. **(2011)**.

[12]M. Kuchaki Rafsanjani, Z. Asghari Varzaneh, N. EmamiChukanlo, J. Math.Computer Sci. 5(3),p 229 **(2012)**.

[13] J. Ryan, M.J. Lin, R. Miikkulainen, J. Neural Information Processing Systems,10, p943 **(1998)**.

[14] W. Luo, X. Wang, X. Wang, A novel fast negative selection algorithm enhanced by state graphs, in: L.N. de Castro, F.J.Z.H. Knidel (Eds.), Artificial Immune Systems, vol. 4628 of Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, **(2007)** 168–181; Verlag Berlin.

[15] S. H. Abulhaiba, S. A. Mahmood, R. J. Green, IEEE Trans.on PAMI, 16(6), p664 **(1994)**.

[16] M. Ostaszewski, F. Seredynski, P. Bouvry, Immune anomaly detection enhanced with evolutionary paradigms, Proceedings of the ACM Genetic and Evolutionary Computation Conference (GECCO'06), Seattle, WA, USA **(2006)** July 119–126; Seattle, Washington, USA.

[17] S. X. Wu, W. Banzhaf, 10(1), p1 **(2010)**.

[18] D. Parikh, T. Chen, , J. IEEE Transactions on Information Forensics and Security,3(6), p381 **(2008)**.

[19] The KDD99 Dataset, http://kdd.ics.uci.edu/databases/kddcup99/task.html.

[20] D. Gavrilis, E. Dermatas, "Real time detection of distributeddenial-of-service attacks using RBF networks and statistical features", J.Computer Networks, 48, 5, p235 **(2005)**.

[21]Hoai-Vu Nguyen, Yongsun Choi, J. International Journal of ComputerSystem Science and Engineering, p247 **(2008)**

[22] R. Jalili, F. Imani–mehr, M. Amini, H. R. Shahriari , J. Lecture Notes in Computer Science(LNCS), 3439, p192 **(2005)**.

[23] S. Seufert, D. O Brein, ''Machine Learning for Automatic Defenseagainst Distributed Denial of Service Attacks'', IEEEInternational Conference (ICC) **(2007)** 1217–1222; Dublin City Univ., Dublin.

[24] Z. Jing-jing, H. Xiao-hong, S. Qiong, M.A. Yan, J. China Universities of Posts and Telecommunications, 15, p68 **(2008)**.