2010 Second International conference on Computing, Communication and Networking Technologies

Intrusion Detection using Artificial Neural Network

Poojitha.G
I.T. 2/4B.Tech ,
Poojitha021@gmail.com

Naveen kumar .K
I.T ¾, B.Tech
naveenk019@gmail.com

JayaramiReddy.P
Associate. Professor,E.C.E ,
jayareddy_net@yahoo.co.in

Sri Venkateswra Institute of Science and Technology,tadigotla(vi&post),kadapa-516003, A.P.,India

Abstract

Intrusion Detection is the task of detecting, preventing and possibly reacting to the attack and intrusions in a network based computer systems. In the literature several machine-learning paradigms have been proposed for developing an Intrusion Detection System. This paper proposes an Artificial Neural Network approach for Intrusion Detection. A Feed Forward Neural Network trained by Back Propagation algorithm is developed to classify the intrusions using a profile data set (ten percent of the KDD Cup 99 Data) with the information related to the computer network during Normal behavior and during Intrusive (Abnormal) behavior. Test result shows that the proposed approach works well in detecting different attacks accurately with less false positive and negative rate and it is comparable to those reported in the literature.

Keywords: Intrusion Detection, Feed Forward Neural Network, Back Propagation Algorithm, KDD Cup'99 data.

Introduction

Confidentiality, integrity and availability of the system resources are the major concerns in the development and exploitation of network based computer systems. Enlargements of computer infrastructure have raised the vulnerability of these systems to security threats, attacks and intrusions. Some specific examples of intrusions that concern system administrators include Attempted break-in, Masquerading or successful break-in,

978-1-4244-6589-7/10/$26.00 ©2010 IEEE

Penetration by legitimate user, Leakage by legitimate user, Inference by legitimate user, Trojan Horse, Virus and Denial-of-Service. Generally these intrusions would cause loss/damage to our system resources in terms of unauthorized modifications of system files, user files or information and any other system information in network components.

Hence a system is needed that detects any unauthorized modification forced by an attacker and able to run continually with minimal human supervision .According to the detection principles there are two types of intrusion detection system: Misuse and Anomaly detection. In Misuse detection, attack patterns or the behavior of the intruder is modeled (attack signature is modeled). Here the system will signal the intrusion once a match is detected. In Anomaly detection system, the normal behavior of the system is modeled and the system will raise an alarm once the behavior of the network does not match with its normal behavior. According to the source of data, there are two types of intrusion detection: Network-based IDS (NIDS) and Host-based IDS (HIDS). A network based IDS captures all network traffic and analyzes the content of individual packets for malicious traffic where as a host-based IDS identifies intrusions by analyzing system calls, application logs, file system modifications (binaries, password files, capability/acl databases) and other host activities and state. In the literature several machine-learning paradigms have been proposed for developing an Intrusion Detection System. Statistical Techniques like Hidden Markov Model, Multivariate Adaptive Regression Splines, Bayesian Classifier and Classification and Regression Trees (CART) have been applied to Intrusion detection. These statistical approaches usually results in an inflexible detection system that is unable to detect an attack if the

Authorized licensed use limited to: Concordia University Library. Downloaded on August 03,2022 at 19:59:14 UTC from IEEE Xplore. Restrictions apply.

sequence of events slightly different from the predefined profile. Most current approaches to the process of detecting intrusions utilize some form of rule-based analysis as proposed by Denning . Expert systems are the most common form of rule-based intrusion detection approaches. They permit the incorporation of an extensive amount of human experience into a computer application that then utilizes that knowledge to identify activities that match the defined characteristics of misuse and attack. The constantly changing nature of network attacks requires a flexible defensive system that is capable of analyzing the enormous amount of network traffic in a manner which is less structured than rule-based systems. In fuzzy logic approach has been combined with data mining techniques for mining association rules which can be applied for detecting intrusions. Recently, Artificial Neural Networks have been successfully applied for developing the IDS. ANN has the advantage of easier representation of nonlinear relationship between input and output and its inherent computational speed. Even if the data were incomplete or distorted, a neural network would be capable of analyzing the data from a network. A Multilayer Perceptron (MLP) was used in for misuse detection with a single hidden layer. A Similar approach was applied in but generic keywords were selected to detect the attack preparations and actions after the break-in. Self-Organizing Map was applied to perform the clustering of network traffic and to detect attacks in [8, 9]. In [8], SOM was used to map the network connections onto 2-dimensional surfaces, which were displayed to the network administrator. The intrusions were easily detected in this view. However, the approach needs a visual interpretation by the network administrator. The SOM is trained by using the normal network traffic in [9]. The trained SOM reflects the distribution of the normal network connections. If the minimum distance between a network connection and the neurons of the trained SOM is more than a pre-set threshold, this connection is classified as an intrusion. A hybrid model of the SOM and the MLP was proposed in [10]. In that work, the self-organizing map was combined with the feed-forward neural network. This model was designed to detect the dispersing and possibly collaborative attacks. In [11], the self-organizing map was combined with the Resilient Propagation Neural Network (RPROP) for visualizing and classifying intrusion and normal patterns. In this paper, a Feed Forward Neural Network trained by Back Propagation algorithm is used to classify the intrusions using a profile data set (ten percent of the KDD Cup 99 Data) with the information related to the computer network during Normal behavior and during Intrusive (Abnormal) behavior. Artificial Neural Networks [12] provide the potential to identify and classify network activity based on limited, incomplete and non-linear data sources. With their ability to generalize from learned data they seem to be an appropriate approach to Intrusion Detection.

2. Proposed model for Intrusion Detection

The proposed methodology for Intrusion Detection in Computer Networks is based on using Artificial Neural Network (ANN) for detecting the Normal and Abnormal conditions of the given parameters, which leads to various attacks. The neural network approach for this purpose has two phases; training and testing. During the training phase, neural network is trained to capture the underlying relationship between the chosen inputs and outputs. After training, the networks are tested with a test data set, which was not used for training. Once the networks are trained and tested, they are ready for detecting the intrusions at different operating conditions. The following issues are to be addressed while developing an ANN for Intrusion Detection [13]:

1. Data Collection
2. Data preprocessing and representation
3. Data Normalization
4. Selection of Network Structure
5. Network Training and Testing

Figure 1 shows the schematic representation of the issues to be addressed while developing an ANN model for Intrusion Detection

2.1 Data Collection

There are two ways to build IDS, one is to create our own simulation network, and collect relevant data and the other one is by using previously collected datasets. Issues like privacy, security, and completeness greatly restrict people from generating data. The beauty of using previously collected datasets is that the results can be compared with others in the literature. Not many data sets have being collected that could built IDS systems. Some of the popularly used IDS datasets are DARPA 1998 data set, DARPA 1999

Authorized licensed use limited to: Concordia University Library. Downloaded on August 03,2022 at 19:59:14 UTC from IEEE Xplore. Restrictions apply.

data set and KDD Cup 1999 data set which are available in the MIT Lincoln Labs [14].

2.2 Data Preprocessing

Before training the neural network, the dataset should be preprocessed to remove the redundancy present in the data and the non-numerical attributes should be represented in numerical form suitably.

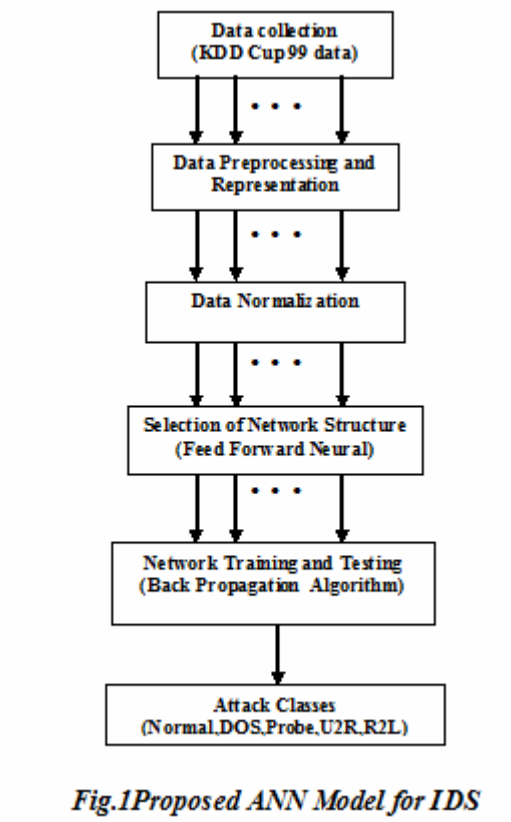

Fig.1Proposed ANN Model for IDS

2.3 Data Normalization

During training of the neural network, higher valued input variables may tend to suppress the influence of smaller ones. Also, if the raw data is directly applied to the network, there is a risk of the simulated neurons reaching the saturated conditions. If the neurons get saturated, then the changes in the input value will produce a very small change or no change in the output value. This affects the network training to a great extent. To minimize the effects of magnitudes among inputs as well as to prevent saturation of the neuron activation function, the input data is normalized before being presented to the neural network. One way to normalize the data x is by using the expression:

$$x_n = \frac{(x - x_{min}) \times range}{(x_{max} - x_{min})} + starting\ \ value \qquad (1)$$

where , xn is the normalized value and xmin and xmax are the minimum and maximum values of the data.

2.4 Selection of Network Structure

To make a Neural Network to perform some specific task, one must choose number of input neurons, output neurons, hidden neurons and how the neurons are connected to one another. For the best network performance, an optimal number of hidden-units must be properly determined using the trial and error procedure. The hidden layer neurons have tangent hyperbolic function as the activation function and the output have linear activation function.

2.5 Network Training and Testing

Once the appropriate structures of the network are selected, the ANN model is trained to capture the underlying relationship between the input and output using the training data. In this work, Back propagation algorithm is used to train the network, which propagates the error from the output layer to the hidden layer to update the weight matrix. After training, the networks are tested with the test data set to assess the generalization capability of the developed network.

3. Review of Artificial Neural Network

Artificial Neural Networks [13] can be viewed as parallel and distributed processing systems which consists of a huge number of simple and massively connected processors. The MLP architecture is the most popular paradigm of artificial neural networks in use today. Figure 1 shows a standard multilayer feed forward network with three layers. The neural network architecture in this class shares a common feature that all neurons in a layer are connected to all neurons in adjacent layers through unidirectional branches. That is, the branches and links can only broadcast information in one direction, that is, the “forward direction”. The branches have associated weights that can be adjusted according to a defined learning rule.

Authorized licensed use limited to: Concordia University Library. Downloaded on August 03,2022 at 19:59:14 UTC from IEEE Xplore. Restrictions apply.

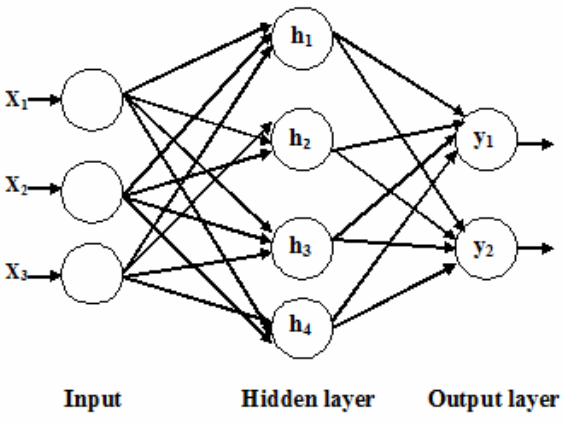

Fig.2 Architecture of feed forward neural network

Feed forward neural network training is usually carried out using the called back propagation algorithm. Training the network with back propagation algorithm results in a non-linear mapping between the input and output variables. Thus, given the input/output pairs, the network can have its weights adjusted by the back propagation algorithm to capture the non-linear relationship. After training, the networks with fixed weights can provide the output for the given input. The standard back propagation algorithm for training the network is based on the minimization of an energy function representing the instantaneous error. In other words, we desire to minimize a function defined as

$$E(m) = \frac{1}{2}\sum_{q=1}^{q}\left[d_q - y_q\right]^2 \qquad (2)$$

Where dq represents the desired network output for the qth input pattern and yq is the actual output of the neural network. Each weight is changed according to the rule:

$$\Delta w_{ij} = -k\frac{dE}{dw_{ij}} \qquad (3)$$

where, k is a constant of proportionality, E is the error function and wij represents the weights of the connection between neuron j and neuron i. The weight adjustment process is repeated until the difference between the node output and actual output are within some acceptable tolerance.

4. Simulation Result

This section presents the details of the simulation study carried out on KDD Cup 1999 Dataset [15] using the proposed method. This data set was collected by simulating a typical U.S Air force local area network (LAN), operated like a real environment and being blasted with multiple attacks. Each KDD records contains 41 input features which is given in table 1 and one output that is labeled as either normal or as an attack, with exactly one specific attack type (DOS, Probe, U2R, R2L).

The 41 input features are divided into four feature subsets. They are Basic or Intrinsic features, Content features, Time-based features and Host-based features. Basic features are features to every network connection like duration of connection, service requested, bytes transferred between source and destination machine, etc. Content features are collected by using domain knowledge of U2R and R2L attacks since these attack categories did not contain any frequently occurring patterns. E.g. logged in flag, number of failed logins, number of root commands, number of compromised conditions, hot indicators, etc. Time-based features are collected by observing various connections in "two-second" time window with respect to current connection. E.g. SYN error rates, Rejection rates, umber of different services requested etc. Host based features are collected based on the past 100 connections similar to the one under consideration. The original data contain 744MB data with 4,940,000 records. A ten percent subset of this data contain 75MB with 4,94,021 records which approximately contain 20% represent normal patterns and the rest 80% of patterns are attacks belonging to four categories (DOS, Probe, U2R and R2L). Among them only 12,723 records randomly for developing the Neural Network. The details of the records selected for training and testing the Neural Network is given in the table II Among the 41 input features, 32 features are continuous variables and 9 features are discrete variables. Suitable integer numbers are assigned to these discrete variables. For example, for the discrete variable protocol_type which describes the type of the

Total number of samples:12,723					
Data Distribution	**Normal**	**DOS**	**Probe**	**U2R**	**R2L**
Training: 6,363	2500	1500	1500	20	843
Testing: 6,360	2500	1500	1500	19	841

Authorized licensed use limited to: Concordia University Library. Downloaded on August 03,2022 at 19:59:14 UTC from IEEE Xplore. Restrictions apply.

protocol we have assigned 1 for tcp, 2 for udp, 3 for http and so on. Accordingly suitable integer numbers are assigned to other discrete variables.

Feature Number	Feature Name	Feature Number	Feature Name
F1	Duration	F22	Is_guest_ login
F2	Protocol-type	F23	count
F3	service	F24	srv_count
F4	flag	F25	serror_rate
F5	src_bytes	F26	srv_serror_rate
F6	dst_bytes	F27	rerror_rate
F7	land	F28	srv_rerror_rate
F8	wrong_fra gment	F29	same_srv_rate
F9	urgent	F30	diff_srv_rate
F10	host	F31	srv_diff_host_rat e
F11	num_failed_lo gins	F32	dst_host_count
F12	logged_in	F33	dst_host_srv_cou nt
F13	num_compro mised	F34	dst_host_same_sr v_rate
F14	root_shell	F35	dst_host_diff_srv _rate
F15	su_attempted	F36	dst_host_same_sr c_port_rate
F16	num_root	F37	dst_host_srv_diff _host_rate
F17	num_file_crea tions	F38	dst_host_serror_r ate
F18	num_shells	F39	dst_host_srv_serr or_rate
F19	num_access_f iles	F40	dst_host_rerror_r ate
F20	num_outboun d_cmds	F41	dst_host_srv_rerr or_rate
F21	is_host_login		

The output attack label is represented as [0 0 0 0] for Normal, [0 0 0 1] for DOS, [0 0 1 0] for Probe, [0 1 0 0] for R2L and [1 0 0 0] for U2R.

Table II Distribution of Data

There are about forty one neurons in the input layer that corresponds to the number of input features and four neurons in the output layer in which all neurons set to zero corresponds to Normal and one in each neuron corresponds to any one of the four attacks (DOS, Probe, R2L and U2R).

Table I:Names of input features in KDD cup '99 data

The number of output neurons is kept constant while the number of input neurons are varied depending on the features selected by Mutual Information. The number of hidden-units is directly related to the capabilities of the network. The neural network model is developed using MATLAB 6.5 Neural Network Toolbox in Pentium 4 with 2.40 GHz processor with 256 MB of RAM. The network is trained with least mean square algorithm until it reaches the mean square error of 0.001. Trial and error procedure was followed to identify the optimal number of hidden nodes. The mean square error achieved during training is 9.9975e-004. With ten hidden nodes, the network took 257.7030 seconds to reach the error goal. The performance of the network during training is shown in fig 3.

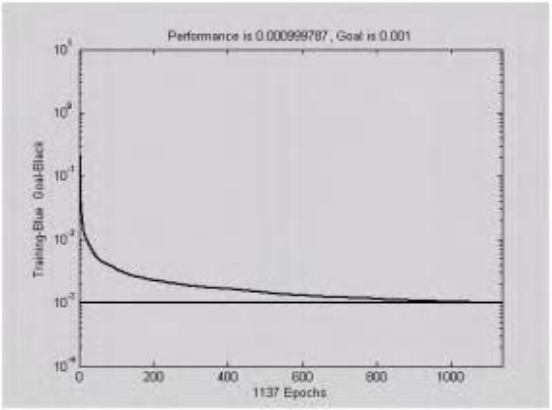

Fig. 3 Training performance of the Neural Network

Authorized licensed use limited to: Concordia University Library. Downloaded on August 03,2022 at 19:59:14 UTC from IEEE Xplore. Restrictions apply.

After training, the generalization performance of the network is evaluated with the 6,360 test data. The proposed Neural Network classifies 6,038 data correctly which shows an overall detection rate of 94.93%. During testing the Mean Square Error achieved by the network is 0.0097. The performance of the network during testing is presented in Table III.

Approaches	Accuracy	False Positive rate	False Negative rate	Ephocs
ANN (proposed approach)	94.93 %	0.002	0.7	1067
ANN	87.07 %	6.66	6.27	412

Testing performance	Normal	DOS	Probe	U2R	R2L
No.of Correctly Classified Class	2494		1500	570	7
Percentage of classifier accuracy	99.76%	100%	1500	67.77%	36.84%

The performance of the proposed Artificial Neural Netework Model has been compared with the other Approaches and it is presented in the table IV and table V. Table IV shows that the proposed ANN approach has a high detection rate with less false positive and negative rate when compared with the result reported in [16].

Table III Performance of the proposed ANN Model

Table IV Performance comparison with other approach

5. Conclusion and Future Work

In this paper, a simple feed forward neural networks trained by the back propagation algorithm was developed to classify the intrusions into one of the attacks (Normal, DOS, Probe, R2L, and U2R). The performance of the network was tested using ten percent of the KDD cup 1999 dataset which is available in the UCI KDD Archive and is compared with other approaches. Test result shows that the proposed approach works well in detecting Normal, DOS and Probe attacks but weak in detecting R2L and U2R cases. The weakness of neural network based approaches is that if the dimension of the input data is very large then it is difficult for it to interpret the relationship between inputs and outputs. The input data used in this work contains 41 input features which may possible to have redundant data and false correlations which hinder the process of detecting intrusions. To make the Neural Network applicable to very large data set, some dimensionality reduction is mandatory. Hence as an enhancement to this proposed work, future research will be directed towards the dimensionality reduction techniques which remove the unwanted input futures and select only the optimal feature for training the neural network thereby increases the detection rate especially for the R2L and U2R type of attacks.

References

[1] A.Zhong and C.F. Jia, "Study on the applications of hidden markov models to computer intrusion detection" in Proceedings of the Fifth World Congress on Intelligent Controll and Automation WCICA, vol. 5, pp. 4352-4356. IEEE, June 2004.

[2] M.Analoui, A.Mizaei, and P.Kabiri, "Intrusion detection using multivariate nalysis of variance algorithms," in Third International Conference on Systems, Signals & Devices SSD05, vol. 3, Sousse, Tunisia. Mar. 2005, IEEE.

[3] Srilatha Chebrolu, Ajith Abraham and Johnson P. Thomas. Feature deduction and ensemble design of intrusion detection system. Computers & Security, Volume 24, Issue 4, June 2005, Pages 295-307.

[4] Denning DE. An Intrusion-Detection Model. IEEE Transactions on Software Engineering, Feb 1987, Vol. SE-13, No. 2, 222-232.

[5] Luo J., Bridges S. M., (2000) "Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection," International Journal of Intelligent Systems, John Wiley & Sons, Vol. 15, No.8. pp. 687-704

Authorized licensed use limited to: Concordia University Library. Downloaded on August 03,2022 at 19:59:14 UTC from IEEE Xplore. Restrictions apply.

[6] Cannady, J., 1998, "Artificial Neural Networks for Misuse Detection," Proceedings, National Information Systems Security Conference(NISSC'98), October, Arlington,pp.443-456.

[7] R.P. Lipmann and R.K Cunningaham. "Improving Intrusion Detection Performance using keyword selection and neural networks", Computer Networks (Amsterdam, Netherlands:1999), 34(4):597-603, 1999.

[8] 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.

[9] 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.

[10] James Cannady and Jim Mahaffey, "The application of Artificial Intelligence to Misuse Detection", in proceedings of the first Recent Advances in Intrusion Detection(RAID) Conference, 1998.

[11] C.Jirapummin, N.Wattanapongsakorn and P.Kanthamanon, "Hybrid Neural Networks for Intrusion Detection System", Proceedings of the 2002 Internation Technical Conference on Circuits / Systems, Computers and Communications (ITC-CSCC 2002), pp.928-931, Thailand, 2002.

[12] J.M. Bonoficio, "Neural Networks Applied in Intrusion Detection Systems", Neural Networks Proceedings, IEEE World Congress on Computational Intelligence, vol. 1, pages:205-210, 1998.

[13] P.GaneshKumar, D.Devaraj, V.Vasudevan, "Artificial Neural Network for Misuse Detection in Computer Network", Proceddings of the International Conference on resource utilization

and Intelligent Systems (INCRUIS-2006), Kongu Engineering Colllege, Perunduari, Erode, 4-6 January'2006, pp.889-893.

[14] DARPA Intrusion Detection Evaluation – MIT Lincoln Laboratory – (http://www.ll.mit.edu/IST/ideval)

[15]KDDcupdataset,http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.htm

[16] Mukkamala S., Janoski G., Sung A.H. (2002) "Intrusion Detection Using Neural Network and Support Vector Machines" Proceedings of IEEE International Joint Conference on Neural Networks, pp. 1702-1707.

Authorized licensed use limited to: Concordia University Library. Downloaded on August 03,2022 at 19:59:14 UTC from IEEE Xplore. Restrictions apply.