Performance Analysis of Network Intrusion Detection System using Back Propagation for Feed Forward Neural Network in MATLAB/SIMULINK

Er. Srinivas Mishra [1], Dr. Sateesh Kumar Pradhan [2] AND Dr. Subhendu Kumar Rath [3]

[1] PhD Scholar, Dept. of Comp. Sc. & Engg., Biju Patnaik University of Technology, Odisha, India,
[2] Prof. & Head, P.G dept. of Computer Sc., Utkal University, Odisha, India
[3] Deputy Registrar, Biju Patnaik University of Technology (BPUT), Odisha, India,
Correspondence Author: Er. Srinivas Mishra

ABSTRACT

Security of an information system is its very important property, especially today, when computers are interconnected via internet. Because no system can be absolutely secure, the timely and accurate detection of intrusions is necessary. For this purpose, Intrusion Detection Systems (IDS) were designed. Most IDS commercial tools are misuse systems with rule-based expert system structure. However, these techniques are less successful when attack characteristics vary from built-in signatures. Different soft-computing based methods have been proposed in recent years for the development of intrusion detection systems. Artificial neural networks offer the potential to resolve these problems. For building anomaly system, neural networks can be used, because they can learn to discriminate the normal and abnormal behavior of a system from examples. Therefore, they offer a promising technique for building anomaly systems. Artificial neural networks provide the potential to identify and classify network activity based on limited, incomplete, and nonlinear data sources. This paper investigates the application of the Feed Forward Neural Network trained by Back Propagation algorithm for intrusion detection. The developed network can be used to identify the occurrence of various types of intrusions in the system. Simulation result shows that the proposed approach detects the intrusions accurately and is well suitable for real time applications.

KEY WORDS: Intrusion Detection System (IDS), Back-propagation Neural Networks, Artificial Neural Network (ANN), Feed Forward Neural Network, Network Security, Soft Computing.

Date of Submission: 12-05-2018 Date of acceptance: 28-05-2018

I INTRODUCTION

The conventional approach to secure information system is to build a protective shield around it. For this purpose different methods of indentation, authentication and mandatory access control techniques are used [1]. Soft computing is a general term for describing a set of optimization and processing techniques that are tolerant of imprecision and uncertainty [2]. The principal constituents of soft computing techniques are Fuzzy Logic (FL), Artificial Neural Network (ANN), Probabilistic Reasoning (PR) and Genetic Algorithm (GA) [3]. 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. Recently, Artificial Neural Networks have been successfully applied for developing the IDS because it has the advantage of easier representation of nonlinear relationship between input and output and is inherent by fast. Even if the data were incomplete or distorted, a neural network would be capable of analyzing the data from a network [4]. Nowadays, most commercial IDSs use rules to create attack pattern. Rule-based systems like expert systems follow fixed rules which should be periodically updated. These intelligent systems construct a general model of existing patterns which will be able to detect new ones [5]. In this paper, a network based

IDS, using a supervised 9 layer feed-forward neural network with back propagation, is proposed. This system can distinguish normal connections and attacks.
The paper is organized as follows: section 2, introduces literature review and related works. Section 3, introduces concept of artificial neural networks. Section 4, describes proposed IDS architecture. Section 5, evaluates the proposed system and results were analyzed. Future scope is described in Section 6 and at last, Section 7 presents the conclusion of this work.

II LITERATURE REVIEW

Several IDSs employ intelligent methods. Heywood et al. [6] propose a hierarchical neural network for intrusion detection based on SOM (Self Organizing Map). This three-layered hierarchical SOM architecture uses two sets of features, one is limited to 6 basic KDD features and the other consists of all 41. Jirapunimm et al. [7] use combination of SOM and MLP (Multi-Layer Perceptions). SOM is used as a preprocessing level and its outputs are fed to MLP as inputs. This hybrid network is formed as a 5-layered feed-forward neural network. The first layer is input layer, the second one is SOM layer and 3 next layers are MLP layers. J. Shum et al. [8] designed an intrusion detection system based on feed-forward neural network with back propagation. Their network composed of an input layer, a hidden layer and an output layer. E. Hernandez-pereira et al. [9] utilized three techniques: SVM (support vector machine), one layer and multilayer feed-forward neural networks. They focused more on conversion of symbolic features to numerical ones and compare effect of different conversion techniques on intrusion detection.

Different machine learning mechanisms, including Artificial Neural Networks, Fuzzy Logic, Genetic Algorithms, etc. have been used on KDD CUP 1999 data for Intrusion Detection [10]-[18]. Different neural network algorithms have been used, including Grey Neural Networks [14], RBF [18],[19] Recirculation Neural Networks [12], PCA and MLP [15], with MLP generally showing better results than others [12].These works are mainly focusing on misuse detection. In order to combine misuse and anomaly detection, many researchers have recently attempted hybrid methods, by combining neural networks with other machine learning mechanisms, such as fuzzy logic or genetic algorithms [11], [15].

III ARTIFICIAL NEURAL NETWORK

Artificial neural networks born after McCulloc and Pitts introduced a set of simplified neurons in 1943. These neurons were represented as models of biological networks into conceptual components for circuits that could perform computational tasks [20]. The basic model of the artificial neuron is founded upon the functionality of the biological neuron [21]. By definition, “Neurons are basic signalling units of the nervous system of a living being in which each neuron is a discrete cell whose several processes are from its cell body. One can differentiate between two basic types of networks, networks with feedback and those without it. In networks with feedback, the output values can be traced back to the input values. However there are networks wherein for every input vector laid on the network, an output vector is calculated and this can be read from the output neurons, there is no feedback. Hence only, a forward flow of information is present. Network having this structure are called as feed forward networks. This network has one input layer, one hidden layer and one output layer. There can be any number of hidden layers. The input layer is connected to the hidden layer and the hidden layer is connected to the output layer by means of interconnection weights. The bias is provided for both the hidden and the output layer, to act upon the net input to be calculated. Neural network is composed of several processing units (nodes) and directed links between them. These connections are weighted representing relation between input and output neurons [22]. Neural networks are classified into two groups based on connections:

III.1 Feed-forward Neural Network:

The Multilayer feed-forward neural network has several neurons structured in layers such as input layer, hidden layers and output layers (Figure 1). Output layer with one or many neurons provides output for one or many inputs. In one neuron example, training process task is to find proper weights for neuron connections which in combination with inputs, achieves the desired output. This process is accomplished by back propagation algorithm [23].

III.2 Back Propagation Neural Network:

Back propagation neural network propagates the error from the output layer to the hidden layers and changes weights recursively through network from output layer to input layer. The main objective of algorithm is to minimize output error by changing weights. Back propagation algorithm is based on gradient descent. In each step, the goal gradient is computed which direction of negative gradient represents the direction where the surface decreases more rapidly and amount of gradient shows the distance through which the direction is valid.

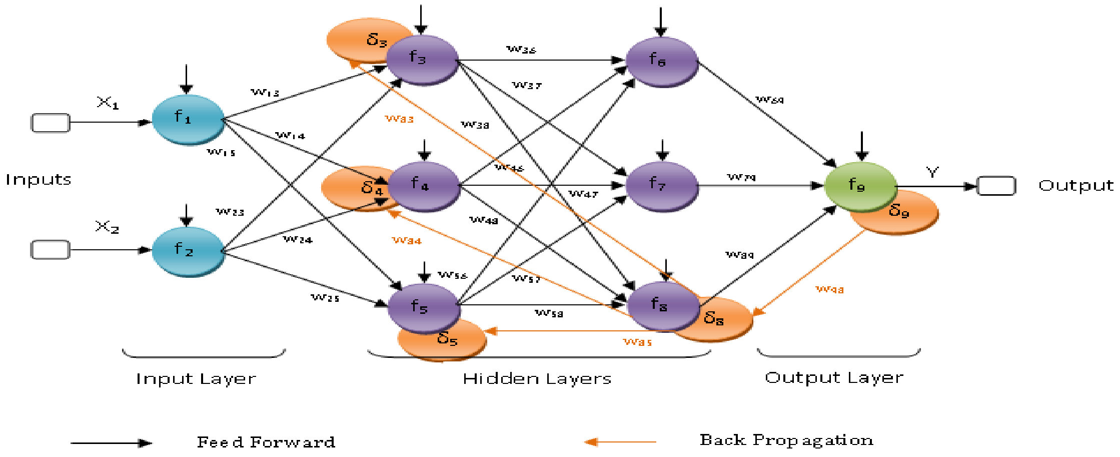

Figure 1. Feed Forward Neural Networks with Back Propagation of errors.

Based on the above figure, the output Y can be computed as $Y = f_9 (w_{69}Y_6 + w_{79}Y_7 + w_{89}Y_8)$
And the back propagation error δ_5 for the neuron f_5 can be computed as $\delta_5 = w_{98}\ \delta_9 + w_{85}\ \delta_8$
In feed-forward networks the flow of data goes from input to output cells, which can be grouped into layers but no feedback interconnections can exist [10]. On the other hand, recurrent networks contain feedback loops and their dynamical properties are very important.

IV THE PROPOSED METHOD

The proposed IDS is structured as a feed forward neural network with back propagation algorithm. Neural network properties like parallelism, distributed computation, learning capability, adaptively and fault tolerance made it suitable for intrusion detection systems. Also as feed forward neural network (with one or more hidden layer) can estimate every function with desired precision and its simplicity over many other neural networks [24], we choose this network for our IDS. The proposed system has three phases: preprocessing, training and detection, which illustrated in Figure 2.

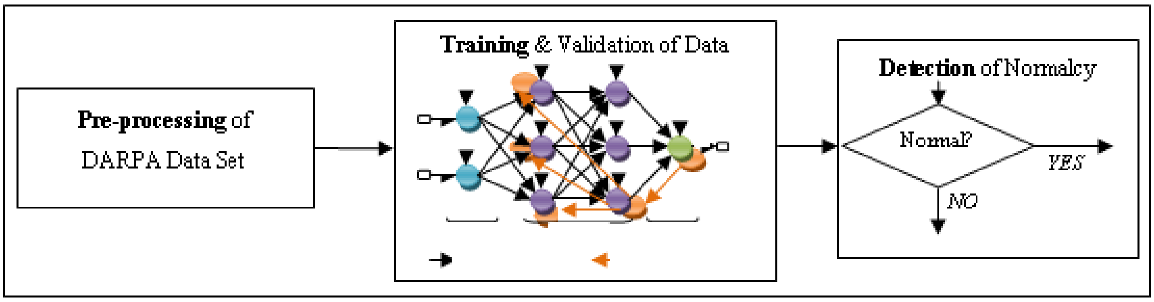

Figure 2. Overall IDS system.

IV.1 TRAINING ALGORITHM:

The training algorithm of back propagation involves four stages, viz.
1. Initialization of Weights
2. Feed Forward
3. Back Propagation of errors
4. Updation of the weights and the biases.

During first stage which is the initialization of weights, some small random values are assigned. During feed forward stage each input unit (X_i) receives an input signal and transmits this signal to each of the hidden units Z_1………Z_p. Each hidden unit then calculates the activation function and sends its signal Z_j to each output unit [25]. The output unit calculates the activation function to form the response of the net for the given input pattern. During back propagation of errors, each output unit compares its computed activation y_k with its target value t_k to determine the associated error for that pattern with that unit. Based on the error, the factor δ_k is computed and is used to distribute the error at output unit y_k back to all units in the previous layer. Similarly factor δ_j is computed for each hidden unit z_j. During final stage, the weight and biases are updated using the δ factor and the activation function.

The training algorithm used in the back propagation network is as follows. The algorithm is given with the various phases:

Initialization of Weights:
Step-1: Initialize weight to small random values.
Step-2: While stopping condition is false, do Steps 3 to 10.
Step-3: For each training pair do steps 4 to 9.

Feed Forward:
Step-4: Each input unit receives the input signal x_i and transmits these signals to all units in the layer above i.e hidden units.
Step-5: Each hidden unit z_j *(j=1,2……,p)* sums its weighted input signals.

$z_{inj}=v_{oj}+\Sigma x_i v_{ij}$

applying activation function $Z_j=f(z_{inj})$ and sends this signal to all units in the layer above i.e. output units.
Step-6: Each output unit y_k sums its weighted input signals.
$y_{ink}=w_{ok}+\Sigma z_j w_{jk}$
and applies its activation function to calculate the output signals.
$Y_k=f(y_{ink})$

Back Propagation of Errors:
Step-7: Each output unit receives a target pattern corresponding to an input pattern, error information term is calculated as
$\delta_k=(t_k-y_k)f(y_{ink})$
Step-8: Each hidden unit z_j sums its delta inputs from units in the layer above
$\delta_{inj}=\Sigma\delta_j w_{jk}$
The error information term is calculated as
$\delta_j=\delta_{inj}f(z_{inj})$

Updation of Weight and Biases:
Step-9: Each output unit y_k updates its bias and weights *(j=0, 1, 2…... p)*
The weight correction term is given by
$\Delta W_{jk}=\alpha\delta_k z_j$
and the bias correction term is given by
$\Delta W_{ok}=\alpha\delta_k$
$W_{jk}(new) = W_{jk}(old) + \Delta W_{jk}$
$W_{ok}(new)=W_{ok}(old) + \Delta W_{ok}$
Each hidden unit z_j *(j=1, 2…….p)* updates its bias and weights *(i=0, 1, 2…..n)*
The weight correction term
$\Delta V_{ij}=\alpha\delta_j x_i$
The bias correction term
$\Delta V_{oj}=\alpha\delta_j$
$V_{ij}(new) = V_{ij} (old) + \Delta V_{ij}$
$V_{oj}(new)= V_{oj} (old) + \Delta V_{oj}$
Step-10: Test the stopping condition. The stopping condition may be the minimization of the errors, number of epochs etc.

The question is if anything is gained by using more than one hidden layer. One answer is that using more than one layer may lead to more efficient approximation or to achieving the same accuracy with fewer neurons in the neural network. MATLAB^{TM} Neural Network Toolbox [26] was used for the implementation of the MLP networks. Using this tool one can define specifications like number of layers, number of neurons in each layer, activation functions of neurons in different layers, and number of training epochs. Then the training feature vectors and the corresponding desired outputs can be fed to the neural network to begin training. Error back-propagation algorithm was used for training.
Each neuron within the hidden layer is represented by transfer function known as activation function. The transfer function should be able to accept an input within any range, and to produce an output in a strictly limited range. We used one of the most common transfer functions, the logistic function. In this case, the output is in the range (0, 1), and the input is sensitive in a range not much larger than (-1, +1). This function is also smooth and easily differentiable. These properties are critical in allowing the network training algorithms to operate [27]. Back-propagation (backward propagation of errors) is supervised learning algorithm, which is the

most useful for feed forward networks. The algorithm can be divided into two main phases, propagation phase and weight update. In propagation phase, inputs are passed to the hidden layer where the initial weights were set. The hidden layer produces certain outputs, which are evaluated against original outputs, and error is calculated. In the second phase, the calculated error is used to update neuron weights. The process continues until optimum weights, are obtained.

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 [28]. 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 [29]:

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

V RESULTS AND DISCUSSION

The training model was performed by means of Root Mean Square (RMS) error analysis [30] using learning rate of 0.80, 2 input layers, 6 hidden layers and 1 output layer. There were three categories of incorrect outputs: false positive, false negative and irrelevant neural network output. The irrelevant outputs were those that did not represent any of the output classes [31] in the data set. The mean square error achieved by the network during training is 9.9979e-004. With six hidden nodes, the network took 249.7030 seconds to reach the error goal. The performance of network during training is shown in below figure.

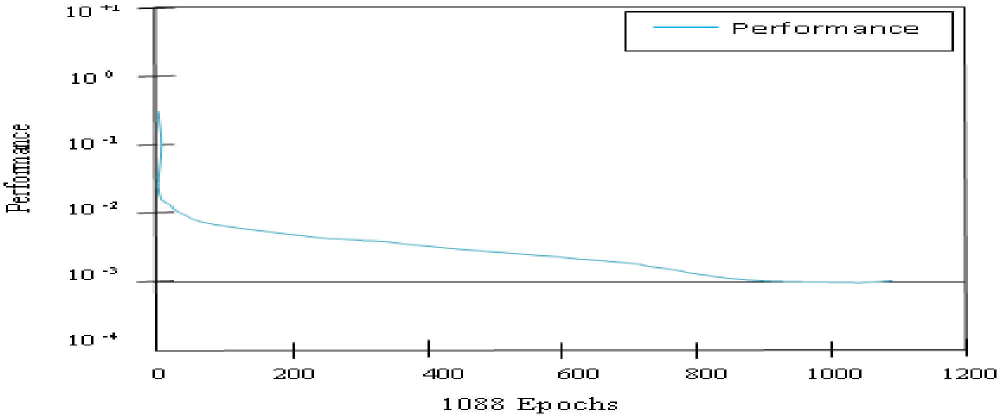

Figure 3. Training Performance of the network.

After training, the generalization performance of the network is evaluated with the test data. During testing the Mean Square Error achieved by the network is 4.2758e-004.

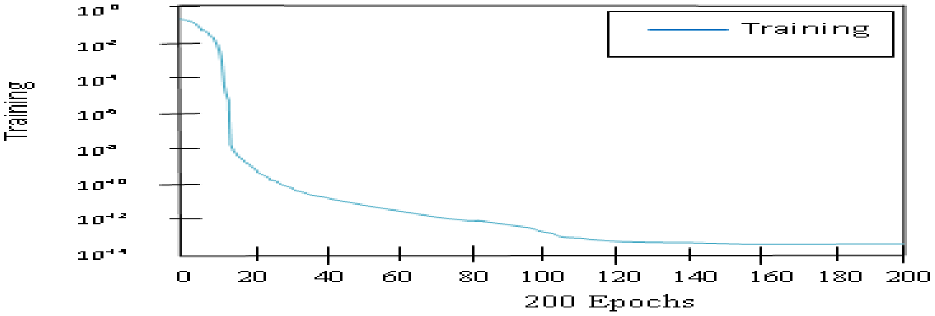

Figure 4. The mean square error (mse) of the back-propagation training procedure versus training epochs.

The decrease in the error was completely satisfactory. The network was over fitted.

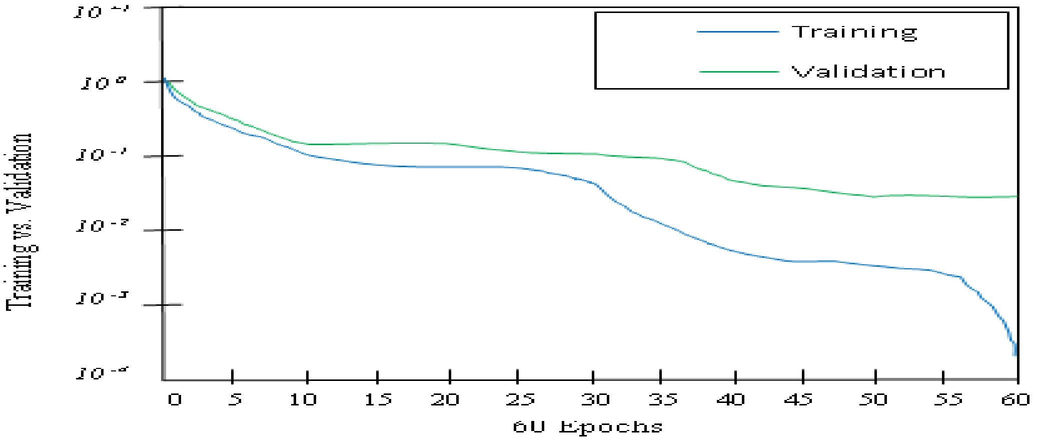

Figure 5. The training process error when the early stopping validation method is applied.

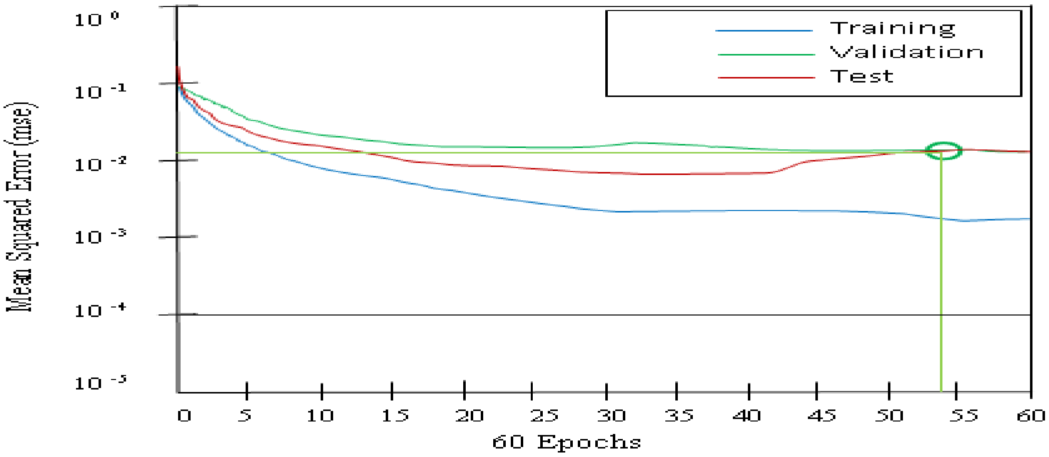

Figure 6. Mean Squared Error (mse).

It has been shown that the best classification was obtained at epoch 54 with 0.00405 mse. At this point, test and validation data gets the best common minimum.

VI FUTURE SCOPE

As mentioned above, there has been a lot of research on intrusion detection, and also on the use of neural networks in intrusion detection. As showed in this paper, back propagation neural networks can be used successfully to detect attacks on a network. The same experiments should also be conducted with other types of neural networks to see if these types can improve the detection rate we got from the experiments with a back propagation neural network. As a possible future development to the present study, one can include more attack scenarios in the dataset. Practical IDSs should include several attack types. In order to avoid unreasonable complexity in the neural network, an initial classification of the connection records to normal and general categories of attacks can be the first step. The records in each category of intrusions can then be further classified to the attack types.

VII CONCLUSION

In the design of an anomaly detection system one can take advantage of the neural network ability to learn and of its capability to generalize. Neural network can learn to discriminate between normal and abnormal behaviour of the system from examples. No explicit definition of abnormal behaviour of the system is necessary and thus the main obstacle in building anomaly system could be overcome. There are various techniques of Artificial Neural Network, which can be applied to Intrusion Detection System. Each technique is suitable for some specific situation. BPNN is easy to implement, supervised learning artificial neural network. Number of the epochs required to train the network is high as compare to the other ANN technique but, detection rate is very high. BPNN can be used when one wants to not only detect the attack but also to classify the attack in to specific category so that preventive action can be taken. By combining the different ANN techniques, one can reduce the number of the epochs required and hence can reduce the training time. The work does not require any additional hardware and is software based.

We applied the early stopping validation method which increased the generalization capability of the neural network and at the same time decreased the training time. Therefore, the neural network based IDS can operate as an online classifier for the attack types that it has been trained for. The only factor that makes the neural network off-line is the time used for gathering information necessary to compute the features. Although it's simple structure, in comparison with similar IDSs, it achieves equivalent performance and reduces computational overhead and memory usage. The overall system showed the classification of 94.82%, with 0.6% both, false positive and false negative rate, and mean square error of 0.004.

REFERENCES

[1]. A. Vesel Y. and D. Brechlerov A, "Neural Network in Intrusion Detection Systems", the international conference Agrarian perspectives XII (CUA prague, September 18-19, 2003), AGRIC. ECON.- CZECH, 50,2004(1), PP. 35-39.

[2]. Mehdi MORADI and Mohammad ZULKERNINE, "A Neural Network Based System for Intrusion Detection and Classification of Attacks", the Natural Sciences and Engineering Research Council of Canada (NSERC). 148-04.

[3]. Piero P. Bonissone, "Soft computing: the convergence of emerging reasoning technologies," Soft Computing Journal, vol.1, no. 1, pp. 6-18, Springer-Verlag 1997.

[4]. P. Ganesh Kumar and D.Devaraj, "INTRUSION DETECTION USING ARTIFICIAL NEURAL NETWORK WITH REDUCED INPUT FEATURES", ICTACT JOURNAL ON SOFT COMPUTING, JULY 2010, VOLUME: 01, ISSUE: 01, DOI: 10.21917/ijsc.2010.0005, pp. 30-36.

[5]. Fariba Haddadi, Sara khanchi, Mehran Shetabi and Vali Derhami, "Intrusion Detection and Attack Classification Using Feed-Forward Neural Network", Second International Conference on Computer and Network Technology, 978-0-7695-4042-9/10 © 2010 IEEE DOI 10.1109/ICCNT.2010.28, IEEE Computer Society, pp. 262-266.

[6]. H. Kayacik, A. Zincir Heywood and M. Heywood, "A hierarchical SOM-based intrusion detection system", in Proc. Elsevier Engineering Application of Artificial Intelligence,2007, pp. 439-451.

[7]. C. Jirapummin, N. Wattanapongsakorn and P. Kanthamanon, "Hybrid neural networks for intrusion detection system", Proceedings of the 2002 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2002), pp. 928-931, Thailand, 2002.

[8]. J. Shum and H.A. Malki, "Network intrusion detection system using neural network", in Proc. IEEE Fourth Int. Conference on Natural Computation, pp. 242-246, 2008.

[9]. E. Hernandez pereira, J.A. Suarez Romero, O. Fontela Romero and A. Alonso Betanzos, "Conversion methods for symbolic features: A comparison applied to an intrusion detection problem", in Proc. Elsevier Expert Systems with Applications, 2009.

[10]. Alma Husagic Selman, Rasit Koker and Suvad Selman, "Intrusion Detection using Neural Network Committee Machine", 2013 XXIV International Conference on Information, Communication and Automation Technologies (ICAT) October 30 - November 01, 2013, Sarajevo, Bosnia and Herzegovina, 978-1-4799-0431-0/13 ©2013 IEEE.

[11]. S. Chavan, K. Shah, N. Dave, S. Mukherjee, A. Abraham and S. Sanyal, "Adaptive neuro-fuzzy intrusion detection systems", Proceedings of ITCC 2004, International Conference on Information Technology: Coding and Computing , vol. 1, pp. 70-74.

[12]. P. Kachurka and V. Golovko, "Neural network approach to real time network intrusion detection and recognition", IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS) , vol. 1, 2011, pp. 393-397.

[13]. J. Zhao, M. Chen, and Q. Luo, "Research of intrusion detection system based on neural networks", IEEE 3rd International Conference on Communication Software and Networks (ICCSN), 2011, pp. 174-178.

[14]. D.X. Xia, S.H. Yang and C.G. Li, "Intrusion detection system based on principal component analysis and grey neural networks", Second International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC), vol. 2, 2010, pp. 142-145.

[15]. M. Muna and M. Mehrotra, "Design network intrusion detection system using hybrid fuzzy-neural network", International Journal of Computer Science and Security, vol. 4, no. 3, pp. 288-294, 2010.

[16]. G. Wang, J. Hao, J. Ma and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering", Expert Systems with Applications, vol. 37, no. 9, pp. 6225 – 6232, 2010.

[17]. (1999, October) Kdd cup 99 competition. [Online], Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[18]. C. Zhang, J. Jiang and M. Kamel, "Intrusion detection using hierarchical neural networks", Pattern Recognition Letters, vol. 26, no. 6, pp. 779 -791, 2005.

[19]. U. Ahmed and A. Masood, "Host based intrusion detection using rbf neural networks", International Conference on Emerging Technologies, ICET 2009, pp. 48-51.

[20]. Tapasya Pandit and Anil Dudy, "A Feed Forward Artificial Neural Network Based System to Minimize Dos Attack in Wireless Network", International Journal of Advances in Engineering & Technology, July 2014, ISSN: 22311963, Vol. 7, Issue 3, pp. 938-947.

[21]. Afrah Nazir, "A Comparative Study of different Artificial Neural Networks based Intrusion Detection Systems", International Journal of Scientific and Research Publications, Volume 3, Issue 7, July 2013.

[22]. S. J. Russell and P. Norvig, "Artificial intelligence: A modern approach (international edition)", Pearson US Imports & PHIPEs, November 2002.

[23]. S. Haykin, "Neural networks: A comprehensive foundation", McMillan, New York, 1994.

[24]. S.Theodorios and K. Koutrrombas, "Pattern recognition", Cambridge: Academic Press, 1999.

[25]. Amit Garg and Ravindra Pratap Singh, " Voltage Profile Analysis in Power Transmission System based on STATCOM using Artificial Neural Network in MATLAB/SIMULINK", International Journal of Applied Information Systems(IJAIS), Foundation of Computer Science, New York, USA, Volume 6, No. 1, September 2013.

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

[27]. R. E. Schapire, "The strength of weak learnability", Machine Learning, vol. 5, pp. 192-227, 1990.

[28]. Manoranjan Pradhan, Sateesh Kumar Pradhan and Sudhir Kumar Sahu, "Anomaly Detection using Artificial Neural Network", International Journal of Engineering Sciences & Emerging Technologies, April 2012.

[29]. P. Ganesh Kumar, D. Devaraj and V. Vasudevan, “Artificial Neural Network for Misuse Detection in Computer Network”, Proceedings of the International Conference on Resource Utilization and Intelligent Systems (INCRUIS-2006), pp. 889-893.

[30]. Sodiya A.S, Ojesanmi O.A and Akinola O.C, “Neural Network based Intrusion Detection Systems”, International Journal of Computer Applications, Volume 106, No. 18, November 2014, pp. 19-24.

[31]. Pinal J. Patel, J.S.Shah and Jinul Patel, “Performance Analysis of Neural Networks for Intrusion Detection System”, International Journal of Computer Technology & Applications, Vol 8(2), ISSN: 2229-6093, Mar-Apr 2017, pp. 88-93.

AUTHORS:

Er. Srinivas Mishra is a Ph. D. scholar in Computer Science & Engineering Dept. under Biju Patnaik University of Technology, Odisha, India. He has published 4 international journal papers and his research interests include Computer Security, Intrusion Detection, Data Mining and Database Security.

Dr. Sateesh Kumar Pradhan is working as a Prof., Post Graduate Department of Computer Science, Utkal University, Odisha, India. He has guided more than 16 Ph. D. Scholars and has published more than 54 international / national journal papers and many conference papers and book chapters. His area of research includes Neural Computing, Cloud Computing, Wireless Sensor Network, Mobile Computing and Data Mining.

Dr. Subhendu Kumar Rath is working as Deputy Registrar at Biju Patnaik University of Technology (BPUT), Odisha, India. He has completed M.Sc, M.Phil, M.Tech, Ph.D, and guided many Ph. D. Scholars and Post Graduate / Under Graduate students and published many journal papers nationally and internationally.

Er. Srinivas Mishra." Performance Analysis of Network Intrusion Detection System using Back Propagation for Feed Forward Neural Network in MATLAB/SIMULINK.” International Journal of Computational Engineering Research (IJCER), vol. 08, no. 05, 2018, pp. 58-65.