Military Technical College
Kobry El-Kobbah
Cairo, Egypt

10th International Conference
On Aerospace Sciences&
Aviation Technology

A NETWORK BASED INTRUSION DETECTION MODEL USING NEURAL NETWORK

Prof. Dr. Mohamed S. Ibrahim* Dr. Ismail A. Taha*
Eng. Housam Shaban Al-Aloun*

ABSTRACT

Intrusion detection systems (IDS) have become an essential issue for computer networks security since each one is vulnerable for violation. This paper presents a neural network based implementation of an intrusion detection system to detect network based attacks. The key idea is to extract the most useful set of features from the packets traversing through the network and utilize them to describe users behavior. These selected features will be used an input features to train a designed neural network architecture to build a classifier that can recognize anomalies and known intrusions. Using a benchmark data set from a KDD (Knowledge Discovery and Data Mining), the designed system was able to correctly detect 99.8% of unusual network activity with a maximum of 5.4% false alarms. In addition, the system was 98.6% accurate in detecting different intrusion types.

Key Words: Intrusion Detection, Computer Security, Network Based Intrusion Detection, and Artificial Neural Networks.

* Egyptian Armed Forces.

1. INTRODUCTION

The tremendous expansion of computers and computer networks make it a target of computer crime more and more often. The process of identifying that an intrusion has been attempted, will occur, is occurring, or has occurred is regarded as intrusion detection. Intrusion detection system takes either a network or a host as a target to recognize and deflect attacks. The network based intrusion detection systems use a raw network's packets as a data source [1-4], while the host based intrusion detection systems look for attack's signature in log files [5-12].

Intrusion detection system's analysis approach can be divided into two categories, anomaly and misuse detection. The first one deals with the detection of a certain anomaly in a user behavior. Since each user has a certain functionality within the system, then this functionality could be observed and usually do not change a lot over time. This means that it is possible to define a set of actions usually performed by a user. This set is called user profile that describes user's normal behavior. After such profiles are defined, they can be managed to trace current user behavior and search for some deviations from it. Such deviations are called anomaly and indicate in most cases as intrusion [13,14]. While the basic principle of the second approach, misuse detection, is that any intrusion can be described in terms of its indications and signs. Patterns of all known attacks must be described in some abstracted forms and given to the intrusion detection systems, these patterns are used by IDS to identify an intrusion [5,7,15].

2. INTRUSION DETECTION SYSTEMS

Essentially, all intrusion detection systems have the same strategy; long-term scheduler compared with short-term scheduler, if the deviation exceeds predefined threshold, intrusion flag must be raised. Many methods are created to implement this strategy; each one has its advantages and disadvantages. This section introduces the most well known intrusion detection methods and approaches.

- ***State Transition Analysis Approach:*** Intrusion is seen as a sequence of intruder actions that bring the system from an initial state to a compromised state through a number of intermediate states [5].

- ***Statistical Approach:*** detects possible system intrusion by identifying departure from historical established normal behavior. User or system behavior is measured by a number of variables sampled over time and stored in a profile [3,6,16].

- ***System Calls Approach:*** each process is represented by its trace, the ordered list of the system calls used by any process form the beginning of its execution to the end, so the program's normal behavior could be characterized by local pattern in its traces and deviation from these patterns could be used to identify security violations of an executing process [12].

- ***Expert System Approach:*** *an* expert system detect intrusion by encoding intrusion scenario as a set of rules, these rules reflect the partially ordered sequence of actions that comprise the intrusion scenario [8,13].

- ***Data Mining Approach:*** Data mining is, at its core, pattern finding. Data miners are experts at using special software to find regularities and irregularities in large data set. The goal of this operational method is to have all alarms reviewed by human analysts [10,11].

- ***Model Based Approach:*** There is a database of attack scenarios, where each scenario comprises a sequence of behaviors making up the attack. At any moment the system considers a subset of these scenarios as likely ones being experienced by the system. It seeks to verify them by seeking information in the audit trail to substantiate or refute the attack scenario [15].

- ***Neural network Approach:*** in this approach a neural network is used to be trained to learn the normal behavior and attack patterns, then significant deviations from normal behavior are flagged as attacks [17-20]. The system presented in this work is network-based intrusion detection system using Muli-Layer Perceptron (MLP) Back-Propagation (PB) neural network. The neural network is first designed then trained with normal user activity and attack patterns. The data used in the implementation of the proposed neural network intrusion detection system is originated from MIT's Lincoln Labs. It was developed for KDD competition by Defense Advanced Research Projects Agency DARPA and is considered a standard benchmark for intrusion detection evaluations [21].

3. INTRUSION DETECTION PARAMETERS

When no intrusion occurs; a normal user, the intrusion detection system must reject to generate an alarm. This case is considered as a system rejection to generate an alarm. But, when an intrusion occurs, the intrusion detection system must not reject to generate an alarm. The following parameters are usually used as industry standards to measure how good is the generalization of the developed IDS. These parameters are:

- **Correct alarm:** an intrusion has occurred and the IDS has generated an alarm. Based on this parameter, the correct classification rate can be computed.

- **Correct rejection:** no intrusion has occurred and the IDS has not detected an intrusion. Based on this parameter, the miss classification rate can be computed.

- **False alarm:** no intrusion has occurred and the IDS has detected an intrusion. Based on this parameter, the positive false alarm rate is computed.

- **False rejection:** an intrusion has occurred and the IDS has not generated an alarm. Based on this parameter, the negative false alarm rate is computed.

- **Accuracy:** the number of correct alarms divided by the number of correct alarms plus the number of false alarms. Which indicates the system generalization capability.

- **Completeness**: the number of correct alarms divided by the number of correct alarms plus the number of false rejections [22].

4. KDD DATA SET

Data was acquired from the 1999 DARPA intrusion detection evaluation program [21]. It was gathered by Lincoln Labs that set up an environment to acquire raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks. The data set was organized as records, each record represents one TCP connection; a connection is a sequence of TCP packets. Each connection has 41 features labeled as either normal, or as an attack, with exactly one specific attack type. Based on this data set, attacks can be categorized into four main categories:

- **Denial of Service (DoS) Attacks:**
  A denial of service attack is a class of attacks in which an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

- **User to Root (U2R) Attacks:**
  User to root attacks exploits are a class of attacks in which an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system.

- **Remote to User (R2U) Attacks:**
  A remote to user attacks is a class of attacks in which an attacker sends packets to a machine over a network but who does not have an account on that machine; exploit some vulnerability to gain local access as a legitimate user of that machine.

- **Probing Attacks:**
  Probing is a class of attacks in which an attacker scan a network of computers to gather information or find known vulnerability. An attacker with a map of machines and service that are available on a network can use this information to look for exploits.
  A complete listing of the set of features defined for the connection records is given in the Table 1 [21].

5. THE PROPOSED NNIDS

The proposed neural Network Intrusion Detection System (NNIDS) consists of the following two phases:

- **Pre-processing phase:** randomly select two separated training and testing data sets from the full DARPA data set, convert the symbolic features into numerical ones, delete high correlated features, and normalize the remaining data set.
- **Neural network creation phase:** determine the number of layers, and the number of nodes per hidden layers, and create the neural network.
  - **Training phase:** in this phase NNIDS is trained using the training data set.

- **Testing phase:** measures the performance of the system to the testing data set.

5.1 The preprocessing phase

Fig. 1 shows a detailed block diagram of the NNIDS preprocessing phase.

- **Data Sets Extraction:** Two subsets of the full data set were randomly selected and used as training and testing data sets.

- **Symbolic to Numerical Conversion:** Some features have symbolic form (e.g. protocol type). These features were converted into numerical ones by assigning a unique number for each feature. The resulting map is used to do the same for the testing data set.

- **High Correlation Deletion:** Since the high correlation features introduce no significant additional information during the neural network-training phase, therefore, out of each two highly correlated features one of them was deleted. Eq. 1 was used to measure the correlation between two vectors X and Y [18].

$$r=\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})/\sqrt{[\sum_{i=1}^{n}(x_i-\bar{x})^2/n][\sum_{i=1}^{n}(y_i-\bar{y})^2/n]} \qquad (1)$$

Where:

$\bar{x}, \bar{y}$: The mean value of X, Y respectively.

n : the number of paired X and Y.

Two features are considered to be highly correlated if r >= 0.8.

- **Normalization:** It is often useful to scale the inputs to fall within a specific range, in the proposed system. Equation 2 was used to normalize the training and testing data sets, [4].

$$X_n = 2 * (X - X_{min}) / (X_{max} - X_{min}) - 1 \qquad (2)$$

Where: X_{min}, X_{max}: are the Minimum and maximum value of the original inputs, respectively.

X_n: is the normalized output.

The resulting output will fall in the range [-1, +1]. The training data set will have M features, N_1 intrusion type, and Q_1 records. The testing data set will have M features, N_2 intrusion type, and Q_2 records.

5.2 Neural network creation phase

5.2.1 Neural Network Architecture

An artificial neural network consists of a collection of processing elements that are highly interconnected to transform a set of inputs into a set of desired outputs. The result of the transformation is determined by the characteristics of of the elements and weights associated with the interconnections among them. By modifying/adapting the connection between the nodes, the network is able to adapt the desired outputs. In the proposed system, the input layer is restricted to have nodes equal to the number of extracted features, and the output layer is also restricted to have nodes equal to the intrusion types plus one that represents the main output, intrusion/normal decision. Multiple neural network architectures with multilayer feed-forward back-propagation networks were tested. The purpose of having multiple networks is to find a suitable architecture that can detect at a faster speed with low error rate, minimizing false negative and false positive alarm rates. The best architecture obtained consists of three layers with 32, 32, and 21 nodes in the input, hidden, and output layers respectively. The activation functions used in the hidden layer and output layer nodes were tangent sigmoid transfer function and log sigmoid transfer function, respectively.

5.2.2 The Training Phase

The training data set consisted of 35,000 records with 32 features, and 20 intrusion types. The designed networks were trained using gradient decent with momentum and adaptive learning rate with back propagation learning function until achieving the minimum mean square error.

5.2.3 The Testing phase and threshold Effect

The testing data set was consisted of 20,000 records. The actual classification for each record was known before hand. Since the output nodes transfer functions are log sigmoid, the output values will be in the range [0-1], so it was necessary to declare the decision level (threshold) that will classify the outputs as zeros or ones.
By applying this testing data set as inputs to the trained neural network, and changing the threshold value at the output nodes between [0 - 0.95], false positive and false negative were changed according to each threshold value. Fig. 2 shows the false positive, false negative rate, and threshold relationship.

It could be noticed that if the threshold value is too low, the higher output values corresponding to the normal connections will considered as intrusion, so the false positive will be high. Since false positive is high, the accuracy will be low, and vice versa. On the other hand, If the threshold value is too high, the lower output values corresponding to the intrusion connections will be considered as normal connections and thus the false rejection will be high. Since false rejection is high, the completeness will be low. Thus, the threshold value is very critical point, and it can directly affect the system performance. Table 2 presents the neural network set of performance parameters along with different threshold values.

6. CONCLUSION

This paper introduced an anomaly network based intrusion detection system using neural network classification approach. A neural network conducts an analysis of the information and provides a probability estimate that the data matches the characteristics that it has been trained to recognize. The probability of match determined by neural network relies totally on the experience the system gains in the training phase. The proposed system was implemented and relatively high generalization capabilities were obtained. However, as an extension to the proposed system a complete and on-line system is currently under investigation. This extended system will be based on the proposed system architecture but with the ability of directly receiving inputs from a network data stream, effectively extracting useful features from the IP datagram, passing it directly to the NNIDS to detect intrusions in real time. Both network and host-based IDS solutions have unique strengths and benefits that complement each other. Combining these two technologies will greatly improves network resistance to attacks and misuse.

REFERENCES

[1] G. Vigna and R.A. Kemmerer, "NetSTAT: A Network-based Intrusion Detection Approach", *Proceeding of 14th Annual Computer Security Application Conference, 1998.*

[2] V. Paxson, "A System for Detecting Network Intruders in Real-Time", Proceeding of the 17th USENIX security Symposium, San Antonio, Texas, 1998.

[3] C. Manikopoulos and S. Papavassiliou, "Network Intrusion and Fault Detection: A Statistical Anomaly Approach", IEEE Communications Magazine, 2002.

[4] Khaled Labib and V. Rao Vemuri, "NSOM: A Real-time Network-Based Intrusion Detection System Using Self-Organizing Maps", University of California, Davis, 2002.

[5] K. Ilgun, R. A. Kemmerer and P.A. Porras, "State Transition Analysis Intrusion Detection Approach", IEEE transaction on software engineering, 1995.

[6] J. Lee, S. Moskovics, and L. Silacci, "A Survey of Intrusion Detection Analysis Methods", University of California, san Diego, 1999.

[7] Mikhail Gordeev," Intrusion Detection: Technique and Approaches", http://olympus.cs.ucdavis.edu/papers

[8] Ulf Lindgvist, Phillip A. Porras, "Detecting Computer and Network Misuse through the Production-Base Expert System Toolset (P-BEST)", IEEE Symposium on Security and Privacy, California, May 1999.

[9] Parbhker Mateti, "Intrusion Detection", Wright State University, http://www.cs.wright.edu/~pmateti/InternetSecurity.

[10] W. Lee and S. J. Stolfo, "Data Mining Approaches for Intrusion Detection", Proceeding of the 17th USENIX security Symposium, San Antonio, Texas, 1998.

[11] W. Lee, S. J. Stolfo, and K. W. Mok."A data mining framework for building intrusion detection models", Proceedings of the 1999 IEEE Symposium on Security and Privacy, May 1999.

[12] S Forrest, S. Hofmeyr, and A. Somayaji," Intrusion detection using sequences of system calls", Journal of Computer Security, 1998.

[13] D. Anderson, T. Frivod, and A. Valdes, "Next-generation Intrusion Detection Expert System (NIDES)", Technical Report, Computer Science Laboratory, SRI International, Menlo Park, 1995.

[14] Zheng Zhang, Jun Li, C.N. Manikopoulos, Jay Jorgenson, and Jose Ucles, "HIDE: a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing And Neural Network Classification", Proceedings of the IEEE workshop on Information Assurance and security, 2001 .

[15] S. Kumar and Eugene H. Spafford, "A Pattern Matching Model For Misuse Intrusion Detection", Proceedings of the National Computer Security Conference 1994.

[16] Paul Helman and Gunar Liepins. "Statistical Foundations of Audit Trail Analysis for the Detection of Computer Misuse", IEEE Transactions on Software Engineering, 1993.

[17] Ismail Taha, "AMIDS: An Adaptive Multilevel Intrusion Detection System", First International Conference on Intelligent Computing and Information Systems (ICICIS), Jun 2002, pp. 491-496.

[18] Sherif M. Badr, "Security Architecture for Internet protocols", Ph.D. dissertation, Military Technical Collage, 2001.

[19] J. Cannady, "Artificial Neural Networks for Misuse Detection", Proceedings of the 1998 National Information Systems Security Conference ,1998.

[20] Jake Ryan and Meng Jang Lin, "Intrusion Detection With Neural Network,", Cambidge, MIT press, 1998.

[21] http://kdd.ics.uci.edu/database/kddkup99.

[22] Task Group on Information Assurance, "Intrusion Detection: Generics and State of the Art", Information System Technology Panel, 2002.

Table (1): Network Extracted Features and Their Corresponding Descriptions and Types

Feature Name	*Description*	*Type*
Duration	Length (number of seconds) of the connection	Continuous
Protocol_type	Type of the protocol, e.g. tcp, udp, etc.	Discrete
Service	Network service on the destination, e.g., http, telnet, etc.	Discrete
Src_bytes	Number of data bytes from source to destination	Continuous
dst_bytes	Number of data bytes from destination to source	Continuous
Flag	Normal or error status of the connection	Discrete
Land	1 if connection is from/to the same host/port; 0 otherwise	Discrete
Wrong_fragment	Number of ``wrong" fragments	Continuous
Urgent	Number of urgent packets	Continuous
Hot	Number of ``hot" indicators	Continuous
Num_failed_logins	Number of failed login attempts	Continuous
Logged_in	1 if successfully logged in; 0 otherwise	Discrete
Num_compromised	Number of ``compromised" conditions	Continuous
Root_shell	1 if root shell is obtained; 0 otherwise	Discrete
Su_attempted	1 if ``su root" command attempted; 0 otherwise	Discrete
Num_root	Number of ``root" accesses	Continuous
Num_file_creations	Number of file creation operations	Continuous
Num_shells	Number of shell prompts	Continuous
Num_access_files	Number of operations on access control files	Continuous
Num_outbound_cmds	Number of outbound commands in an ftp session	Continuous

Feature Name	*Description*	*Type*
Is_hot_login	1 if the login belongs to the ``hot" list; 0 otherwise	Discrete
Is_guest_login	1 if the login is a ``guest"login; 0 otherwise	Discrete
Count	Number of connections to the same host as the current connection in the past two seconds	Continuous
Serror_rate	% of connections that have ``SYN" errors	Continuous
Rerror_rate	% of connections that have ``REJ" errors	Continuous
Same_srv_rate	% of connections to the same service	Continuous
Diff_srv_rate	% of connections to different services	Continuous
Srv_count	Number of connections to the same service as the current connection in the past two seconds	Continuous
Srv_serror_rate	% of connections that have ``SYN" errors	Continuous
Srv_rerror_rate	% of connections that have ``REJ" errors	Continuous
Srv_diff_host_rate	% of connections to different hosts	Continuous

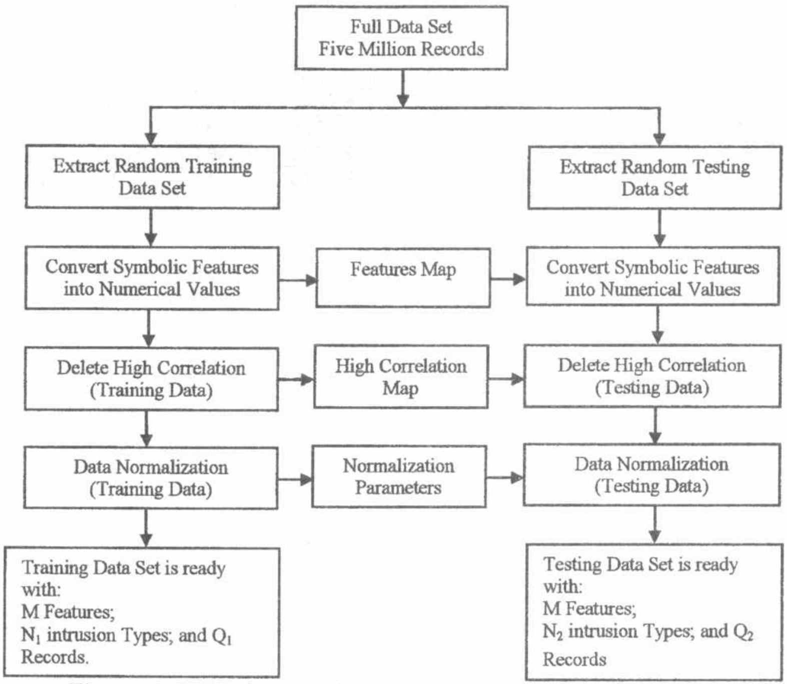
Full Data Set
Five Million Records
Extract Random Training Data Set
Extract Random Testing Data Set
Convert Symbolic Features into Numerical Values
Features Map
Convert Symbolic Features into Numerical Values
Delete High Correlation (Training Data)
High Correlation Map
Delete High Correlation (Testing Data)
Data Normalization (Training Data)
Normalization Parameters
Data Normalization (Testing Data)
Training Data Set is ready with: M Features; N1 intrusion Types; and Q1 Records.
Testing Data Set is ready with: M Features; N2 intrusion Types; and Q2 Records

Figure 1: NNIDS preprocessing phase major components

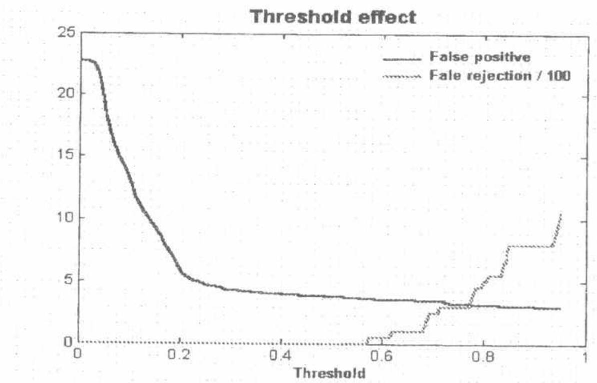

Figure 2: Output nodes threshold effect on the output of the neural network.

Table 2: Obtained results with different threshold values

Threshold Parameter	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Correct Alarms %	100%	99.994	99.987	99.981	99.961	99.955	99.877	99.832
Correct Rejections %	72.994	81.129	86.278	88.646	91.38	92.504	93.43	94.466
Detection Accuracy %	92.66	94.755	96.133	96.776	97.533	97.848	98.107	98.4
Completeness %	100%	99.994	99.987	99.981	99.961	99.955	99.877	99.832
Intrusion Type Accuracy %	99.004	99.547	99.515	99.515	99.547	99.114	98.718	98.692