Journal of Engineering Science and Technology
Vol. 8, No. 1 (2013) 107 - 119
© School of Engineering, Taylor's University

A COMPARISON STUDY FOR INTRUSION DATABASE (KDD99, NSL-KDD) BASED ON SELF ORGANIZATION MAP (SOM) ARTIFICIAL NEURAL NETWORK

LAHEEB M. IBRAHIM[1,*], DUJAN B. TAHA[1], MAHMOD S. MAHMOD[2]

[1]Department of Software Engineering, College of Computer
Sciences and Mathematics, University of Mosul, Iraq
[2]College of Sciences, University of Mosul, Iraq
*Corresponding Author: dr.laheeb2007@yahoo.com

Abstract

Detecting anomalous traffic on the internet has remained an issue of concern for the community of security researchers over the years. The advances in the area of computing performance, in terms of processing power and storage, have fostered their ability to host resource-intensive intelligent algorithms, to detect intrusive activity, in a timely manner. As part of this project, we study and analyse the performance of Self Organization Map (SOM) Artificial Neural Network, when implemented as part of an Intrusion Detection System, to detect anomalies on acknowledge Discovery in Databases KDD 99 and NSL-KDD datasets of internet traffic activity simulation. Results obtained are compared and analysed based on several performance metrics, where the detection rate for KDD 99 dataset is 92.37%, while detection rate for NSL-KDD dataset is 75.49%.

Keywords: Anomaly, Intrusion detection system, Artificial neural network,
Self-organization map, KDD99, NSL-KDD.

1. Introduction

Question is often asked of intrusion detection advocates. Why bother detecting intrusions if you've installed firewalls, patched operating systems, and checked passwords for soundness? The answer is simple: because intrusions still occur. Just as people sometimes forget to lock a window, for example, they sometimes forget to correctly update a firewall's rule set. Even with the most advanced protection, computer systems are still not 100% secure [1].

Security policies or firewalls have difficulty in preventing attacks because of the hidden weaknesses and bugs contained in software applications. Moreover, hackers constantly invent new attacks and disseminate them over the internet. Disgruntled employees, bribery and coercion also make networks vulnerable to attacks from the inside. Mere dependence on the stringent rules set by security personnel is not sufficient. Intrusion Detection Systems (IDS), which can detect, identify and respond to unauthorized or abnormal activities, have the potential to mitigate or prevent such attacks [2].

Computer security gives user features such as network connectivity; but we'll never achieve the goal of a completely secure system. Then we must design intrusion detection systems to discover and react to computer attacks. The goal of intrusion detection system is to detect intrusions. Intrusion detection systems (IDS) have emerged to detect actions which endanger the integrity, confidentiality or availability of a resource as an effort to provide a solution to existing security issues. This technology is relatively new, however, since its beginnings, an enormous number of proposals have been put forward to sort this situation out in the most efficient and cost effective of manners [3].

Many methods have been proposed to build intelligent and automated IDS swhich can detect and prevent the attacks to do there piracy on the computer network. Rule-based expert system and statistical are used as a detector in many IDSs, a rule-based expert can detect some well-known intrusions, but it is difficult to detect novel intrusions, and its signature database needs to be updated manually and frequently, also a statistical-based IDS needs to collect enough data to build a complicated mathematical model, which is impractical in the case of complicated network traffic [4].

Artificial neural network (ANN) is one of the main soft computing algorithms used in many researches as detector agent in IDSs. ANN in these researches is used to solve a number of problems encountered by other current intrusion detection methods, and have been proposed as alternatives to the statistical analysis component of anomaly detection systems.

Neural network initially gains experience by training the system to correctly identify preselected examples of the problem. The neural network response is reviewed, and the system configuration is refined until the neural network analysis of the training data reaches a satisfactory level. In addition to the initial training period, the neural network also gains experience over time as it conducts analyses on data related to the problem [3, 5].

In order to solve the problems of traditional methods used as detector, an off-line anomaly intrusion detection system is developed based on ANNs. This system uses normal behaviour to detect those unexpected attacks. In particular, Self Organization Map Artificial Neural Network have considered for anomaly detection based on newest NSL-KDD dataset.

The remainder of the paper is organized as follows: Section 2 presents related work regarding IDSs with ANN. Section 3 introduces our proposed system. Section 4 contains the experiments conducted, Section 5 discusses the results, and Section 6 presents conclusions and plans for future studies.

2. Related Work

An ID is becoming one of the main technologies used to monitor network traffics and identify network intrusions. There are different taxonomies have been suggested for IDSs [6-8]. One of these taxonomies depends on the source of audit data that will be used to detect possible intrusions.

A number of approaches based on computing have been proposed for detecting network intrusions. The guiding principle of soft computing is exploiting the tolerance of imprecision, uncertainty, partial robustness and low solution cost. Soft computing includes many theories such as Fuzzy Logic (FL), Artificial Neural Networks (ANNs) and Genetic Algorithms (GAs). When used for intrusion detection, soft computing is a general term for describing a set of optimization and processing techniques that are tolerant of imprecision and uncertainty [9-11].

To overcome low detection rate and high false alarm problems in currently existing IDS, SOM (Self Organizing Map) Artificial Neural Network can be used to enhance the performance of intrusion detection for rare and complicated attacks. Unsupervised learning neural nets can be used to classify and visualize system input data to separate normal behaviours from abnormal or intrusive ones. Most of the systems in this category use Self-Organizing Maps (SOMs), while a few use other types of unsupervised neural nets. Fox was the first to apply an SOM to learn the characteristics of normal system activity and identify statistical variations from the normal trends [12].

In 2002, Labib and Vemuri [13] described an implementation of a real-time network-based intrusion detection system using self organization maps. In 2002 also, Lichodzijewski et al. [14] and Cortada et al. [15] and in 2003, Ramadas [16] tried to trained SOM on a collection of normal data from UNIX audit data and used it for detecting anomalous user activity. In 2005, Albayrak et al. [17] proposed approach focus on improving the usage of SOMs for anomaly detection by combining the strengths of different SOM algorithms. In 2006, Vokorokos et al. [18] presented intrusion detection systems and design architecture of intrusion based on neural network self organization maps. In 2007, Oksuz [19] in his thesis evolved around intrusion detection system (IDS) and neural networks. This thesis outlines an investigation on the unsupervised neural network models and choice one of them for implementation and evaluation. In 2011, Mahmood [20] established an anomaly intrusion detection system that detect intrusive activities using self-organizing map N.N and classify the attack by using the ant-miner algorithm. Also in 2011, Halema [21] built a misuse IDS using back-propagation networks and use self-organizing map to create an anomaly IDS.

Artificial Neural Network (ANN) consists of a collection of processing units called neurons that are highly interconnected in a given topology. ANNs have the ability of learning-by-example and generalization from limited, noisy, and incomplete data; they have, hence, been successfully employed in a broad spectrum of data intensive applications ,The property of dimensionality reduction and data visualization in neural networks can be very useful to reduce the many dimensions of a network connection to 2-dimension .

After make survey on researchers deal with intrusion detection system based on SOM Neural Network and all other methods of neural networks, most of these researchers used up to date dataset for intruders (KDD 99), and there is a new dataset of intruders (NSL-KDD).

There are a few number of researchers deal with NSL-KDD dataset, when they design IDS systems, For this reason we intend to build an effective intrusion detection system use Self -Organizing Map (SOM) neural network that detect attacks based on anomaly approach with the KDD99 and NSL_KDD data sets.

3. Architecture of Self Organization Map (SOM) Artificial Neural Network

The Self-Organizing Map is a competitive network where the goal is to transform an input data set of arbitrary dimension to a one- or two-dimensional topological map. SOM is partly motivated by how different information is handled in separate parts of the cerebral cortex in the human brain. The model was first described by the Finnish professor Teuvo Kohonenand is thus sometimes referred to as a Kohonen Map. The SOM aims to discover underlying structure, e.g. feature map, of the input data set by building a topology preserving map which describes neighbourhood relations of the points in the dataset [20].

The SOM is often used in the fields of data compression and pattern recognition. There are also some commercial intrusion detection products that use SOM to discover anomaly traffic in networks by classifying traffic into categories. The structure of the SOM is a single feed forward network [19], where each source node of the input layer is connected to all output neurons. The number of the input dimensions is usually higher than the output dimension. The algorithm tries to find clusters such that two neighbouring clusters in the grid have codebook vectors close to each other in the input space. Another way to look at this is that related data in the input data set are grouped in clusters in the grid [20].

The training utilizes competitive learning, meaning that neuron with weight vector that is most similar to the input vector is adjusted towards the input vector. The neuron is said to be the ’winning neuron’ or the Best Matching Unit (BMU). The weights of the neurons close to the winning neuron are also adjusted but the magnitude of the change depends on the physical distance from the winning neuron and it is also decreased with the time [22].

In this research the Self Organization Map SOM Artificial Neural Network is used to detect attackers. The 41 features from KDD99 and from NSL-KDD datasets are used as input data, SOM transforms 41-dimensional input data vector into 2 outputs vector (0 if entrance pattern is not an attack (Normal), and 1 values for attackers (abnormal). The SOM processes those given data to recognize type of attacks or normal transactions.

4. Proposed Intrusion Detection System

The proposed intrusion detection system (IDS) consists of three modules, as shown in Fig. 1

- Create database module
- Preprocessing database module.
- Detection module (Normal or abnormal packet).

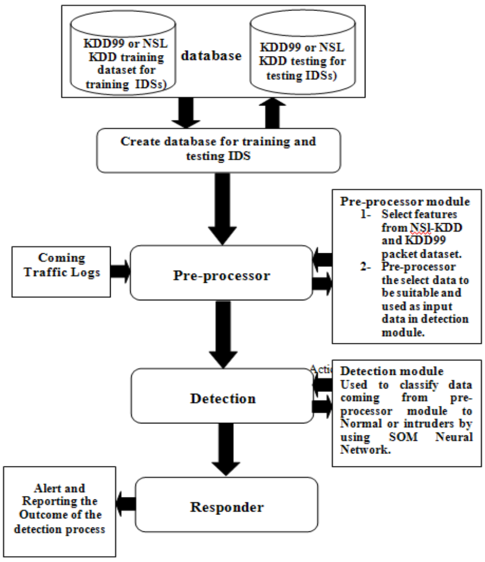
KDD99 or NSL KDD training dataset for training IDSs)
database
KDD99 or NSL KDD testing for testing IDSs)
Create database for training and testing IDS
Coming Traffic Logs
Pre-processor
Pre-processor module
1- Select features from NSl-KDD and KDD99 packet dataset.
2- Pre-processor the select data to be suitable and used as input data in detection module.
Detection
Acti
Detection module
Used to classify data coming from pre-processor module to Normal or intruders by using SOM Neural Network.
Responder
Alert and Reporting the Outcome of the detection process

Fig. 1. Intrusion Detection System.

4.1. Database module

The first module of proposed IDS is creating database module that means collects and formats the data to be analyzed by the detection algorithm. In proposed IDS we used two databases:-

I. **KDD99 Database (Knowledge Discovery in Databases)**: The KDD99 data is original from 1998 DARPA Intrusion Detection Evaluation. Under the sponsorship of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), MIT Lincoln Labs has collected and distributed the datasets for the evaluation of computer network intrusion detection system [20, 21, 23].

II. **NSL-KDD Database:** NSL-KDD is a dataset proposed by Tavallaee et al. [24]. NSL-KDD dataset is a reduced version of the original KDD 99 dataset. NSL-KDD consists of the same features as KDD 99. The KDD99 dataset consists of 41 features and one class attribute. The class attribute has 21 classes that fall under four types of attacks: Probe attacks, User to Root (U2R) attacks, Remote to Local (R2L) attacks and Denial of Service (DoS) attacks. This dataset has a binary class attribute. Also, it has a reasonable number of training and test instances which makes it practical to run the experiments on [25].

The NSL-KDD has the following differences over the original KDD 99 dataset [25-27]:

- It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records.
- There are no duplicate records in the proposed test sets; therefore, the performances of the learners are not biased by the methods which have better detection rates on the frequent records.

- The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD 99 data set.

The numbers of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

4.2. Preprocessing database module

SOM neural network is using only numerical data and in the same range to make SOM give an accurate result. For this reason, the proposed IDS create a preprocessing module to transform value of features of each packet from characters to numeric value After that, a Normalization process is performed on the numeric values to make it in the same range, the preprocessing module is done according to the following steps:

- **Step 1: Convert characters value to numeric values:** There are three futures in each packets have characters values (protocol type, Service, Flag), which must converted to numeric value by compute number of time each feature is repeated, then ascending feature according to its repeated time, like 1 give to the feature have a greater number of repeated time, 2 for the feature have less frequently, … etc., as shown in Table 1.

Table 1. Numeric Values of NSL-KDD Features.

Protocol type	Feature value	Service	Feature value	Flag	Feature value
ICMP	3*	HTTP	1	SF	1
ICMP	3*	HTTP	1	SF	1
TCP	1	Private	2	S0	2
UDP	2	.	.		.
		.	.		.
		HTTP_2748	70	OTH	11

(*) Because the TCP In Protocol Field is recurrence more than other protocol types likes (UDP, ICMP), it coded by 1 and according to descending order UDP coded by 2 and ICMP coded by 3, at the same way other fields (Service , Flag) are coded.

- **Step 2: Normalized numeric values**: As we mentioned in section 4.2, because SOM Neural Network using only numerical data and it must in the same rang to made SOM give an accurate results, normalization phase must do it on all features in each packets (see *Appendix A*), on KDD 99 and NSL-KDD dataset. To normalize numeric values to range between *MinX* and MaxX that are the minimum and maximum values for feature X, we first convert [*MinX*, *MaxX*] to new range [New *MinX*, New *MaxX*], according to Eq. (1) each value of *V* in the original range is converted to a new value [20]

$$NewV = \frac{V \text{ - } MinX}{MaxX - MinX} \tag{1}$$

4.3. Detection module

The most important component of the proposed IDS is the detection module whose function is to analyze and detect intrusion using Artificial Neural Network. Neural Network used as a detection module because of the utilization of a neural Network in the detection of Intrusion and flexibility that the network would provide. A Neural Network would be capable of analyzing data from the network, even if the data is incomplete or distorted. Similarly, the network would possess the ability to conduct an analysis with data in a non-linear fashion. Both of these characteristics are important in a networked environment where the information which is received is subject to the random failings of the system.

Further, because some attacks may be conducted against the network in a coordinated assault by multiple attackers, the ability to process data from a number of sources in a non-linear fashion is especially important. The inherent speed of Neural Networks is another benefit of this approach. Because the protection of computing resources requires the timely identification of attacks, the processing speed of the Neural Network could enable Intrusion responses to be conducted before irreparable damage occurs to the system [2]. In this project Self Organization Map (SOM) Artificial Neural Network is used as a detection module in the proposed IDS.

4.3.1. Structure of the proposed (SOM) neural network

The structure of the proposed Self Organization Map ANN is shown in Fig. 2, which indicates 41 input nodes with two output nodes.

4.3.2. SOM Algorithm [22] :

```
• Select output layer network topology
  Initialize current neighborhood distance, D(0), to a positive value
• Initialize weights from inputs to outputs to small random values
• Let $t$ = 1
• While computational bounds are not exceeded do

    1) Select an input sample $i_l$
    2) Compute the square of the Euclidean distance of $i_l$
       From weight vectors ($w_j$) associated with each output node

       $w_j = \sum_{k=1}^{n} (i_{l,k} - w_{i,k}(t))^2$          (2)

    3) Select output node $j*$ that has a weight vector with minimum
       value from step 2.

    4) Update weights to all nodes within a topological distance given by
       $D(t)$ from $j*$, using the weight update rule below:

       $w_j(t+1) = w_j(t) + \eta(t)(i_l - w_j(t))$          (3)

    5) Increment $t$
• End while
```

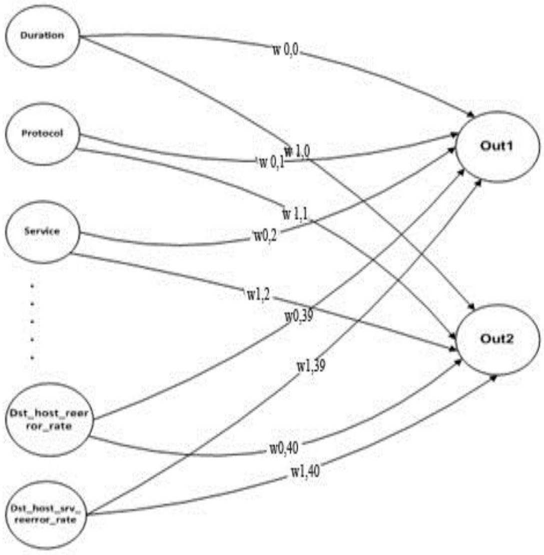

Fig. 2. Structure of Proposed (SOM) Neural Network.

5. Results and Discussion

In this section, we summarize our experimental results to detect Anomaly Intrusion Detections using SOM Artificial Neural Network over KDD99 dataset and NSL_KDD.

- For KDD99 the training dataset have 494021patterns, and testing dataset consist of 311029 patterns.
- For NSL-KDD the training dataset have 125973 patterns, and testing dataset consist of 22544 patterns.

We are only interested in knowing to which category (normal, abnormal) a given connection belongs. Four experiments are made on the proposed IDS on 4 computers using SOM as a detection module and KDD99 as a dataset, with changes on values of [Epoch, Initial Rate, and Changed Rate] parameters. The result of the training phase is shown in Table 2, whereas the result of Detection Rate in testing phase is explained in Table 3.

Table 2. Training Proposed IDS on KDD99 Dataset.

Experiment No.	Epochs	Initial Rate	Changed Rate	Normal Node	Training Time (H:m)
1	1000	0.9	0.7	Out1	1 : 40
2	200	0.9	0.2	Out 2	0 : 35*
3	1000	0.8	0.5	Out 2	1 : 40
4	5000	0.2	No.	Out 1	2 : 38

(*) the training time in Exp. No. 2 is less than other because the No. of Epochs is 200 Epochs while the No. of Epochsin Exp. (1,3,4) is more than or Equal to 1000 Epochs.

. **Table 3. Testing Detection Rate on KDD99 Dataset.**

Experiment No.	Normal Detection	Attack Detection	False Positive rate	Detection Rate	Testing Time (m:s)
1	56996	228475	5.93 %	91.78%	2:00
2	57719	229602	4.67 %	92.37%	2:00
3	59648	166454	1.55 %	72.69%	2:00
4	59703	166405	1.40 %	72.69%	2:00

Many experiments also are made on the proposed IDS on 4 computers using SOM as a detection module and NSL-KDD as a dataset, with changes on values of [Epoch, Initial Rate, and Changed Rate] parameters, the result of detection attackers in training phase is shown in Table 4, whereas the result of detection attackers in testing phase is shown in Table 5.

Table 4. Training Detection Rate on NSL-KDD Dataset.

Experiment No.	Epochs	Initial Rate	Changed Rate	Normal Node	Training Time (H:m)
1	1000	0.9	0.7	Out 1	1 :07
2	1000	0.9	0.2	Out 2	1 : 09
3	300	0.9	0.2	Out 2	0 : 20
4	1000	0.8	0.2	Out 2	1 : 05

Table 5. Testing Detection Rate on NSL-KDD Dataset.

Experiment No.	Normal Detection	Attack Detection	False Positive rate	Detection Rate	Testing Time (m:s)
1	8408	8603	5.77 %	75.49	0:55
2	5206	10104	19.9 %	67.19	0:55
3	6777	10633	13.01 %	77.23	0:55
4	8872	3665	3.7 %	55.61	0:55

After four experiments are made on the proposed IDS system based on KDD99 and NSL-KDD datasets, we found that IDS with KDD99, works in a good detection rate (92.37%) with 200 epochs, and 0.9 learning rate value. Change of learning rate in each epoch is 0.2.

For NSL-KDD dataset we found that IDS works with good detection rate (75.49 %) with 1000 epochs, and 0.9 learning rate. Change of learning rate in each epoch is 0.7. The Results mean that KDD99 is still the suitable database with any detection method, and NSL-KDD dataset is not a suitable dataset with SOM as a detection module.

NSL-KDD is still not perfect representative of existing real networks, because of the lack of public data sets for network-based IDS. We believe it still can be applied as an effective benchmark dataset to help researchers compare different intrusion detection methods.

Several recently published result and our results on the same datasets are listed in Table 6. We can find that our IDS are greatly competitive with the others and Fig. 3 indicates that our system has possibilities for detection computer attacks.

Table 6. Compression for Intrusion Detection Systems on KDD99 and NSL-KDD.

Research	ANN type	Database	% of Successful Detection Rate
Cannady,1998 [28]	MLFF	Real Secure™ network monitor	91%
Moradi,2004 [29]	2 hidden layers MLP	KDD99	91%
Siddiqui, 2004 [30]	Back propagation and fuzzy ARTMAP	KDD99	81.37% for BP and 80.52% for fuzzy ARTMAP (overall PSC = 80.945)
Sheikhan, 2009 [31]	Fuzzy AR	KDD99 (15000)	91 %
Sheikhan, 2009 [31]	K-NN	KDD99 (15000)	91 %
Sheikhan, 2009 [31]	Data mining	KDD99 (15000)	80 %
Panda, 2010 [32]	Multinomial Naïve Bayes + N2B	NSL-KDD	38.89 %
Proposed IDS	SOM	KDD99	92.37%
Proposed IDS	SOM	NSL-KDD	75.49%

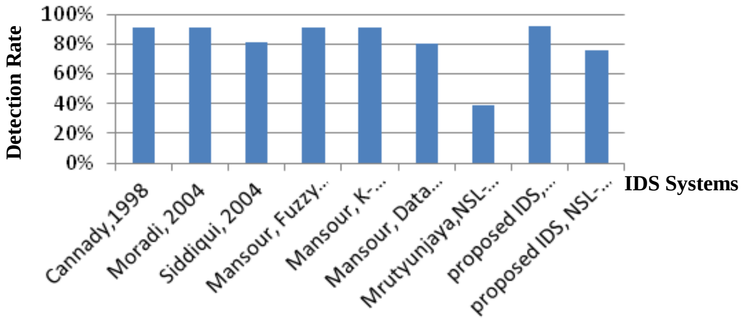

Fig. 3. Detection Rate of Proposed System Compared with IDS Systems using KDD99 and NSL-KDD Dataset.

6. Conclusions

In this research, we presented a practical solution of using unsupervised Artificial Neural Network in hierarchical Anomaly Intrusion Detection System. The system is able to employ SOM neural nets for detection and separate normal traffic from the attack traffic.

The proposed system was used to tuning, training, and testing SOM Neural Network in intrusion detection. Evaluation of the SOM efficiency in anomaly intrusion detection was performed detection performance. The results show that SOM with KDD99 is 92.37% able to recognize attack traffic from normal one, while with NSL-KDD is 75.49% able to recognize attack traffic from normal one.

Experiments on the KDD99 network intrusion dataset show that SOM are best suited due to their high speed and fast conversion rates as compared with other learning techniques. SOM are more powerful than static networks because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns. It is also shown that our approach using SOM obtains superior performance in comparison with other state-of-the-art detection methods.

Experiments on the NSL-KDD show that NSL-KDD is still not perfect representative of existing real networks. In the future, we will hope to detect attackers, combine Artificial Neural Network methods to improve the accuracy of IDS on NSL-KDD.

Acknowledgments

Researchers want to thank and appreciate students; Khalil A. Ibrahim, Mohammad F. Natheer and Anmar A. Younis in the Department of software engineering - University of Mosul to help them to collected (NSL_KDD, KDD99) dataset and spending time and effort in the training of artificial neural network.

References

1. Kemmerer, R.A.; and Vigna, G. (2002). Intrusion detection a brief history and overview. *Computer*, 35(4), 27-30.
2. Chen, W.-H.; Hsu, S.-H.; and Shen, H.-P. (2005). Application of SVM and ANN for intrusion detection. *Computers & Operations Research*, 32(10), 2617-2634.
3. Lorenzo-Fonseca, I.; Maciá-Pérez, F.; Mora-Gimeno, F.J.; Lau-Fernández1, R.; Gil-Martínez-Abarca, J.A.; and Marcos-Jorquera, D. (2009). Intrusion detection method using neural networks based on the reduction of characteristics. *Lecture Notes in Computer Science*, Volume 5517, 1296-1303.
4. Zhang, C.; Jiang, J.; and Kamel, M. (2005). Intrusion detection using hierarchical neural networks. *Pattern Recognition Letters*, 26(6), 779-791.
5. Peddabachigari, S.; Abraham, A.; Grosan, C.; and Thomas, J. (2007). Modelling intrusion detection system using hybrid intelligent systems. *Journal of Network and Computer Applications*, 30(1), 114-132.
6. Balzarotti, D. (2006). *Testing network intrusion detection systems*. Ph.D. Dissertation, Politecnico di Milano, Italy.
7. Bace, R.; and Mell, P. (2001). *Intrusion detection systems*. NIST Special Publication on IDS, 1-51.
8. Brown, D.J.; Suckow, B.; and Wang, T. (2001). *A survey of intrusion detection systems*. http://charlotte.ucsd.edu/classes/fa01/cse221/projects/group10.pdf .
9. Mukkamala, S. (2002). Intrusion detection using neural networks and support vector machine. *Proceedings of the* 2002 *IEEE International Joint Conference on Neural Networks. Honolulu*.
10. Sammany, M.; Sharawi, M.; El-Beltagy, M.; and Saroit, I. (2007). Artificial neural networks architecture for intrusion detection systems and classification of attacks. Cairo University.http://infos2007.fci.cu.edu.eg/Computational%20 Intelligence/07177.pdf.

11. Selvakani, S.; and Rajesh, R.S. (2009). Escalate intrusion detection using GA–NN. *International Journal of Open Problems in Computer Science and Mathematics*, 2(2), 272-284.
12. Fox, K.L.; Henning, R.R.; Reed, J.H.; and Simonian, R. (1990). A neural network approach towards intrusion detection. *In Proceedings of the* 13*th* *National Computer Security Conference*, 10.
13. Labib, K.; and Vemuri, R. (2002). NSOM: A real-time network-based intrusion detection system using self-organizing maps. *Technical Report*, Department of Applied Science, University of California, Davis.
14. Lichodzijewski, P.; Zincir-Heywood, A.N.; and Heywood, M.I. (2002). Dynamic intrusion detection using self-organizing maps. In *the* 14*th* *Annual Canadian Information Technology Security Symposium* (*CITSS*).
15. Cortada, P.; Sanroma, G.; and Garcia, P. (2002). IDS based on self-organizing maps. *Technical Report*, Red IRIS.
16. Ramadas, M. (2003). Detecting anomalous network traffic with self-organizing maps. MSc. Thesis, Faculty of the College of Engineering and Technology, Ohio University.
17. Albayrak, S.; Scheel, C.; Milosevic, D.; and Muller, A. (2005). Combining self-organizing map algorithms for robust and scalable intrusion detection. In *Proceedings of International Conference on Computational Intelligence for Control and Automation, International Conference on Intelligent Agents, Web Technologies and Internet Commerce,* (*CIMCA-IAWTIC* 06) - Volume 02.
18. Vokorokos, L.; Baláž, A.; and Chovanec, M. (2006). Intrusion detection system using self organizing map. *Acta Electrotechnica et informatica*, 1(6), 1-6.
19. Oksuz, A. (2007). *Unsupervised intrusion detection system*. MSc. Thesis, Informatics and Mathematical modelling, DTU, Richard Petersens Plads, Kongens, Lyngby.
20. Mahmood, S. M. (2011). *Using ant and self-organization maps algorithms to detect and classify intrusion in computer networks*. MSc. Thesis, University of Mosul.
21. Halema, I.M. (2011), *Development network intrusion detection system by using neural networks*. MSc. Thesis, University of Mosul.
22. http://genome.tugraz.at/MedicalInformatics2/SOM.pdf.
23. Alzobaidy, L. (2010). Anomaly network intrusion detection system based on distributed time-delay neural network (DTDNN). *Journal of Engineering Science and Technology* (*JESTEC*), 5(4), 457-471.
24. Tavallaee, M.; Bagheri, E.; Wei Lu; and Ghorbani, A. (2009). A detailed analysis of the KDD CUP 99 data set. *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications* (*CISDA* 2009), 1-6.
25. Shaheen, A. (2010). *A comparative analysis of intelligent techniques for detecting anomalous internet traffic*. MSc. Thesis, King Fahd University.
26. Eid, H.F.; Darwish, A.; Hassanien, A.E.; and Abraham, A. (2010). Principle components analysis and support vector machine based intrusion detection system. In the *Proceedings of* 10*th* *International Conference on Intelligent Systems Design and Applications* (*ISDA* 2010), Cairo, Egypt.

27. Datti, R.; and Verma, B. (2010). Feature reduction for intrusion detection using linear discriminant analysis. (*IJCSE*) *International Journal on Computer Science and Engineering*, 2(4), 1072-1078.

28. Cannady J. (1998). Artificial neural networks for misuse detection. *Proceedings of the 1998 National Information Systems Security Conference (NISSC'98)*, 443-456, Arlington, VA.

29. Moradi, M.; and Zulkernine, M. (2004). A neural network based system for intrusion detection and classification of attacks. *IEEE International Conference on Advances in Intelligent Systems - Theory and Applications*, Luxembourg-Kirchberg, Luxembourg.

30. Siddiqui, M.A. (2004). *High performance data mining techniques for intrusion detection*. MSc. Thesis, University of Engineering & Technology, School of Computer Science, College of Engineering & Computer Science at the University of Central Florida.

31. Sheikhan, M.; and Gharavianm, D. (2009). Combination of Elman neural network and classification-based predictive association rules to improve computer networks security, *World Applied Sciences Journal*, 7, *Special Issue of Computer & IT*, 80-86

32. Panda, M. (2010). Discriminative multinomial Naïve Bayes for network intrusion detection. 2010 *Sixth International Conference on Information Assurance and Security* (*IAS*), 5-10.

Appendix A

Features in Each Packet

feature	**duration**	**protocol_type**	**service**	**flag**	**src_bytes**
First step of preprocessing	0	tcp	http	SF	181
feature	dst_bytes	land	wrong_fragment	urgent	hot
First step of preprocessing	5450	0	0	0	0
feature	num_failed_logins	logged_in	num_compromised	root_shell	su_attempted
First step of preprocessing	0	1	0	0	0
feature	num_root	num_file_creations	num_shells	num_access_files	num_outbound_cmds
First step of preprocessing	0	0	0	0	0
feature	is_host_login	is_guest_login	count	srv_count	serror_rate
First step of preprocessing	0	0	8	8	0.00
feature	srv_serror_rate	rerror_rate	srv_rerror_rate	same_srv_rate	diff_srv_rate
First step of preprocessing	0.00	0.00	0.00	1.00	0.00
feature	srv_diff_host_rate	dst_host_count	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate
First step of preprocessing	0.00	0.00	9	0.00	0.00
Target data	Target data				
First step of preprocessing	Normal				