A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection

Sanjiban Sekhar Roy[1(✉)], Abhinav Mallik[1], Rishab Gulati[1], Mohammad S. Obaidat[2,3], and P.V. Krishna[4]

[1] School of Computer Science and Engineering, VIT University, Vellore, India
sanjibanroy09@gmail.com, gulati.rishab5@gmail.com, abhinavmalik94@yahoo.com
[2] Fordham University, New York, USA
m.s.obaidat@ieee.org
[3] University of Jordan, Amman, Jordan
[4] Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati, India
dr.krishna@ieee.org

Abstract. Security of data is considered to be one of the most important concerns in today's world. Data is vulnerable to various types of intrusion attacks that may reduce the utility of any network or systems. Constantly changing and the complicated nature of intrusion activities on computer networks cannot be dealt with IDSs that are currently operational. Identifying and preventing such attacks is one of the most challenging tasks. Deep Learning is one of the most effective machine learning techniques which is getting popular recently. This paper checks the potential capability of Deep Neural Network as a classifier for the different types of intrusion attacks. A comparative study has also been carried out with Support Vector Machine (SVM). The experimental results show that the accuracy of intrusion detection using Deep Neural Network is satisfactory.

Keywords: Security · Intrusions · Deep Neural Network · Support Vector Machine

1 Introduction

Intrusion Detection System [1, 2] is a type of security management system for computers and networks. It gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). ID uses vulnerability assessment, developed to assess the security of a computer system or network. Data is considered to be the most important aspect of any organization. If the organization's data is secure, only then it can successfully carry out its operations. However, data have always been under a constant threat from external attacks. The hackers and crackers come up with new ways every day to destroy or steal the data that every organization holds so precious. In this paper, we have analyzed a

D. Giri et al. (Eds.): ICMC 2017, CCIS 655, pp. 44–53, 2017.
DOI: 10.1007/978-981-10-4642-1_5

dataset containing information about the various attacks that have been carried out by the hackers and based on the parameters, an attempt to predict the kind of attack that will be used by the hacker, is carried out. The data set has been obtained from UCI machine learning repository. The data set is related to intrusion detection system (IDS) and in this work, a Deep learning [3] approach based on neural network has been adopted to predict different types of IDS attacks.

An Intrusion Detection System, popularly known as IDS, is a software that monitors the network for malicious activities or violations of policies regarding cybercrime and produces a report to the management system. IDS is related to network security just like a firewall, it differs from a firewall in the manner of looking for intrusions. The firewall looks at the outward intrusions in order to prevent them and limits the access between networks to prevent intrusion. On the other hand, IDS evaluates an intrusion that has already taken place and then sends an alarm signal. A lot of predictions has been accomplished using machine learning [4, 5, 12, 13, 15]. Also, several intrusion detection systems were proposed by several authors using roughest theory and other methods [7]. In this paper, we have used a multilayer feed forward network to represent a deep learning concept for IDS. The feed forward network includes input layers, about 400 hidden layer neurons and output neurons. The activation functions used are rectifier activation function and softmax activation function.

Deep learning has been used in this paper. It is a branch of machine learning that attempts to model higher level abstractions in data by using model architectures with non-linear transformations [6]. It is chosen since it focuses on computational models for information representation. It is implemented in such a way that it is able to display classification invariance with respect to a wide range of transformations and distortions. It enables us to train a network having a large set of observations and excerpt signals from this network. The deep learning algorithms use simple features in the lower layers and more complex features in the higher layers. Here, each hidden layer has statistical knowledge about the lower layers while higher layer representations are more complex. The network is trained using greedy layer-wise training which involves the training of the hidden layers one at a time in a bottom-up fashion. Deep learning has a myriad of applications. It is used in the medical field where robotics surgery is becoming a common trend, which relies extensively on tactile equipment. Deep learning is utilized for developing the robotic equipment. This may enable the doctors to move to a precision of a millimeter. Also, we can see the application of deep learning in the field of automotive in terms of self-driving cars, which apply the concepts of deep learning to emulate the senses of sight and hearing. It is also used in military forces in a country where a large number of military drones utilize the concept of deep learning to follow a moving target. Much research is required in this field as it is not yet fully functional. Currently, Google Brain is a technology used by Google that uses neural networks to recognize high level inputs only from watching unlabeled images from YouTube.

IDS set has been used in the Support Vector Machine (SVM) as well and the result is juxtaposed with the one obtained by using the Neural Network. The results obtained from the Support Vector Machine are complimentary to the ones obtained by using Neural Networks. Thus, it confirms that the results obtained are satisfactory.

2 Deep Neural Network

The neural network used is a multilayer feed forward neural network. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes [8]. There are no cycles or loops in the network. Each neuron in one layer has direct connections to the neurons in the subsequent layers. It contains an input layer, a number of hidden layers and an output layer. The back propagation method is used for learning the weights of the network. The input layer has an identity function as its activation function. The output layer and the hidden layers may have rectifier or softmax activation function. Also, a multilayer neuron does not have a linear activation function in all its neurons. Some of its neurons might have a nonlinear activation function (Fig. 1).

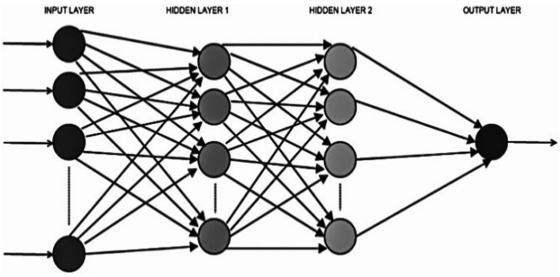

Fig. 1. Feed forward neural network [18]

Feed forward neural network is popular due to 2 factors:

(i) It has the ability to give very closely related approximations for complex multivariate nonlinear function directly from input values.
(ii) It has a strong modelling capability for a large class of natural and artificial phenomena.

However, in most of the practical scenarios, all parameters of a feed forward network need to be adjusted in a backward way which leads to creation of dependencies among various neurons in various layers.

Mean squared error (MSE) measures the average of the squares of the "errors", that is, the difference between the estimator and what is being estimated [9].

The mean square error is calculated in the following way:

$$\begin{aligned} \text{MSE} &= \text{RSS}/\text{N} \\ &\text{where MSE} - \text{Mean Squared Error} \\ &\text{RSS} - \text{Residual Sum of Squares} \\ &\text{N} - \text{Population Size} \end{aligned}$$

RSS is also known as Sum of Squared Residuals (SSR) and Sum of Squared Error (SSE). It is given by [9, 10],

$$\mathrm{RSS} = \sum (y_i - f(x_i))^2 \tag{1}$$

So, MSE is given by,

$$\frac{1}{N}\sum (y_i - f(x_i))^2 \tag{2}$$

The value of R^2 denotes how close the obtained result is to the expected regression line. R^2 can have a value within the range [0,1]. The higher value of R^2, the more accurate the obtained result is. It can be computed in the following way:

$$\begin{aligned} R^2 &= SS_R/SS_T \text{ where,} \\ SS_T &= \sum (y_i - f(y_i))^2 \end{aligned} \tag{3}$$

$$SS_R = \sum (\hat{y}_l - \overline{y})^2 \tag{4}$$

In some of the research experiments, another class of neural network is used which is known as deep belief network and is composed of Restricted Boltzmann Machines (RBMs) and uses a greedy layer by layer learning algorithm. However, the type of architecture used in this paper has a better approach since it provides discriminating powers for pattern classification by characterizing the posterior distributions of classes conditioned on the data. The following table contains definitions of the terms used here (Table 1).

3 Experimental Results and Outcome

The data set used in the experiment is the KDD Cup 1999 dataset which is a collection of simulated raw TCP dump data over an epoch of 9 weeks on a LAN. The training data has about 5 million connection records from seven weeks of network traffic and two weeks of testing data yielded around 2 million connection records. The training data have 22 of the total 29 attacks present in the test data. The known attack types are present in the training set while the novel attacks are additional attacks that are present in the test data set and not in the training data set. The attack types are grouped into 4 categories:

- DOS – Denial of Service (DoS) attack – e.g. syn flooding
- Probing – Surveillance and other probing – e.g. port scanning
- U2R – Unauthorized access to the root user privileges. e.g. Buffer overflow attacks
- R2L – Unauthorized access from a remote machine, e.g. password guessing.
- The training set has about 494,021 records from which 97,277 are normal, 391,458

are DOS attacks, 4107 are Probe, 1126 are R2L and 52 are U2R connections. Each connection has about 41 attributes describing different features of connection and a label assigned to each either as an attack type or normal. This data set was used originally in The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

Table 1. Basics terminology [11, 16, 17]

Terminology	Meaning
Deep Learning	It is a class of machine learning techniques, based on a set of algorithms that use multiple layers with complex structures composed of non-linear transformations to model high level data
Deep Belief Networks	It is a probabilistic generative model composed of multiple layers of stochastic, hidden variables. The top two layers have undirected, symmetric connections. The lower layers have direct connections from above and as such receive top-down
Boltzmann Machine	It is a network of neuron like units that are symmetrically connected. They are concerned with making stochastic decisions about whether to be on or off
Restricted Boltzmann Machine	It consists of a layer of visible units and a layer of hidden units with no visible-visible and hidden-hidden connections
Deep Boltzmann Machine	It is a special kind of BM where hidden neurons are arranged in a deep layered manner. There exist no visible-visible or hidden-hidden connections within the same layer. This involves a connection between only the adjacent layers
Deep Neural Network	It is a multilayer network with many hidden layers. The weights in these networks are fully connected and pre-trained
Deep Auto Encoder	It is a special kind of deep neural network where the output target is the input itself. Deep Belief Networks or distorted training data are used to train the network
Distributed Representation	It is the representation of the data in such a way that it appears to be generated by interaction of various hidden factors. They form a basis for deep learning

3.1 Simulation Results

The data set that was used had response values in column 42 with losses being set as Cross Entropy in order to get classification model (Table 2). The input data set has been divided into two parts - training frame and validation frame. 75% of the data set has been assigned as the training frame and 25% of the data set has been assigned as the validation frame. Upon running the algorithm, a scoring history in the form a graph was obtained as shown below. The graph produced is between training and validation frame as x axis and epochs as the y axis. It depicts the similarity between the training and validation frame and that the model that has been created is correct (Fig. 2).

Table 2. Model parameters

Parameter	Value	Description
Response column	C42	Response column
Hidden	200,200	Hidden layer sizes (e.g. 100,100)
Seed	7069314529076090000	Seed for random numbers (affects sampling) - Note: only reproducible when running single threaded
Loss	Cross Entropy	Loss Function

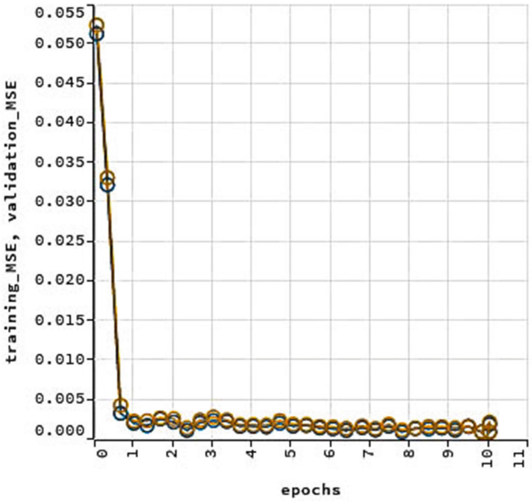

Fig. 2. Training and validation error of deep learning neural network

3.1.1 Experimental Outcome of Deep Neural Network

The activation functions used are rectifier activation function and softmax activation function (Table 3).

Table 3. Status of neurons

A	B	C	D	E	F	G	H	I	J	K	L
1	119	Input	0	0	-	-	-	-	-	-	-
2	200	Rectifier	0	0	0.6364	0.4589	0	0.0015	0.1133	0.4745	0.1081
3	200	Rectifier	0	0	0.6957	0.4432	0	0.0028	0.0984	0.9853	0.0676
4	23	Softmax	0	0	0.9427	0.2252	0	0.3050	0.4532	−0.2707	0.0619

A – Layer, B – Union, C – Type, D – L1, E – L2, F – Mean Rate, G – rate_RMS, H – Momentum, I – Mean Weight, J – Weight RMS, K – Mean Bias, L – Bias RMS.

The rectifier is an activation function defined as,

$$f(x) = \max(0, x) \tag{5}$$

Where x is the input.
It can also be expanded to include Gaussian noise given as,

$$f(x) = \max(0,\ x + N(0, \sigma(x))) \tag{6}$$

Softmax function is a generalization of logistic function that squashes a M-dimensional vector z of arbitrary real values to a M dimensional vector $\sigma(z)$ of real values in the range (0,1) that add up to 1. The function is given by,

$$P(y = j|x) = \frac{e^{x^T w_j}}{\sum e^{x^T w_k}} \tag{7}$$

3.1.2 Output - Training Metrics

This includes the output obtained from the training set. The following training metrics depict the efficacy of the implementation (Table 4).

Table 4. Output training metrics

Parameters	Values
Description	Metrics reported on temporary training frame with 9910 samples
Model_category	Multinomial
Scoring Time	1442054607700
MSE	0.000961
R^2	0.999944
Logloss	0.012146

The Mean Square Error is approximately 0.09%. The value of R^2 is 0.999944 which means that it is more than 99% similar to the expected result. Log loss function maps the variables to the real numbers which represent the cost associated. Hit Ratio is the number of times a correct prediction was made over total predictions. Top 10 hit ratios are used for the prediction and that has been given in the following Table 5.

3.1.3 Output - Validation Metrics

Output Validation metrics depict the output of the testing set. The following output metrics help in determining the efficacy of the model (Table 6).

Here as well, the MSE value is 0.09%. The R^2 value is more than 99%, which means the predicted value is 99% correct. The hit ratio is given in the following Table 7.

Table 5. Hit ratio for training set

K (Number of hits)	Hit ratio
1	0.9989
2	1.0
3	1.0
4	1.0
5	1.0
6	1.0
7	1.0
8	1.0
9	1.0
10	1.0

Table 6. Output validation metrics

Name of the parameter	Outcomes
Description	Metrics reported on full validation frame
Model_category	Mutinomial
MSE	0.000970
R^2	0.999944
Logos	0.011482

Table 7. Hit ratio for validation metrics

K	Hit ratio
1	0.9989
2	0.9997
3	0.9998
4	0.9998
5	0.9999
6	0.9999
7	0.9999
8	0.9999
9	0.9999
10	0.9999

4 Comparison with Support Vector Machine (SVM)

Support vector machines are supervised learning models that are used in machine learning that utilize learning algorithms to analyze and recognize patterns for classification [14]. It's training algorithm creates a model that assigns new examples into one category or the other and thus is a non-probabilistic binary linear classifier. It is a representation in terms of points in space such that there exists a clear gap in between various kinds of points grouped together. New data are predicted and classified based on how much it is closer to one particular group than the other.

4.1 Simulation Results for SVM

SV type: C-svc (classification)
Parameter: cost C = 5, Gaussian Radial Basis kernel function.
Hyperparameter: sigma = 0.05
Number of Support Vectors: 16860
Objective Function Value:
−2.0098 −11.4563 −31.787 −98.428 −50.5466 −1.999 −22.3287 −1.999 −1.7028 −1.8817 −1.9603 −1.9239 −1 −1.8357 −1 −8.426 −10.269 −9.3452 −1.7028 −7.5755 −1.7028 −24.0647 −19.539 −1.8817 −13.127 −1.8817 −33.4674 −1.9603 −18.2219 −1.9603 −1.924 −15.5029 −1.924 −1.8357 −1 −1.8357
Training error: 0.15365
Cross validation error: 0.00435

As we can see, the cross validation error is very low. Hence the model is accurate.

Comparison between the neural network and SVM can be tabulated as follows (Table 8):

Table 8. Comparison between deep neural network & SVM

Deep neural network	SVM
Error: 0.000961	Error: 0.15365
Accuracy: 0.999944	Accuracy: 0.84635

5 Conclusion

In this work, the training and validation models have a very high R^2 value. This high value has indicated that the adopted model is highly accurate. Application of the deep learning algorithm to the Intrusion detection System has enabled us to produce a detailed confusion matrix for the training set, as well as for the validation set. The result is supported along with a precise MSE graph. With the loss being set as Cross Entropy, we get a classification model that can be used to detect future intrusion attacks. The results obtained by Deep Neural Network are compared with the results obtained by Support Vector Machine.

References

1. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA 2001) (2001)
2. Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection using neural networks and support vector machines. In: Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN 2002, vol. 2, pp. 1702–1707. IEEE (2002)

3. Ouyang, W., Wang, X.: Joint deep learning for pedestrian detection. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 2056–2063. IEEE, December 2013
4. Roy, S.S., Mittal, D., Basu, A., Abraham, A.: Stock market forecasting using LASSO linear regression model. In: Abraham, A., Krömer, P., Snasel, V. (eds.) Afro-European Conference for Industrial Advancement. AISC, vol. 334, pp. 371–381. Springer, Cham (2015). doi:10.1007/978-3-319-13572-4_31
5. Basu, A., Roy, S.S., Abraham, A.: A novel diagnostic approach based on support vector machine with linear kernel for classifying the Erythemato-Squamous disease. In: 2015 International Conference on Computing, Communication Control and Automation (ICCUBEA), pp. 343–347. IEEE, February 2015
6. Arel, I., Rose, D., Coop, R.: DeSTIN: a scalable deep learning architecture with application to high-dimensional robust pattern recognition. In: AAAI Fall Symposium: Biologically Inspired Cognitive Architectures, November 2009
7. Roy, S.S., Viswanatham, V.M., Krishna, P.V., Saraf, N., Gupta, A., Mishra, R.: Applicability of rough set technique for data investigation and optimization of intrusion detection system. In: Singh, K., Awasthi, A.K. (eds.) QSHINE 2013. LNICST, vol. 115, pp. 479–484. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37949-9_42
8. Wang, S., Jiang, Y., Chung, F.L., Qian, P.: Feedforward kernel neural networks, generalized least learning machine, and its deep learning with application to image classification. Appl. Soft Comput. **37**, 125–141 (2015)
9. Wackerly, D., Mendenhall, W., Scheaffer, R.: Mathematical Statistics with Applications. Cengage Learning (2007)
10. Draper, N.R., Smith, H., Pownell, E.: Applied Regression Analysis, vol. 3. Wiley, New York (1966)
11. Deng, L.: Three classes of deep learning architectures and their applications: a tutorial survey. APSIPA Trans. Sig. Inf. Process. (2012)
12. Roy, S.S., Viswanatham, V.M.: Classifying spam emails using artificial intelligent techniques. Int. J. Eng. Res. Africa **22**, 152–161 (2016)
13. Roy, S.S., Viswanatham, V.M., Krishna, P.V.: Spam detection using hybrid model of rough set and decorate ensemble. Int. J. Comput. Syst. Eng. **2**(3), 139–147 (2016)
14. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
15. Mittal, D., Gaurav, D., Roy, S.S.: An effective hybridized classifier for breast cancer diagnosis. In: 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), pp. 1026–1031. IEEE, July 2015
16. Bengio, Y.: Learning deep architectures for AI. Found. Trends® Mach. Learn. **2**(1), 1–127 (2009)
17. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
18. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Trans. Patt. Anal. Mach. Intell. **12**, 993–1001 (1990)