Intrusion Detection and Attack Classification Using Feed-Forward Neural Network

Fariba Haddadi[1], Sara khanchi[2], Mehran Shetabi[3], Vali Derhami[4]
Electrical and Computer Engineering Department
Yazd University
Yazd, Iran.
[1]f.haddadi @stu.yazduni.ac.ir
[2]s.khanchi@stu.yazduni.ac.ir
[3] mshetabi@yazduni.ac.ir
[4] vderhami@yazduni.ac.ir

Abstract— **Fast Internet growth and increase in number of users make network security essential in recent decades. Lately one of the most hot research topics in network security is intrusion detection systems (IDSs) which try to keep security at the highest level. This paper addresses a IDS using a 2-layered feed-forward neural network. In training phase, "early stopping" strategy is used to overcome the "over-fitting" problem in neural networks. The proposed system is evaluated by DARPA dataset. The connections selected from DARPA is preprocessed and feature range is converted into [-1, 1]. These modifications affect final detection results notably. Experimental results show that the system, with simplicity in comparison with similar cases, has suitable performance with high precision.**

Internet; Artificial Neural network; Back propagation; DARPA; Feed-forward; Intrusion;

I. INTRODUCTION

By internet expansion, information security and privacy have been became more important in recent decades. Security methods like cryptography and firewalls do not satisfy user's needs. This causes the use of more complex security systems, such as Intrusion Detection Systems (IDS's), to be crucial.

IDS gathers information from a computer or computers network and attempts to detect intruders or system abuse. Generally, an IDS will notify a human analyst of a possible intrusion and take no further action, but some newer systems take active steps to stop an intruder at the time of detection [1]. IDSs are categorized into two major groups: network-based and host-based.

Network-based IDSs detect attacks based on network traffic analysis but host-based IDSs use system information like CPU load, system calls and etc., for detection purpose.

IDS techniques are arranged into three general groups: Anomaly Detection, Signature Detection (Misuse) and hybrid.

Anomaly detection IDSs model normal behavior of system and consider each event that differs from this model more than a threshold value as an intrusion. Signature based IDSs have a database of previous attack signatures and compare each behavior with database entries. If a match found, report it as intrusion.

Modeling normal behavior of a system, in anomaly IDSs, is so cumbersome due to its complexity. But in other hand, they are best suited for detection of new attacks.

Due to the availability of predefined pattern of previous attacks, misuse IDSs are vulnerable to new ones. Hybrid IDSs take advantages of two previous systems.

Nowadays, most commercial IDSs use rules to create attack pattern. Rule-based systems like expert systems follow fixed rules which should be periodically updated. Soft computing techniques lend a hand to intrusion detection systems to solve this problem [2].

Soft computing techniques like ANN[1], fuzzy logic, genetic algorithm and etc., are utilized in IDSs due to their flexibility and learning capability. These intelligent systems construct a general model of existing patterns which will be able to detect new ones.

Several IDSs employ intelligent methods. Heywood et al. [3] propose a hierarchical neural network for intrusion detection based on SOM[2]. This three-layered hierarchical SOM architecture uses two sets of features, one is limited to 6 basic KDD features and the other consists of all 41. Jirapunimm et al. [4] use combination of SOM and MLP[3]. SOM is used as a preprocessing level and its outputs are fed to MLP as inputs. This hybrid network is formed as a 5-layered feed-forward neural network. The first layer is input layer, the second one is SOM layer and 3 next layers are MLP layers. J. Shum et al. [5] designed an intrusion detection system based on feed-forward neural network with back propagation. Their network composed of an input layer, a hidden layer and an output layer. E. Hernandez-pereira et al. [6] utilized three techniques: SVM, one layer and multilayer feed-forward neural networks. They focused more on conversion of symbolic features to numerical ones and compare effect of different conversion techniques on intrusion detection. There are more IDSs based on soft computing techniques such as [7], [8] and [9].

[1] Artificial Neural Network
[2] Self Organizing Map
[3] Multi-Layer Perceptrons

978-0-7695-4042-9/10 $26.00 © 2010 IEEE
DOI 10.1109/ICCNT.2010.28

In this paper, a network based IDS, using a supervised 2-layer feed-forward neural network with back propagation, is proposed. This system can distinguish normal connections and attacks. Also it is able to classify attacks into four major known types.

The paper is organized as follows: section 2, introduces neural networks and back propagation algorithm. Section 3, reviews neural network IDS's shortcomings. Section 4, describes proposed IDS architecture. Section 5, introduces KDD cup 99 data set. Section 6, evaluates the proposed system and at last, section 7 presents the conclusion of this work.

II. Artificial Neural Network

Artificial Neural Network is inspired from human neural system and it is used in different areas like pattern recognition, optimization, control and etc. Neural network is composed of several processing units (nodes) and directed links between them. These connections are weighted representing relation between input and output neurons [10].

Neural networks are classified into two groups based on connections:

- **Feed-forward networks:** represent a function of its current input; thus, it has no internal state other than the weights themselves.
- **Recurrent networks:** feeds its outputs back into its own inputs

A. *Feed-forward Neural Network*

The Multilayer feed-forward neural network has several neurons structured in layers such input layer, hidden layers and output layers (Fig.1). Output layer with one or many neurons provides output for one or many inputs. In one neuron example, training process task is to find proper weights for neuron connections which in combination with inputs, achieves the desired output. This process is accomplished by back propagation algorithm [11].

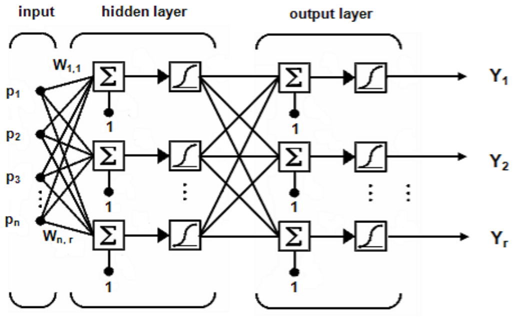

Figure 1. feed-forward neural network

B. *Back Propagation Algorithm*

Back propagation algorithm propagates the error from the output layer to the hidden layers and changes weights recursively through network from output layer to input layer. The main objective of algorithm is to minimize output error by changing weights. Back propagation algorithm is based on gradient descent. In each step, the goal gradient is computed which direction of negative gradient represents the direction where the surface decreases more rapidly and amount of gradient shows the distance through which the direction is valid.

The Classic back propagation adjusts weights in gradient descent direction (negative gradient) in which the performance function decreases more rapidly. When function is decreasing in negative gradient, it doesn't necessarily result in fastest convergence. But in conjugate gradient algorithms, a search is done in conjugate with directions which cause faster convergence. In most conjugated algorithms, step size is adjusted each iteration.

III. Proposed Intrusion Detection System

Most of neural network based IDSs suffer from three major problems:

- **Over-fitting:** this problem is encountered when neural network is over-fitted with a portion of data. In this case, error is very few in training but high in test because it loses the generalization ability.
- **High memory consumption:** in IDS field, we encounter huge amount of data which neural networks suffer low memory in training phase. So selection of a proper training function is playing an important role.
- **Overhead:** there are lots of computations in complex neural networks and cause overhead. This computational overhead is grown by complexity of system.

Here we propose our IDS and next show how our solution figure out the problems stated above.

The proposed IDS is structured as a feed-forward neural network with back-propagation algorithm. Neural network properties like parallelism, distributed computation, learning capability, adaptively and fault tolerance made it suitable for intrusion detection systems.

Also as feed-forward neural network (with one or more hidden layer) can estimate every function with desired precision [12] and its simplicity over many other neural networks, we choose this network for our IDS. The proposed system has three phases: preprocessing, training and detection, which illustrated in Fig.2.

A. *Preprocessing Phase*

In preprocessing phase, we choose "Corrected" file from KDD cup 99 dataset [13] which consists of different types of attacks and normal connections all together. We divide each kind of attack and normal connections in separate files and change symbolic feature into numerical ones (in a range [-n, n]) to be used in training phase. Symbolic features are: protocol type, service and flag. Then we map all connection features into range [-1, 1] for unification and error reduction.

B. *Training Phase*

We construct a two-layered feed-forward neural network (a hidden layer) and set weights randomly. Input layer consists of 41 inputs corresponding to number of features. Similarly, the number of connection features and hidden layer has 35 neurons.

Classic back-propagation in neural network IDSs is not applicable due to huge amount of data. Selecting a proper training function solves this problem. We use “trainscg” fast training function.

Layer 1 transfer function is “tansig” and layer 2 is “pureline”.

We take advantage of “early stopping” strategy. For this purpose, we choose 20% of training data randomly as validated data. This data is not used in network training and its role is stopping training when network is going to be over-fitted.

In this phase, two types of data are available, training and target. Indeed, training is supervised and for each input data, there is a target. Target of normal connection and each attack category is as:

- Normal: [1 0 0 0 0]
- DoS: [0 1 0 0 0]
- Probe: [0 0 1 0 0]
- R2L: [0 0 0 1 0]
- U2R: [0 0 0 0 1]

Along training phase each connection instances is mapped to corresponding target.

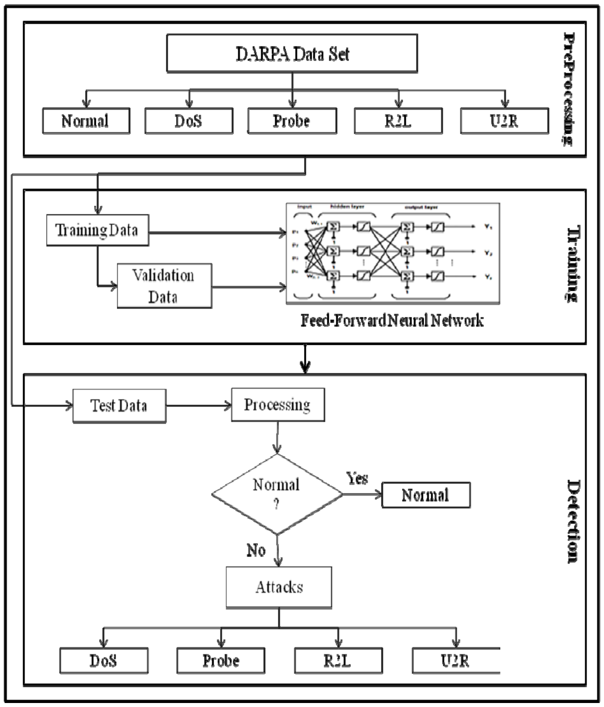

Figure 2. Overall system

C. Detection Phase

In this phase, we use test data for evaluation of proposed IDS. Neural network output is a three element vector with real numbers. For mapping output vector to one of target vectors, maximum value of a vector, transforms to one and others to zero. After conversion is done, it is compared with the target vectors and when a match found, the type of input connection is detected (index of entry with value one is the index of the group).

IV. DARPA Dataset

DARPA project is expanded in MIT University in 1998 to provide IDS developers with a benchmark to compare their products [14].

The project simulation period was seven weeks. All the traffic was recorded in TCPdump file consists of normal traffic and four major groups of attacks [1]. Attack types are:

- DoS[4]: Attacker tries to prevent legitimate users from using a service.
- R2L[5]: Attacker does not have an account on the victim machine, hence tries to gain access.
- U2R[6]: Attacker has local access to the victim machine and tries to gain super user privileges.
- Probe: Attacker tries to gain information about the target host.

In 1999, the original TCPdump was preprocessed for practical use. The TCPdump information packets were summarized into several connections. A connection is a sequence of TCP packets which flows between two specified source and destination under a known protocol in special time. A file with about 5,000,000 connections was prepared which was named “KDD cup 99”. Each connection has 41 features which among them 38 are numerical and others are symbolic. 22 of these features describe the connection itself and 19 of them explain the properties of connections to the same host in last 2 seconds. A complete description of all 41 features is available [14], [15].

V. Experimental Results

First we extract some connections among files separated and prepared in preprocessing phase to be used as test data in proposed IDS. Then a trivial experiment is done on connection feature’s range. First the features with original values are used in training and testing phases which end up to a very low detection rates. We decide to change feature range to [-1, 1] to unify effect features. This conversion has a satisfiable result. Therefore, this mapping is added to preprocessing tasks.

We train and test our system with two datasets with different number of connections in each set. Number of these two datasets in each experiment is shown in table.1. In each group, 80% of data are used as training data and 20% as test data.

In training phase, we examine several number of validation datasets. The system is run under each validation set and the final results are registered for each group. Among these selections, we conclude selection of 20% of training data as validation set, is the best choice and results in the

[4] Denial of Service

[5] Remote to Local

[6] User to Remote

best detection rates. This validation data is used to overcome "over-fitting" problem in neural network.

TABLE I. NUMBER OF CONNECTIONS

Connection type	All connections	Dataset 1	Dataset 2
Normal	60,693	5,000	10,000
DoS	229,853	5,000	10,000
Probe	4,166	4,000	4,000
R2L	16,347	5,000	10,000
U2R	70	70	70

Using dataset 1, after 599 epochs, training stopped by "early stopping" technique. Convergence process is shown in fig.3. The same is occurred in 611 epochs for dataset 2 which is shown in fig.4.

After training phase, proposed IDS is evaluated by test data in each group of dataset which is mentioned in table1.

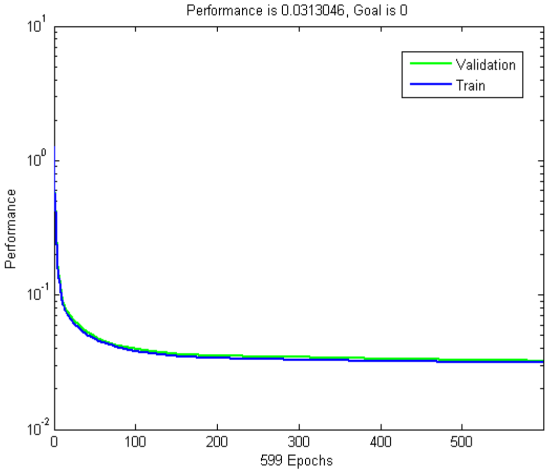

Figure 3. Convergence process (dataset 1)

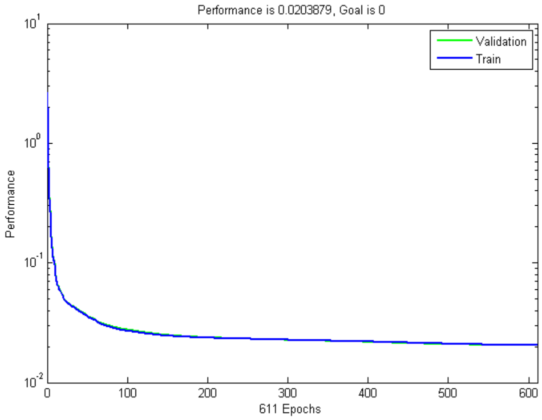

Figure 4. Convergence process (dataset 2)

As table.2 illustrates proposed IDS evaluated by two datasets under the same condition, doesn't differ so much. Indeed, selection of dataset 1 with less number of connections appears to be more suitable because of faster training and almost the same detection rate with dataset 2.

TABLE II. PROPOSED IDS RESULTS

	Dataset 1	Dataset 2
Normal	79.1%	79.8%
DoS	97.5%	97.5%
Probe	99%	99.1%
R2L	96.6%	98.9%
U2R	40%	34.5%

VI. CONCLUSION

A network-based intrusion detection system using a 2-layered feed-forward neural network was proposed. The system classified the normal connections and attacks. After detection of attack, type of attack was determined by the system in detail. Using conjugated training function and validation dataset caused: faster training, less overhead, less memory consumption and over-fitting prevention. Two experiments have been performed on different number of connections in training and testing datasets. These data is obtained from KDD cup 99 dataset after some preprocessing such as: symbolic feature mapping, feature range conversion to [-1, 1] and etc.. Results implied that proposed IDS performance, in these two experiments, was almost the same and detection rates were very close. Therefore, because of lower computational overhead, IDS with less data is more suitable.

R2L and U2R detection rates in neural network IDSs are not good enough. But the proposed system has achieved an enormous improvement in these two types of attacks detection rates. The proposed system is very simple and new in IDS field. Although it's simple structure, in comparison with similar IDSs, it achieves equivalent performance and reduces computational overhead and memory usage.

REFERENCES

[1] K. Kendall, "A databases of computer attacks for the evaluation of intrusion detection systems", Master Thesis, MIT, 1999.

[2] M. Moradi and M. Zulkhenine, "A Neural Network Based System for Intrusion Detection and Classification of Attacks", in Proc. IEEE Advances in Intelligent Systems - Theory and Applications, pp. 148:1-6, Luxembourg, November 2004.

[3] H. Kayacik, A. Zincir-Heywood, and M. Heywood, "A hierarchical SOM-based intrusion detection system", in Proc. Elsevier Engineering Application of Artificial Intelligence, pp. 439-451, 2007.

[4] C. Jirapummin, N. Wattanapongsakorn and P. Kanthamanon, "Hybrid neural networks for intrusion detection system", Proceedings of the 2002 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2002), pp. 928-931, Thailand, 2002.

[5] J. Shum and H.A. Malki, "Network intrusion detection system using neural network", in Proc. IEEE Fourth Int. Conference on Natural Computation, pp. 242-246, 2008.

[6] E. Hernandez-pereira, J.A. Suarez-Romero, O. Fontela-Romero and A. Alonso-Betanzos, "Conversion methods for symbolic features: A

comparison applied to an intrusion detection problem", in Proc. Elsevier Expert Systems with Applications, 2009.

[7] W.H. Chen, H.S. Hsu and H.P. Shen, "Application of SVM and ANN for intrusion detection", in Proc. Elsevier Computers and Operations Research, pp. 2617–2634, 2005.

[8] F. JEmii, M. Zaghdoud and M. Ben Ahmed, "A Framework for an Addaptive Intrusion Detection System Using Bayesian Network", in Proc. IEEE, PP. 66-70, 2007.

[9] G. Liu, Z. Yi and S. Yang, "A Hierarchical Intrusion Detection Model Based on the PCA Neural Networks", in Proc. Elsevier NeroComputing, pp.1561-1568, 2007.

[10] S. J. Russell and P. Norvig, "Artificial intelligence: A modern approach (international edition)", Pearson US Imports & PHIPEs, November 2002.

[11] S. Haykin., "Neural networks: A comprehensive fundation", McMMillan, New York, 1994.

[12] S.Theodorios and K. Koutrrombas, "Pattern recognition", Cambridge: Academic Press, 1999.

[13] KDD99, KDD cup 1999 data, <http://kdd.ics.uci.edu/databases /kddcup99/kddcup99.html>, 1999.

[14] MIT Lincoln Laboratory, The 1998 intrusion detection off-line evaluation plan, <http:// WWW.ll.mit.edu>, 1998.

[15] S. Mukkamalla, "Intrusin Detection Using Neural Networks and Support Vector Machine", in Proc. IEEE, 2002