Intrusion Detection using Multi-Stage Neural Network

Sahar Selim, Mohamed Hashem and Taymoor M. Nazmy
Faculty of Computer and Information Science
Ain Shams University
Cairo, Egypt

Abstract— **Security has become a crucial issue for computer systems. New security failures are discovered everyday and there are a growing number of bad-intentioned people trying to take advantage of such failures. Intrusion detection is a critical process in network security. Intrusion Detection Systems (IDS) aim at protecting networks and computers from malicious network-based or host-based attacks. This paper presents a neural network approach to intrusion detection. We compare the use of our proposed multi-stage to single-stage neural network for intrusion detection using single layer perceptron. The advantage of the proposed mutli-stage system is not only accuracy but also the parallelism as every network can be trained on separate computer which provides less training time. Also the multi-stage powers the system with scalability because if new attacks of specific class are added we don't have to train all the networks but only the branch (the neural networks) affected by the new attack. The results showed that the designed multi-stage network is capable of classifying records with 99.71% accuracy and 98.67% accuracy for single stage network.**

Keywords-component; network intrusion detection; neural network, NSL-KDD dataset

I. INTRODUCTION

The rapid development and expansion of World Wide Web and local network systems have changed the computing world in the last decade. The costs of temporary or permanent damages caused by unauthorized access of the intruders to networks and computer systems have urged different organizations to, increasingly; implement various systems to monitor data flow in their networks. These systems are generally referred to as Intrusion Detection Systems (IDSs) [1].

There exist two main types of network intrusion detection methods: anomaly-based and misuse-based. Misuse detection methods, uses well-defined patterns of the attack that exploit weaknesses in the system and application software to identify the intrusions. A characteristic trait of the intrusion is developed offline, and then loaded in the intrusion database before the system can begin to detect this particular intrusion. It has drawbacks: firstly in most systems, all new attacks will go unnoticed until the system is updated (i.e. they cannot detect new attacks that have never occurred in the training data), creating a window of opportunity for attackers to gain control of the system under attack. Secondly, only known attacks can be detected [2].

Anomaly-based systems (ABS), on the other hand, build statistical models that describe the normal behavior of the network, and flags any behavior that significantly deviates from the norm as an attack. This has the advantage that new attacks will be detected as soon as they take place [3].

II. PREVIOUS WORK

An increasing amount of research has been conducted on the application of neural networks for detecting network intrusions. The idea behind the application of soft computing techniques and particularly ANNs in implementing IDSs is to include an intelligent agent in the system that is capable of disclosing the latent patterns in abnormal and normal connection audit records, and to generalize the patterns to new (and slightly different) connection records of the same class [4].

There are researches implement an IDS using MLP which have the capability of detecting normal and attacks connection as in [5], [6], [7]. They are implemented using MLP of three and four layer neural network. References [8], [4] used three layers MLP (two hidden layers) not only for detecting normal and attacks connection but also identify attack type.

Neural Network was also used for dimension reduction of features as in [9]. The SOM was also applied to perform the clustering of network traffic and to detect attacks in [10], [11], [12] and [13]. In [14], self-organizing maps was used for data clustering and MLP neural networks for detection.

Most of the previous studies that used MLP were implemented with at least three layers. Our study use MLP with no hidden layer to perform less complicated network structure and decrease the computation time. The idea of this study is based on the combination of both ideas which are to be able to identify normal and attack records without exhausting the network of identifying attack type to get higher accuracy and also being able to detect attack type by the next levels. This approach has the advantage to flag for suspicious record even if attack type of this record wasn't identified correctly.

III. DATASET DESCRIPTION

KDDCUP'99 is the mostly widely used data set for the evaluation of these systems. The KDD Cup 1999 uses a version of the data on which the 1998 DARPA Intrusion Detection Evaluation Program was performed. They set up an environment to acquire raw TCP/IP dump data for a local-area network (LAN) simulating a typical U.S.Air Force LAN.

A. Types of Networking Attacks

There are four major categories of networking attacks. Every attack on a network can be placed into one of these groupings [15].

1) Denial of Service Attack (DoS): is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. e.g. apache, smurf, Neptune, ping of death, back, mail bomb, UDP storm,etc.

2) User to Root Attack (U2R): is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system. e.g. xlock, guest, xnsnoop, phf, sendmail dictionary etc.

3) Remote to Local Attack (R2L): occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine. e.g. perl, xterm.

4) Probing Attack: is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls. e.g. satan, saint, portsweep, mscan, nmap etc.

There are some inherent problems in the KDDCUP'99 data set [16], which is widely used as one of the few publicly available data sets for network-based anomaly detection systems. The first important deficiency in the KDD data set is the huge number of redundant records. Analyzing KDD train and test sets, it was found that about 78% and 75% of the records are duplicated in the train and test set, respectively. This large amount of redundant records in the train set will cause learning algorithms to be biased towards the more frequent records, and thus prevent it from learning infrequent records which are usually more harmful to networks such as U2R attacks. The existence of these repeated records in the test set, on the other hand, will cause the evaluation results to be biased by the methods which have better detection rates on the frequent records [15].

The data in the experiment is acquired from the NSL-KDD dataset which consists of selected records of the complete KDD data set and does not suffer from mentioned shortcomings by removing all the repeated records in the entire KDD train and test set, and kept only one copy of each record [15]. Although, the proposed data set still suffers from some of the problems discussed by McHugh [17] and may not be a perfect representative of existing real networks, because of the lack of public data sets for network-based IDSs, but still it can be applied as an effective benchmark data set to help researchers compare different intrusion detection methods. The NSL-KDD dataset is available at [18].

IV. Proposed Multi-Stage Neural Network

A. Dataset

In this study we examine using two attacks from each DOS and Probe classes to check the ability of the intrusion detection system to identify attacks from different categories. The sample dataset contains 20000 record for training (10000 normal and 2500 for each attack type) and 1200 for testing (600 normal and 150 for each attack type).

B. System Architecture

The proposed system architecture is shown in Fig. 1. The input data are preprocessed. The data must be of uniform representation to be processed by the neural network.

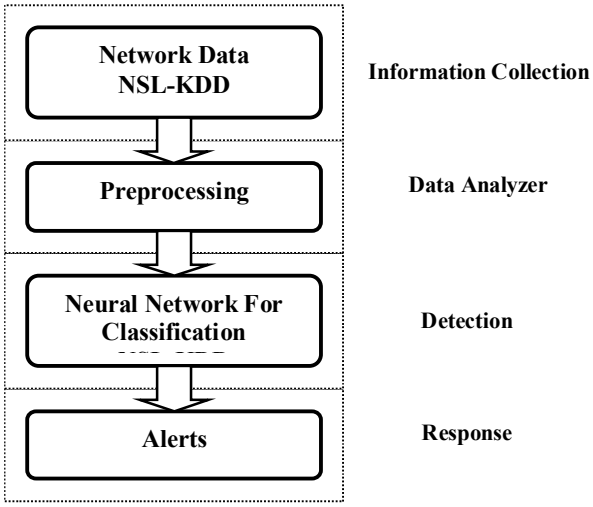

Figure 1. System Architecture.

1) Information Collection: The first module is responsible for data collection. We use the NSL-KDD dataset.

2) Data Analyzer: The second module is for preprocessing.

The preprocessing phase: Features selection, Numerical Representation and Normalization

a) Dimension reduction by excluding the features that are constantly zero over all data records. Hence the data vector is reduced to 30 dimensional vectors.

b) Converts non-numeric features into a standardized numeric representation. This process involved the creation of relational tables for each of the data type and assigning number to each unique type of element. (e.g. protocol_type feature is encoded according to IP protocol field: TCP=0, UDP=1, ICMP=2). This numerical representation was necessary because the feature vector fed to the input of the neural network has to be numerical.

c) It is important to shuffle examples before training so that the network weights are not biased towards a specific attack.

d) The ranges of the features were different and this made them incomparable. Some of the features had binary values where some others had a continuous numerical range

(such as duration of connection). As a result, the features were normalized by mapping all the different values for each feature to [0, 1] range.

3) Detection

We use neural network for classification. We compare between the proposed multi-stage neural module and single-stage neural network.

a) Multi-stage Neural Network

Attacks of the same class have a defined signature which differentiates between attacks of every class/category from others, i.e. DOS attacks have similar characteristics which identifies them from attacks of Probing. That's why there's often misclassification between attacks of the same class. For that reason, we thought of making a multi-stage neural network consisting of three levels as shown in Fig 2:

- **Level 1**: is a Neural Network that identifies attacks from normal
- **Level 2:** is a Neural Network that identifies classes
- **Level 3:** is a neural network that specify attack type

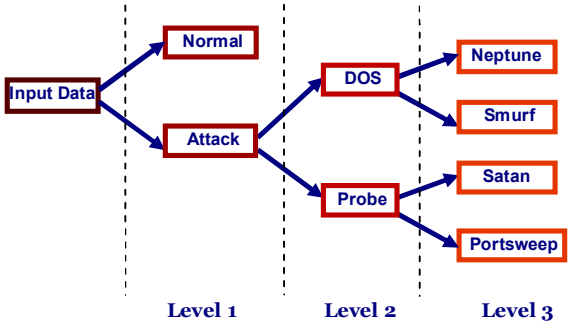

Figure 2. Multi-stage Levels.

The data is input in the first level which identifies if this record is a normal record or attack without exhausting the network to identify the attack name. If the record is identified as an attack then the module would raise a flag to the administrator that the coming record is an attack then the module inputs this record to the second level which identifies the class of the coming attack. If record was classified by network II to be DOS then it would be entered to the DOS network of the third level that identify attacks' type of DOS otherwise it would be introduced to the Probe network. The idea is that if ever the attack name of the third level is misclassified then at least the admin was identified that this record is suspicious after the first level network. Finally the admin would be alerted of the suspected attack type to guide him for the suitable attack response.

1. *Level 1 Architecture:* Neural Network that identifies attacks from normal as shown in Fig. 3.

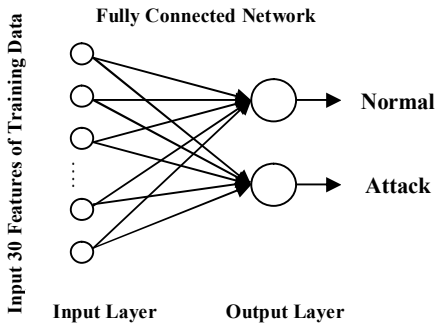

Figure 3. First Level Network which differentiate between Normal and Attack.

2. *Level 2 Architecture:* Neural Network that identifies classes DOS and Probe as shown in Fig. 4.

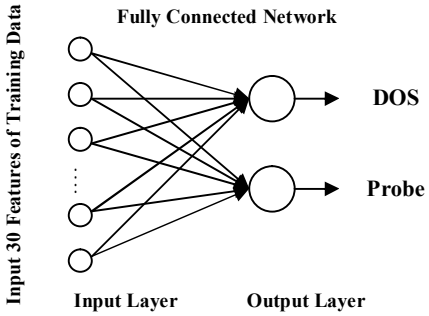

Figure 4. Single Layer Perceptron of Second Level Network which Classify the Attack Class DOS or Probe

3. *Level 3 Architecture:* Neural network that specify attack type

ATTACK TYPE OF DOS CLASS WHETHER NEPTUNE OR SMURF AS SHOWN IN FIG. 5.

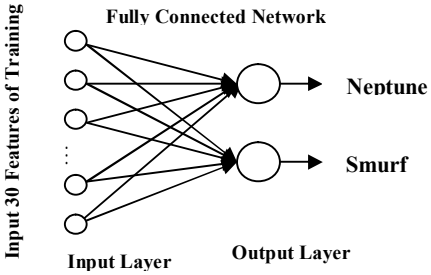

Figure 5. Single Layer Perceptron of third Level Network which Classify Attack type of DOS category.

ATTACK TYPE OF PROBE CLASS WHETHER SATAN OR PORTSWEEP AS SHOWN IN FIG. 6.

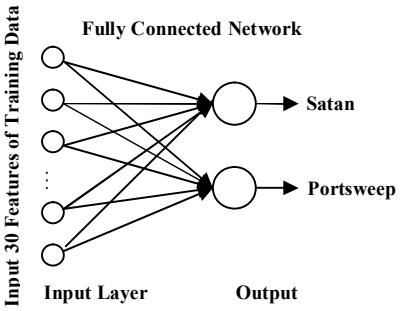

Figure 6. Third Level Network Single Layer Perceptron which Classify Attack type of Probe category.

b) Single Stage Neural Network

In this experiment we examine the use of the neural network for classifying normal and attack type, which means that we input the record and let the MLP identifying the normal and specify the attack name as shown in Fig. 7.

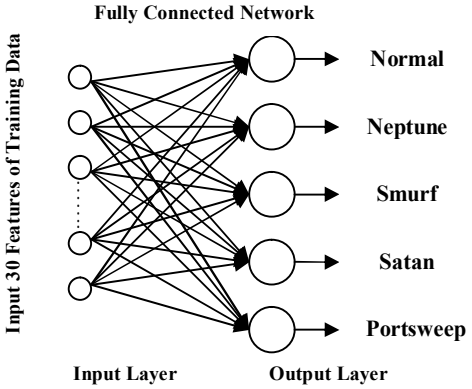

Figure 7. Single-Stage Single Layer Perceptron Network which Classify Normal and Attack type

C. The Over-fitting Problem

One problem that can occur during neural network training is over-fitting. In an over fitted ANN, the error (number of incorrectly classified patterns) on the training set is driven to a very small value, however, when new data is presented, the error is large. In these cases, the ANN has memorized the training examples; however, it has not learnt to generalize the solution to new situations. One possible solution for the over-fitting problem is to find the suitable number of training epochs by trial and error which isn't reasonable for cases that which takes too much time in training. A more reasonable method for improving generalization is called early stopping. In this technique, the available data is divided into three subsets. The first subset is the training set, which is used for training and updating the ANN parameters. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training similar to the training set error. However, when the ANN begins to over-fit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights that produced the minimum error on the validation set are retrieved [19]. In the present study, this training-validation strategy was used in order to maximize the generalization capability of the ANN.

D. Performance Measures

To evaluate our system we used two major indices of performance. We calculate the detection rate and the false alarm rate according to [20] the following assumptions:

- FP: the total number of normal records that are classified as anomalous
- FN: the total number of anomalous records that are classified as normal
- TN: the total number of normal records
- TA: the total number of attack records
- Detection Rate = [(TA-FN) / TA]*100
- False Alarm Rate = [FP/TN]*100

V. Experiments and Results

A. Training of Neural Network

This research aims to examine the difference between a multi-stage MLP and single-stage MLP. Also one of the objectives of the present study is to evaluate the possibility of achieving the same results with this less complicated neural network structure. Using a less complicated neural network is more computationally efficient. Also it would decrease the training time. Therefore we use a single layer perceptron with no hidden layers for all the networks in the two experiments. For each network 20% of the training data were set for cross validation. Early stopping criterion for validation set was applied to stop the training process to prevent over-fitting.

1) Training multi-stage Neural Network

All the 3 levels are a single layer perceptron feed-forward networks (which is the output layer as the input layer contains no processing so it's not considered a layer) with softmax activation function which output results of summation equal to one.

The output layer of first level consists of two neurons one for normal and other for attack. The training process was stopped with mean square error equal 0.0015 at 10000 epochs.

The output layer of second level consists of two neurons one for DOS and other for Probe. The training process was stopped with mean square error equal to 0.000672 at 7914 epochs.

There are two networks in level three. The first one contains two neurons one for Neptune and the other for smurf. The training process is stopped with mean square error equal to 0.000001 at 1574 epochs.

The second network of level three consists of 2 neurons one for satan and the other for portsweep. The training process was terminated with performance 0.00233 at 5838 epochs.

2) Multi-stage Neural Network Testing Results:

a) Level 1 Testing

The testing phase resulted in success rate 99.83 with error rate 0.167. Table I shows Correct Classification Rate for each of the 2 classes (Attack-Normal) and the total average classification accuracy.

TABLE I. LEVEL 1 CLASSIFICATION RESULTS

Class Name	Training Set	Testing Set
Normal	99.48	99.67
Attack	99.99	100
Average Success Rate	99.74	99.83
Error Rate	0.265	0.167

b) Level 2 Testing

The testing phase resulted in success rate 100. Table II shows the Correct Classification Rate for each of the 2 classes of Level 2 and the total average classification accuracy.

TABLE II. LEVEL 2 CLASSIFICATION RATE

Class Name	Training Set	Testing Set
DOS	99.95	100
Probe	99.77	100
Average Success Rate	99.86	100
Error Rate	0.14	0

c) Level 3 Testing

The testing phase resulted in success rate 99.5 with error rate 0.5. Table III shows the Correct Classification Rate for each of the 4 classes and the total average classification accuracy.

TABLE III. LEVEL 3 CLASSIFICATION RATE

Level 3 Networks	Class Name	Correct Classification	
		Training Set	***Testing Set***
DOS Network	Neptune	100	100
	Smurf	100	100
Probe Network	Satan	100	98.67
	Portsweep	100	99.33
Average Success Rate		100	99.5
Error Rate		0	0.5

3) Training single-stage Neural Network

This network is a single layer feed-forward networks with SoftMax activation. The output layer of this network consists of 5 neurons (normal, Neptune, Smurf, Satan, Portsweep). The training process was terminated with mean square error equal to 0.00034 at 12078 epochs.

4) Single-Stage Neural Network Testing Results

The testing phase resulted in success rate 98.8 with error rate 1.2. Table IV shows the Correct Classification Rate for each of the 5 classes and the total average classification accuracy of the single-stage neural network.

TABLE IV. SINGLE-STAGE CLASSIFICATION RATE

Class Name	Training Set	Testing Set
Normal	99.4	99.33
Neptune	100	99.33
Smurf	99.8	100
Satan	100	100
Portsweep	99.85	94.67
Average Success Rate	99.81	98.67
Error Rate	0.19	1.2

B. Discussion

Building all the networks with a single layer perceptron with no hidden layers gave the advantage of less computation time and less complicated network. The experimental results show that using a multi-stage neural network is more promising than single-stage network as shown in following tables and figures. Table V shows the Correct Classification Rate of testing dataset for each of the 5 classes for both Multi-stage and single-stage.

TABLE V. CLASSIFICATION RATE OF MULTI-STAGE AND SINGLE-STAGE

Class Name	Multi-Stage	Single-Stage
Normal	99.67	99.33
Neptune	100	99.33
Smurf	100	100
Satan	98.67	100
Portsweep	99.33	94.67

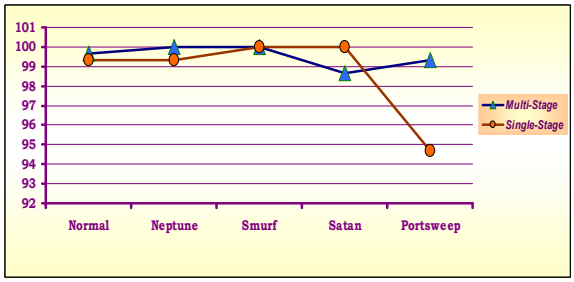

Figure 8. Comparison between Multi-Stage and Single-Stage

TABLE VI. FALSE ALARM COMPARISON

Method	Multi-Stage	Single-Stage
FP	2	3
FN	0	7
TN	600	600
TA	600	600
Detection Rate	100	98.83
False Alarm Rate	0.33	0.5

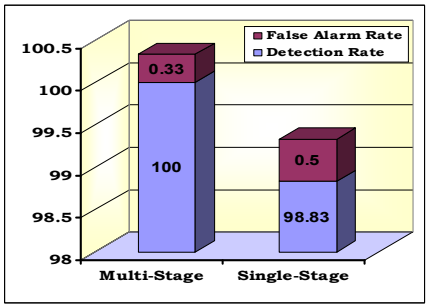

Figure 9. Detection and False Alarm Rate of Multi-Stage and Single-Stage

VI. CONCLUSION

In this paper we develop a multi-stage neural network and compare its results to results of single-stage neural network. The proposed multi-stage neural network consists of three detection levels. The network data are introduced to the network of the first level which aims to differentiate between normal and attack without exhausting the network in identifying the attack name. If the input record was identified as an attack then the administrator would be alarmed that the coming record is suspicious and then this suspicious record would be introduced to the second level which specifies whether this attack is DOS or probe. The similar characteristics between the attacks of the same class that often results in misclassification between attacks of same class gave the importance of the second level that we have at least identified the class type of the coming attack. The third detection level consists of two networks one to identify attacks of denial of service and the other for probe attacks. Finally the administrator would be alarmed of the expected attack type. The second experiment is for a single stage where the input is classified as one of the 5 classes (normal, Neptune, Smurf, Satan, Portsweep).The results show that the designed multi-stage system has detection rate equal to 100% while the single stage network has detection rate equal to 98.83. The advantage of the proposed mutli-stage system is not only higher accuracy but also the parallelism as every network can be trained on separate computer which provides less training time. Also the multi-stage powers the system with scalability because if new attacks of specific class are added to the dataset we don't have to train all the networks but only the branch (the networks) affected by the new attack.

VII. FUTURE WORK

The use of multi-stage network gives the opportunity of making hybrid neural network where each level can be trained by different algorithm. Also we can use larger dataset and adding attacks of U2R and R2L classes.

REFERENCES

[1] R. A. Kemmerer and G. Vigna, "Intrusion Detection: A Brief Introduction and History," Security & Privacy, IEEE Computer Magazine, pp. 27-30, 2002.

[2] Bolzoni, D., E. Zambon, S. Etalle, and P. Hartel, "POSEIDON: a 2-tier Anomaly-based Network Intrusion Detection System," in Proceedings of the 4th IEEE International Workshop on Information Assurance (IWIA), pp. 144-156, IEEE Computer Society Press, 2006.

[3] D. Bolzoni, S. Etalle, P. Hartel and E. Zambon "POSEIDON: a 2-tier Anomaly-based Network Intrusion Detection System," 2006.

[4] Mohammed Sammany, Marwa Sharawi, Mohammed El-Beltagy, Imane Saroit, "Artificial Neural Networks Architecture For Intrusion Detection Systems and Classification of Attacks," Cairo University, Egypt, 2007.

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

[6] J. Ryan, M. Lin, and R. Miikkulainen, "Intrusion Detection with Neural Networks," AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop, Providence, RI, pp. 72-79, 1997.

[7] Srinivas Mukkamala, "Intrusion detection using neural networks and support vector machine," Proceedings of the 2002 IEEE International Honolulu, HI, 2002.

[8] M. Moradi, and M. Zulkernine, "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, November 15-18, 2004.

[9] Y. Bouzida, F.e.e. Cuppens, N. Cuppens-Boulahia, S. Gombault, "Efficient intrusion detection using principal component analysis," in: Proceedings of the 3ème Conférence sur la Sécurité et Architectures Réseaux (SAR), Orlando, FL, USA, 2004.

[10] L. Girardin, "An eye on network intruder-administrator shootouts," In Proceedings of the Workshop on Intrusion Detection and Network Monitoring (ID'99), pages 19–28, Berkeley, CA, USA, 1999. USENIX Association.

[11] M. Ramadas, S. Ostermann, and B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," In Recent Advances in Intrusion Detection, 6th International Symposium, RAID 2003, pages 36–54, 2003.

[12] S. Zanero, "Improving Self Organizing Map Performance for Network Intrusion Detection," International Workshop on Clustering High-Dimensional data and its applications, SDM 05 SIAM conference On Data Mining, page. 30-37, 2005.

[13] P. Lichodzijewski, A. N. Zincir-Heywood, M. I. Heywood, "Dynamic intrusion detection using self-organizing maps," Proceedings of the 14th Annual CITASS, Ottawa, Canada, May 2002.

[14] A. Bivens, C. Palagiri, R. Smith, B. Szymanski and M. Emrechts, "Network-Based Intrusion Detection Using Neural Networks," Intelligent Engineering Systems through Artificial Neural Networks, Vol. 12, Proc. ANNIE, 2002.

[15] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[16] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, October 2007

[17] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.

[18] "Nsl-kdd data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/NSL-KDD/, March 2009

[19] MATLAB online support: www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml

[20] S.T. Sarasamma, Q.A. Zhu, and J. Huff, "Hierarchal Kohonenen Net for Anomaly Detection in Network Security," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 35(2), 2005, pp. 302-312.