Intrusion Detection Using Error Back - Propagation Neural Network

[1]Ashutosh Kumar Agarwal, [2]Sandhya Tarar
[1]Post Graduate Student, [2]Faculty Associate
[1]School of Information and Communication Technology,
[1] Gautam Buddha University, Greater Noida, India

Abstract : The growth in Internet, makes the network vulnerable. At present, companies spend a large sum of money to protect their confidential data from the various attacks they face. This paper suggests an Intrusion Detection System based on the EBPNN model combined together with the Genetic Algorithm (GA). The system which is proposed in the paper was simulated using a reference intrusion data set for NSD-KDD to verify its feasibility and effectiveness for this data.

IndexTerms **- Genetic Algorithm (GA), Error Back-Propagation Neural Network (EBPNN), Intrusion Detection System (IDS), Recurrent Neural Network (RNN).**

I. INTRODUCTION

The intrusion detection system [1] monitors abnormal situations and activates the alarm clock. There are two types of detection systems, namely host-based IDS and network-based identifiers [2], [3]. The basic methods used to detect penetration are detection of anomalies, misuse or signature-based detection [4], [5], [6]. Anomalies are detected in traffic, while abuse attempts to match the data to a recognized attack pattern. One of the major flaws in detecting misuse [5] is the discovery of a new form of attack. Therefore, most investigations focused on detection techniques for anomalies [3]. The anomaly-based technique contains statistical data, neural networks, automated learning [7], data extraction [8], and immune system approach [6], [9]. Neural network approaches have the ability to detect all the recognized and unrecognized attacks in a network. It can be differentiated in a controlled and uncontrolled training algorithm. Multiprotect perceptron (MLP) is a model of the supervised algorithm, while the Self-Organizing Map is an unmanaged algorithm. Several papers have used neural network approaches [10]. According to the survey conducted in [11], LBW has a good detection rate compared to other neural network technologies and therefore can be used to classify specific attacks so that you can take preventive measures. BPN uses a learning approach under the supervision of the training. According to the experimental results of previous research work, it was noted that BPN achieved good results. The purpose of this paper is to construct a predictive model using a sequential propagation algorithm to identify and classify the type of attack. The first Back Propogation Network is trained using the NSL-KDD data set with 23 types of attacks. After training, you will make predictions about test data to classify events in your types of attacks.

II. RELATED WORKS

Neural Network uses an adaptive learning technology to define abnormal behavior, neurons in many simple processing units, using weighted communication. The example interaction can be used to adapt or configure the self-learning weight function of the neural network, so that the network understands and resolves the particular problem and achieves the best performance possible. Neural networks can be classified according to different angles, for example, according to the topology of network points and the type of network and network feedback; model, network learning guide, without learning guide and learning networks reinforcement; according to mathematical model and mathematical model and network cognitive model; The network is continuous and discrete, along with models and network type identification. The common neural network model is the cognitive network, the linear neural network, the BP network and the primary beam. Network nction, Hopfield network, self-organization network, ect. Among them, the BP network model is the most advanced multi-layer neural network learning algorithm, and is primarily used for approach functionality, pattern recognition, classification and data compression. Currently, in the practical applications of the artificial neural network, most of the majority of the neural network model relies on the BP algorithm or its another form. Before people perfected the design of the reverse propagation network, the sensors and the adaptive linear network were only suitable for a single training network model. The neural network of the neural network is the front neural network. In the calculation of the output value, the input values of the input layer module can be propagated step by step, through the hidden layer to the output layer to obtain the end at the end. Before the first layer of the network unit and the second layer of all the cells are combined together, the second layer and the cell layer in their association, there is no connection between the units in the same layer. Neural networks in the excitation function can use the linear hard threshold function or the nonlinear function unit, etc. In the training process, the algorithm weights are adjusted with the learning base of the delta teacher. After determining the BP network structure, through a set of training samples on the input and output network, no threshold and network weight learning and debugging, so that the network reaches the specified I / O mapping relationship. The learning process for the BP network is divided into two phases:

1. Positive communication process: Enter the known learning samples, the composition of the network structure and the last frequency of weights and thresholds, from the first calculation layer to the back of each neural network output.

2. Reverse diffusion process: weights and thresholds, modified from the last layer of the forward weight calculation and the threshold effect on the total error, adjusting the weights and thresholds accordingly.

The last two processes, alternatively, until convergence repeatedly. Due to an error, it can be republished step-by-step to correct the weight and threshold between the layer and the layer, so the algorithm returns the algorithm. Learn to correct your weight in opposite direction to get a learning error correction rule, gradient error.

The artificial neural network consists of a group of highly correlated processing elements and converts a set of input into the set of required outputs. The result of change is determined by the characteristics of the weight associated with the elements and their interconnectedness. By modifying communication between the nodes, the network may be favorable for the required output. BPN's biggest strength lies in the non-linear solution of unspecified problems [6]. Background spread grid has at least one input layer, output layer and at least one hidden layer (see Figure 1) [10]. Even though there is no theoretical limit for the number of hidden layers, only one or two layers are used usually.

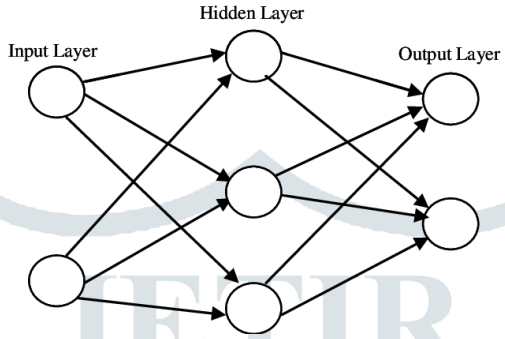

Figure 1: Structure of EBPNN

III. INTRUSION DETECTION SYSTEM ARCHITECTURE

The proposed IDS structure can be divided into four sub-processes. The functions of these subprocesses are explained below. Prefabricated data acquisition tool Neural network categorization Figure 2 Suggested IDS architecture entries

1. Information complex: based on detection of penetration DARPA program, collection of data sets NSL KDD, available to the public through the MIT Lincoln Lab, for the first time in this block.

2. Pre-Processor: This block takes original data from the MIT Lincoln Lab, removes the required attributes and converts the dataset to a mutually compatible format. Mainly performs the cleaning process.

3. Encryption: The attributes in the dataset are converted to dual data types so they can be compatible with the Matlab's ANN toolbox. The authors change the variable property "protocol type" values such as tcp = 0, UDP = 1, ICMP = 2; "error marked" with S0 = 0, SF = 1, S1 = 2, REJ = 3, S2 = 4 values. Attacks are classified in the dataset as DOS = 1, 0 = normal, check = 2, R2L = 3, u2r = 4. Information on all the characteristics of 41 in Appendix I [16].

4. Neural Network Classification: In the encoding phase output, data is entered into the neural network so that attack types can be properly categorized.

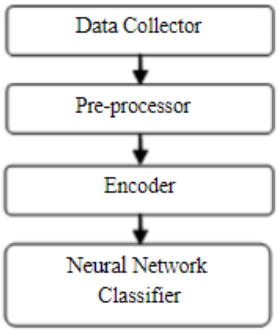

Figure 2 : Sub-processes of the proposed IDS architecture

IV. RESEARCH METHODOLOGY

The first to collect data from NSL-KDD intrusion detection and data collection kits [19] A typical small part of training and testing of data from NSL-KDD datasets for use. Pre-treatment is done [20]. BPN Classifier is designed to detect and categorize

events. BPN [13] Data trained by training algorithm After training and testing, it classifies the vehicles in 23 categories (22 types of attack and general).Backwards algorithm is an essential part of the neural network. The algorithm is learning algorithm rather than a training or a single network. The commonly used network is simple type of illustration in chapter 1 and in the examples shown in Fig. 1.1. These feed networks are called further (we can see why Chapter 7 on Hopfield Network) or sometimes multilayer scenario (MLP). The network works exactly as others have seen. Now let's think what is back posting and how to use it. Behind the promotional network is taught by example. You can give examples of algorithm for what you want to do for the network, and change the load of the network so that when the training is complete, it gives you the output needed for a specific input. As we have just mentioned, you need to train the network for a specific input that what you want (called the goal).

4.1 Data Set

The NSL-KDD data set [21], [22] that was created in 2009, is used extensively in the efforts to detect intrusion. In the latest work [23] - [25], all researchers use NSL-KDD as a reference data set, which not only solve problems of additional recording problems contained in the data set of KDD Cup 1999, but also Make the record number a proper training set. The workbook does not classify records more consistently. The KDDTrain + Dataset Data Set includes KDDTest + and KDDTest-21 datasets as test groups with various natural entries and four different types of attack logs, as shown in Table 1. KDDTest + 21 dataset is a subset of KDDTest +. There are 41 posts and one category tag for each traffic record. Features include basic functions (number 1-number 10), content functions (number 11 - number 2), and traffic feature (number 223 - number 4). Table 2. According to their characteristics, the attacks in the data set have been classified into four types of attacks: DoS (denial of service attacks), R2L (from root to local attacks), U2R (attacking user by root), and Investigation (test attack). There are some specific types of attacks in the test set that disappear into the training package, which makes it possible to provide a more realistic theoretical basis for the detection of intrusion.

Table 1: Features of NSL - KDD dataset

No.	Features	Types	No.	Features	Types
1	duration	Continuous	22	is_guest_login	Symbolic
2	protocol_type	Symbolic	23	count	Continuous
3	service	Symbolic	24	srv_count	Continuous
4	flag	Symbolic	25	serror_rate	Continuous
5	src_bytes	Continuous	26	srv_serror_rate	Continuous
6	dst_bytes	Continuous	27	rerror_rate	Continuous
7	land	Symbolic	28	srv_rerror_rate	Continuous
8	wrong_fragment	Continuous	29	same_srv_rate	Continuous
9	urgent	Continuous	30	diff_srv_rate	Continuous
10	hot	Continuous	31	srv_diff_host_rate	Continuous
11	num_failed_logins	Continuous	32	dst_host_count	Continuous
12	logged_in	Symbolic	33	dst_host_srv_count	Continuous
13	num_compromised	Continuous	34	dst_host_same_srv_rate	Continuous
14	root_shell	Continuous	35	dst_host_diff_srv_rate	Continuous
15	su_attempted	Continuous	36	dst_host_same_src_port_ra	Continuous
16	num_root	Continuous	37	dst_host_srv_diff_host_rat	Continuous
17	num_file_creations	Continuous	38	dst_host_serror_rate	Continuous
18	num_shells	Continuous	39	dst_host_srv_serror_rate	Continuous
19	num_access_files	Continuous	40	dst_host_rerror_rate	Continuous
20	num_outbound_cmds	Continuous	41	dst_host_srv_rerror_rate	Continuous
21	is_host_login	Symbolic			

4.2 Back – propagation Classifier

Use the BPN algorithm [13] to create a BPN class to classify events. • BPN: Step 1) Network design and static parameters Step 2) Initial weight with random values for a specified number of training duplicates: Output from input - 2) Calculate error output neurons - Error hidden neurons - C) Load differences (new by weight) (Step3) Learn from a new weight. For BPN, the parameters are set Backflow algorithm is a very important neural network. The algorithm is a training or learning algorithm rather than a grid. The network used is typically the simplest type shown in Figure 1.1 in Lesson 1 and in example examples. These are called frontal feeding networks (see chapter 7 on the Hopfield network) or sometimes multi-layer perception (MLP). The network works in the same way that others have now seen, let's think about what the back deployment is and how to use it. For example, learning sequential learning networks gives an example of a logarithm of what you want to do in the network so that when the training is completed, it gives you the desired output for a particular entry. Background promotional networks are ideal for identifying simple patterns and mapping work. The grid, you will need to give examples you want to get output (called target) for the given input to.

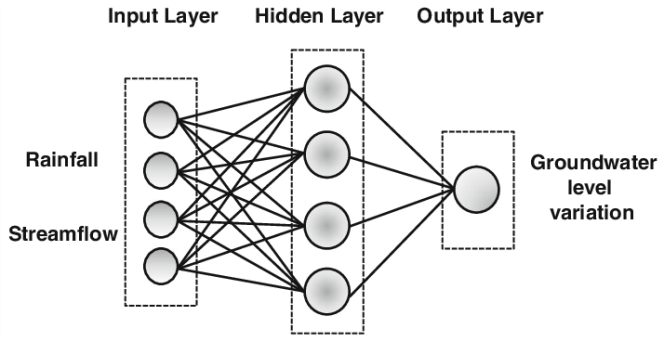

Figure 3 : EBPNN Model

Therefore, if the first pattern of the grid, we would like the result to be 0 1 as shown in Figure 2 (represents the black pixel with the symbol 1 as shown in the previous examples 0 in white). The input and corresponding target are called the training pair.

Once the network is trained, the output required for any input pattern will be provided. Let us now see how the training works. The network is first called between random numbers - 1 and +1. Next, the calculated input and output style (called this path) is applied. The account gives completely different outputs for you (target), because all loads are random. Then calculate the error of each neuron, namely: the targets - the actual outputs (ie what you want - what you actually get) - and then use this error to calculate the weight to make the error smaller. In other words, the output of each neuron will be close to its target (this part is called reverse). The process is repeated repeatedly until the error is reduced. Let's look at a real network example of how the process works. First look at the connection in the output layer between the neurons and in the hidden layer.

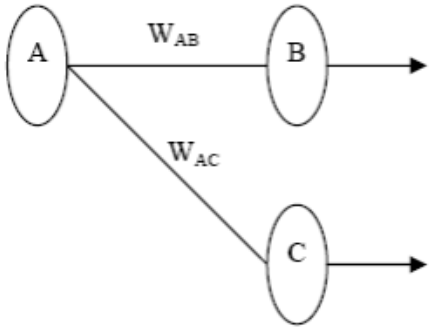

Figure 4: A Single Connection Learning in a Back Propagation network.

Interest in the relationship between neurons A (hidden layer neurons) and B neurons (output neuron) and its weight is WAB. The diagram also depicts another connection between Neuron A and Neuron C, but we will come back to it later. The algorithm works like this:

1. Apply input to the first grid and work on the output - remember that this initial output can be anything, because the initial weight was random numbers.
2. For the following neuron, the error works. The error is what you want - which you already receive, in other words: ERRB = OutputB (1-outputB) (target-outputb) "Output (1-output)" is necessary in the equation because of the sine function - if we were using only a threshold neuron, It will be (target - output) only.
3. Change weight. Explain that W + AB is the new weight (trainers) and WAB is the initial weight. W + AB = WAB + (BX output error) Note that this is the result of connecting the neurons (nerve A) we use (not B). All weights in the output layer are updated like this.
4. Calculate errors for hidden layer neurons. Unlike the output layer, we can not directly calculate it (because we have no target), so we publish it with the output layer. This production is done by taking errors from neurons and returning to weight to get hidden layer errors. For example, if neuron A is connected to B and C, we have A Array = output A (1 - output A) (Error BAB + Arbecc WAC) Again, errors of B and C arise to create an error for a factor. "Output (1 - Output)" exists because of the scowishing function.
5. After getting the hidden neurons error, now go to step 3 to replace the hidden layer. By repeating this method, we can train the network layers by any number. You may have some doubts about this process, so tell clearly that 2 inputs, which clearly show all calculations for a full-size network with 3 hidden layer neurons and 2 neuron output, as shown in 3.4. W + represents the recalculated new weight, where W (without a high mark) represents the old weight. The reverse process is done in the same way. So attackers are calculated.

V. RESULTS AND DISCUSSION

The following results are obtained from EBPNN model and compared with that of RNN model:

Table 2 : Precision Value Comparison

Data-Set Size	Precision Value Comparison	
	RNN	EBPNN
3000	0.889333	0.981552
6000	0.879694	0.981516
9000	0.880034	0.983288

From above Table 2, it is obtained that with the increase in dataset size precision value rate increase. As number of patterns are more in the dataset so results are more accurate.

Table 3 : Recall Value Comparison

Data-Set Size	Recall Value Comparison	
	RNN	EBPNN
3000	0.976574	0.987204
6000	0.978754	0.987705
9000	0.978784	0.987621

From above Table 3, it is obtained that with the increase in dataset size recall value rate increase. As number of patterns are more in the dataset so results are more accurate.

Table 4 : F-Measure Value Comparison

Data-Set Size	F-Measure Value Comparison	
	RNN	EBPNN
3000	0.930914	0.98437
6000	0.926584	0.984601
9000	0.926786	0.98545

From above table 4 , it is obtained that use of EBPNN in proposed work has high F-measure value as compared to RNN work.

Table 5: : Accuracy Comparison

Data-Set Size	Accuracy Value Comparison	
	RNN	EBPNN
3000	0.834667	0.991231
6000	0.834333	0.981003
9000	0.836222	0.982335

From above Figure 5, it is obtained that with the increase in dataset size accuracy value increase.

V. CONCLUSION

In this work, to solve problems with computer networks, there is a special way to protect computer network problems in intrusion infiltration systems (IDS) in a specific way to prevent security problems. Many steps have been taken to improve network security and the network is becoming increasingly important. To detect intrusion into the network IDS systems were developed. The main target of this paper was to find the type of session any general or infiltration where infiltration was found compared to the infiltration category. The entire work is designed so that the automatic assembly of different sessions using the steps of the genetic algorithm takes into account data in the form of sets of inputs in the neural network of training. Therefore, there was a need for a special determination in this work to separate the meeting. This work was done using training and testing by the neural network propagation error. The experiment was conducted on the actual data set where different sets of test data were passed to compare to different evaluation criteria. There are many ways to verify the type of problems you encounter in network concepts such as remote service (R2L), local remote (U2R), authentication, etc. It is used to detect intrusion in the network. The results obtained are highly appreciated because the trained networks can detect all types of penetration accuracy of over 99%.

VI. REFERENCES

[1]Koushal Kumar, Jaspreet Singh Batth "Network Intrusion Detection with Feature Selection Techniques using Machine-Learning Algorithms" International Journal of Computer Applications (0975 – 8887) Volume 150 – No.12, September 2016

[2] R.Karthik, Dr.S.Veni, Dr.B.L.Shivakumar "Improved Extreme Learning Machine (IELM) Classifier For Intrusion Detection System" International Journal of Engineering Trends and Technology (IJETT) – Volume-41 Number-2 - November 2016

[3] 1Premansu sekhara rath, 2manisha mohanty, 3silva acharya, 4monica aich "optimization of ids algorithms using data mining technique" International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982 Volume-4, Issue-3, Mar.-2016

[4] Mohammadreza Ektefa, Sara Memar, Fatimah Sidi, Lilly Suriani Affendey "Intrusion Detection Using Data Mining Techniques", 978-1-4244-5651-2/10/$26.00 ©2010 IEEE

[5] YU-XIN MENG," The Practice on Using Machine Learning For Network Anomaly Intrusion Detection" Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, 978-1-4577-0308-9/11/$26.00 ©2011 IEEE.

[6] Liu Hui, CAO Yonghui "Research Intrusion Detection Techniques from thePerspective of "Machine Learning" 2010 Second International Conference on MultiMedia and Information Technology 978-0-7695-4008-5/10 $26.00 © 2010 IEEE

[7] Jingbo Yuan , Haixiao Li, Shunli Ding , Limin Cao "Intrusion Detection Model based on Improved Support Vector Machine", Third International Symposium on Intelligent Information Technology and Security Informatics 978-0-7695-4020-7/10 $26.00 © 2010 IEEE

[8] Kamarularifin Abd Jalill, Mohamad Noorman Masrek "Comparison of Machine Learning Algorithms Performance in Detecting Network Intrusion" 201O International Conference on Networking and Information Technology 978-1-4244-7578-0/$26.00 © 2010 IEEE

[9] Devendra kailashiya, Dr. R.C. Jain "Improve Intrusion Detection Using Decision Tree with Sampling" Vol 3 (3), 1209-1216 ijcta 2012

[10] Megha Aggarwal, Amrita "Performance Analysis Of Different Feature Selection Methods In Intrusion Detection" INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 2, ISSUE 6, JUNE 2013.

[11] Kaberi Das, Prem Pujari Pati, Debahuti Mishra, Lipismita Panigrahi "Empirical Comparison of Sampling Strategies for Classification" ICMOC-2012, Elsevier science direct.

[12] Ligang Zhou," Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods." Contents lists available at SciVerse ScienceDirect Knowledge-Based Systems journal homepage: www.elsevier.com/locate/knosys online 3 January 2013

[13] Nitesh V. Chawla "Data mining for imbalanced datasets: an overview" sprimger.

[14] KDD CUP 1999. Availabe on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html October 2007

[15] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", 2009 IEEE

[16] Ch Ambedkar ,V Kishore Babu, " Detection of Probe Attacks Using Machine Learning Techniques",InternationalJournal of Research StudiesIn Computer Science And Engineering, Volume 2, Issue 3, March 2015

[17]Rajesh Wankhede, Vikrant Chole,"Intrusion Detection System Using Classification Technique", International Journal of Computer Applications(0975 – 8887) Volume 139 – No.11, April 2016

[18] Mr.Kamlesh Patel, Mr.Prabhakar Sharma, " An Implementation of Intrusion Detection System Based on Genetic Algorithm" International Journal of Advanced Research in Computer and Communication Engineering,Vol. 5, Issue 11, November 2016.

[19] Arun K Pujari "Data mining techniques" Universities Press.

[20] Subaira A. S., Anitha P. "An Efficient Classification Mechanism For Network Intrusion Detection System Based on Data Mining Techniques:A Survey" International Journal of Computer Science and Business Informatics 2013.