I. J. Computer Network and Information Security, 2019, 3, 8-14
Published Online March 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2019.03.02

Deep Learning Approach on Network Intrusion Detection System using NSL-KDD Dataset

Sandeep Gurung
Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majitar, Sikkim, India
E-mail: sandeep.gu@smit.smu.edu.in

Mirnal Kanti Ghose
Sikkim University, Sikkim, India
E-mail: mkghose@cus.ac.in

Aroj Subedi
Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majitar, Sikkim, India
E-mail: arojsubedi@gmail.com

Received: 30 November 2018; Accepted: 18 January 2019; Published: 08 March 2019

Abstract—The network infrastructure of any organization is always under constant threat to a variety of attacks; namely, break-ins, security breach or system misuse. The Network Intrusion Detection System (NIDS) employed in a network detects such penetration attacks and intrusions within a network. Known classes of attacks can be detected easily by performing pattern matching while the unknown attacks are harder to detect. An attempt has been made to design a system using a deep learning approach for intrusion detection that not only learns but also adjusts itself to the patterns not defined earlier. Sparse auto-encoder has been used for unsupervised feature learning. Logistic classifier is then utilized for classification on NSL-KDD dataset. The performance of the system has been measured with respect to accuracy, precision and recall and the results have been found to be very promising for future use and modifications.

Index Terms—NIDS, deep learning, Sparse auto-encoder, logistic classifier, NSL-KDD.

I. Introduction

The network architecture is always vulnerable to various types of security breaches, attempted break-ins, penetration attacks and other similar intrusions by unauthorized and malicious users. The network being a repository aims at sharing resources between authorized users, also attracts unwanted users who are interested in exploiting them. In addition, formulations of global protection policies are rare and difficult to implement. The security breach or intrusion is a critical issue for any organization. It is thus important to develop precautionary measures to safeguard the interest of the organization from various categories of attacks to which it is susceptible to.

As defined by Heady et al. [7], "an intrusion is a set of actions that attempt to compromise the integrity, confidentiality or availability of information resources."

The system employed to detect such malicious actions in a network is termed as a Network Intrusion Detection System (NIDS). It should be able to detect a wide range of attacks and security violations inflicted by outsiders. The system should also be able to check on any activity of malpractices and abuses practiced by the insiders. Intruders can broadly be classified into three different categories. Masqueraders are typically outsiders who are not authorized users but penetrate the system using legitimate user accounts. A Misfeasor is an insider, a legitimate user who misuses the privileges given and accesses resources that they are not authorized to. A Clandestine can be either an insider or an outsider who tries to gain supervisory access to the system [1].

The NIDS are of two categories namely; Signature-based Network Intrusion Detection System (SNIDS) and Anomaly detection based Network Intrusion Detection System (ADNIDS). SNIDS raises an alarm for intrusion by performing a pattern matching on the features of the information it is aware of. ADNIDS on the other hand, raises an alarm for intrusion if there are any significant deviations of the user activity under analysis from the normal traffic pattern. SNIDS, therefore, has a higher detection rate for the known types of attacks, while ADNIDS performs better in case of novel/unknown patterns of attacks. However, due to the variations in the behavior of the intruder, an ADNIDS has a tendency of generating high false alarms. The security violations can be detected by monitoring the system audit record for any abnormal pattern of system usage [2].

Different kinds of machine learning techniques have been employed to develop a Network Intrusion Detection System for anomaly detection [9]. The NIDS model designed can be trained and tested for performance using NSL-KDD dataset [5], which is a significant upgrade of

the KDD Cup 99 dataset [4]. Different machine learning techniques perform differently based on the input features, the training and the test datasets selected [3]. Similar types of approaches, learning techniques, and input features do not always guarantee the same results for a variety of different classes of possible unknown attacks. Deep learning techniques are popular as they facilitate the design of robust and efficient NIDS. A deep learning approach based on Self-taught Learning (STL) [6] and a Non-symmetric Deep Auto-Encoder (NDAE) [8] have been found to be useful for unsupervised feature learning of unlabeled data to understand the intrinsic behavioral patterns of intruders. A classification of the patterns can be performed using suitable classifiers like soft-max regression. In the proposed work a deep learning approach based on sparse auto-encoders is used to learn the nature of the patterns and a logistic regression classifier is used to classify the behavior of users learned through the stacked encoders. The related work is discussed in Section II. The proposed work is given in Section III followed by its design in Section IV. The experimental results and discussion are given in Section V. The conclusion and future work are given in Section VI.

II. Related Works

Most of the works carried out for Intrusion detection predictive modeling part is performed using similar types of datasets for training and testing. It is difficult to generalize the real-time events through these datasets. The performance measure of the majority of these predictive models thus decreases when thrown into real network traffic.

Several approaches have been proposed for the classification of normal connections with anomalies to detect intrusions in a network. Shyu et al. [9] proposed a novel scheme using Principal Component Analysis (PCA) treating anomalies as outliers. The anomaly detection scheme performed better with the KDD'99 dataset. The detection rate rose to 99% while the false alarm rate dropped to as low as 1%. Revathi, et al. [3] performed a detailed analysis on the NSL-KDD dataset using only relevant features both with and without feature reduction of the dataset on different classification algorithms like J48 decision tree, Random Forest, Support Vector Machine, Naive Bayes algorithm, etc. Random Forest achieved the highest test accuracy in both the cases.

Deep learning techniques facilitate the design of flexible and robust NIDS. Khaled et al. [10] proposed a deep learning approach for intrusion detection using one hidden layer of Restricted Boltzmann Machine for unsupervised feature reduction and Logistic regression with multi-class soft-max for classification. The model was tested on the total 10% KDD-Cup'99 test dataset and a detection rate of 97.9% was achieved. The KDD-Cup'99 dataset doesn't remotely impose reasonable challenge as that of real network traffic. Niyaz et al. [6] proposed a Self-Taught Learning (STL), a deep learning technique for unsupervised feature learning and soft-max regression for classification. The model was evaluated for 2 class, 5 class and 23 class classification against the benchmark NSL-KDD dataset and the results obtained were encouraging and the model showed better performance. Yin et al. [11] proposed a deep learning approach for intrusion detection using Recurrent Neural Networks (RNN). The experimental results showed that the performance of the model was promising in both binary and multiclass classification and the model was able to classify with high accuracy. Shone et al. [8] proposed a novel deep learning classification model constructed using stacked Non-Symmetric Deep Auto-Encoder (NDAE) for unsupervised feature learning and RF classification algorithm for classification. The model was implemented in Tensor Flow using benchmark KDD Cup'99 and NSL-KDD datasets. The model achieved a consistent level of classification accuracy with the reduction in training time and a high level of precision and recall.

The problem with building a robust NIDS is unavailability of real-time pattern of network data consisting of both intrusions and normal uses, constantly evolving and changing attack patterns, long training time and insufficient knowledge about modifications required in datasets. A model may achieve high accuracy against the test datasets but the accuracy always seems to degrade while analyzing the real network traffic. The NIDS implemented using deep learning somewhat broke this trend. Most of works found in the literature implementing deep learning had significant detection rate and could somewhat detect anomalies not known earlier. Majority of the work is yet to be done in the intrusion detection field to build an applicable and efficient NIDS but this certainly seem to be the way forward.

III. Proposed Work

The proposed work aims at using a deep-learning based approach for network intrusion detection. The system uses a deep network to train itself with the patterns of anomalies and classify the network traffic between the normal connections and the intrusions. The approach is also focused at reducing the false alarm rate to a minimum value. The approach has the flexibility to adjust to new patterns of intrusions and the behavior of the person that might change during the course.

The proposed system implements a deep network system (sparse auto-encoder with logistic regression), trained by the NSL-KDD dataset. It gives an output value of 0 or 1, where 1 denotes an intruder and 0 corresponds to a normal user.

The system utilizes a total of 115 features as an input to the system some of which are; protocol used, source address, destination address, the time-stamp, services, flag, number of failed logins, number of logins. Each feature is given as an input to the neurons. A sparse auto-encoder with sparsity constraint is utilized for training and learning new features from the data set. A deep network is created by stacking the auto-encoders and the classification from the features learned is implemented

using logistic regression network. Logistic regression is taken as the output involves the identification of two classes of users.

IV. Design

Pre-processing of the dataset is done before being applied to the network. The non-numeric parameters are replaced with numeric values and the data set is normalized using max-min operation. The overall flow of the proposed system is given below in Fig 1. The KDD-Cup Dataset, a modification of the NSL-KDD dataset includes 41 features derived from TCP/IP connections, traffic features accumulated in window interval and content features extracted from the application layer data of connections. Out of the 41 features, 34 are continuous, 4 are binary and 3 are symbolic (protocol_type, service, flags).

An auto-encoder is an artificial neural network used for unsupervised learning capable of adapting to understanding new features from a set of input data. The input layer represents the original sets of features; the hidden layer facilitates the better understanding of the new features with reduced dimension helps. The output layer represents the target feature which is the same as that of the input source. Sparse auto-encoders with a sparsity constraint allow the network for a clear exploration of the effects of sparseness for a given dataset thus helping in finding new pattern distribution of the input data.

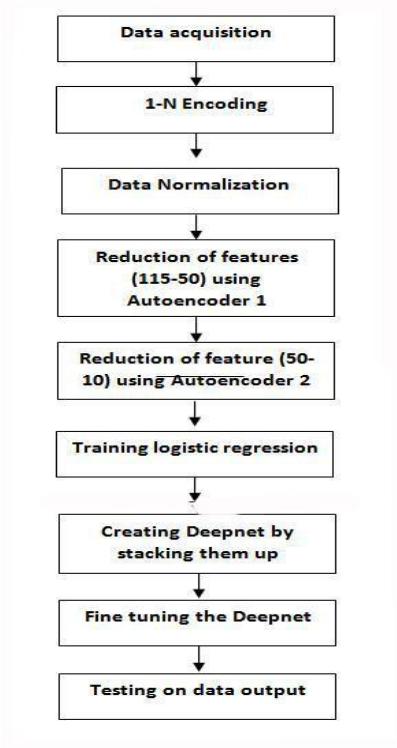

Fig.1. Design Flowchart of the Proposed System

The auto-encoder uses a stochastic conjugate gradient for error minimization with the sigmoid function as the activation function. While training a sparse auto-encoder, the process of optimizing the loss function involves the sum of three terms: Mean Squared Error term, L2 Regularization term and the Sparsity Regularization term.

The loss function used for training a sparse auto-encoder is given in equation 1.

$$E = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}(x_{kn} - \hat{x}_{kn})^2 + \lambda * \Omega_{weights} + \beta * \Omega_{sparsity} \quad (1)$$

Where,

λ = coefficient for the L2 regularization term and
β = coefficient for the Sparsity regularization term

Different parameters used for training a sparse auto-encoder are discussed below:

- Regularization: Protection against over-fitting problems. The parameter value set to 0 implies no protection against over fitting has been applied..
- L2 Weight Regularization: It quantifies the complexity of the model as the sum of squares of all feature weights. It takes a positive scalar value to control the impact of regularization used in the loss function (also termed as weight decay).
- Sparsity: It reduces the dependency between the feature vectors thus, allowing the users to increase the number of features. It is set close to 0 to provide average activation on the hidden layer.
- Sparsity Regularization: It is positive scalar value to control the impact of Sparsity Regularization term in the loss function.

The first level of the sparse auto-encoder reduces the 115 feature set to 50 as shown in Fig 2. In the diagram, $X_{i(1\text{-}115)}$ represent the input nodes, $h_{i(1\text{-}50)}$ represents the hidden layer nodes and $X_{i(1\text{-}115)}$ represents the output layer nodes.

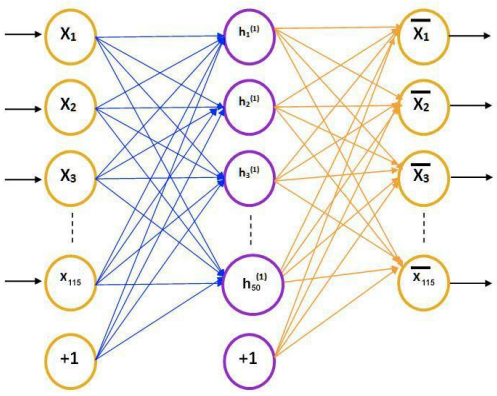

Fig.2. Level 1 Auto-Encoder (Reduction of 115 to 50 Features)

The level 2 sparse auto-encoder further reduces the 50

learned features to 10 new features which are then given as inputs to logistic regression. In the diagram as shown in Fig 3, h_{1i}(1-50) represents input nodes, h_{2i}(1-10) represents the hidden layer nodes and h_{1i}(1-50) represents the output layer nodes.

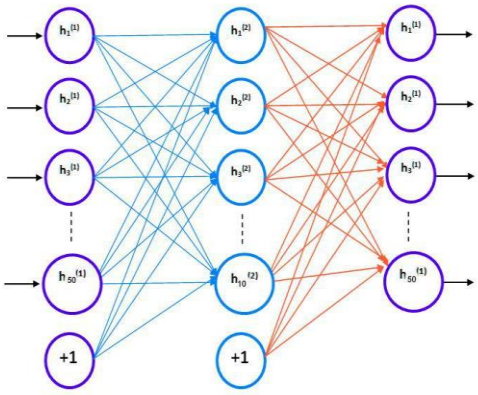

Fig.3. Level 2 Auto-encoder (Reduction of 50 to 10 Features)

The new features learned from level 2 auto-encoder are fed into the logistic classifier which identifies whether a user is normal (0) or an intruder (1) as given in Fig 4. Logistic Regression uses a sigmoid or logistic function as its activation function giving the probability measure of the output in the range of [0,1].

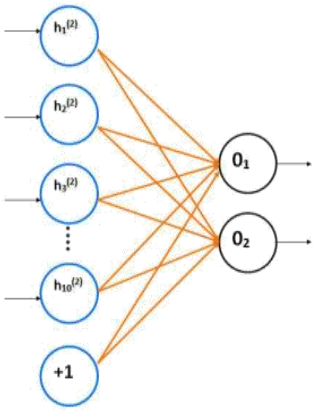

Fig.4. Logistic Classifier (Classifies 10 Inputs to two Outputs)

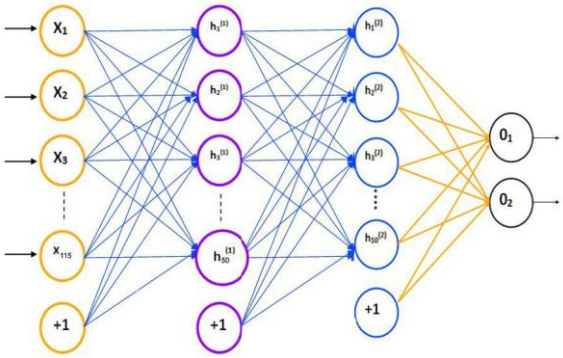

Fig.5. Fully Connected Layer (Input: 115, Hidden 1:50 Hidden 2:10 Output: 2)

The final stack implements a fully connected network consisting of 1 input layer, 1 output layer, and 2 Hidden Units as shown in Fig 5. The 115 inputs from the original dataset are compressed and reduce to 50 nodes in the second layer and to 10 nodes in the third layer. The final output layer classifies whether a user is normal or not.

V. Experimental Results

A total of 22,545 data with 41 features was taken from the NSL-KDD dataset for training. The 3 symbolic features (protocol, service, flag) were expanded using 1-N encoding. The encoded data contains 115 features (3 from protocol, 64 from service and 11 from flag). The protocol_type has 64 variations namely; FTP, HTTP, login, etc which indicates the protocol used. The service type describes the ICMP, TCP, and UDP services. The flags REJ, SF, S0, S1, etc denote the priority of the data. The num_access_files is ignored as it stays 0 throughout the dataset. The NSL-KDD dataset is normalized with a max-min operation.

The confusion matrix is utilized to measure the performance of the model. It takes the following parameters into consideration. True Positive (TP) represents the correct classification of the Intruder. A False Positive (FP) is the incorrect classification of a normal user taken as an intruder. The True Negative (NP) represents a normal user classified correctly, where as a False Negative (FN) is an instance where the intruder is incorrectly classified as a normal user.

The accuracy is the ratio of the correctly predicted values to the total number of test cases.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

The precision is the ratio of the correctly predicted positive values to the total predicted positive values.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

The recall measures the proportion of the positive values that are correctly classified.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

The specificity measures the proportion of the negative values that are correctly classified.

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (5)$$

A. Sparse Auto-encoder 1:

The parameters associated with auto-encoder ‘msesparse’ at level 1, as shown in Fig 6 are:

1. Regularization=0,
2. L2WeightRegularization=0.001
3. Sparsity Regularization= 4,
4. Sparsity=0.2

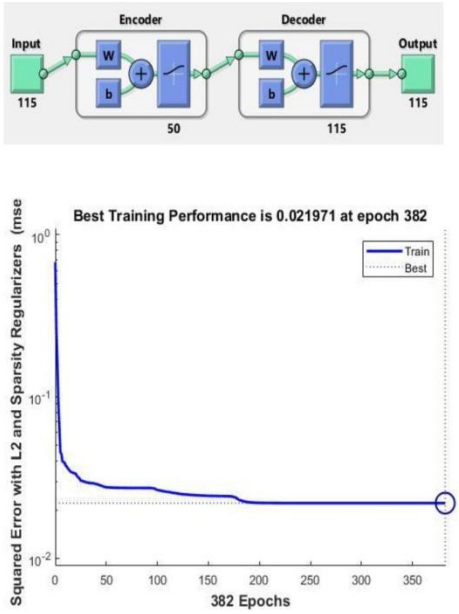

Fig.6. Sparse Auto-Encoder 1 (Output of View (Network 1)) and Performance Plot for Sparse Auto-encoder 1

For fine-tuning, the best performance validation for auto-encoder 1 of 0.021971 was evaluated at 382 epochs. The epoch specifies the number of times the training sample is used to update the weights. Fine tuning adjusts the parameters of a trained model precisely.

B. Sparse Auto-encoder 2:

The parameters associated with auto-encoder 'msesparse' at level 2, as shown in Fig 7 are:

1. Regularization = 0,
2. L2WeightRegularization = 0.001
3. Sparsity Regularization = 1,
4. Sparsity = 0.05

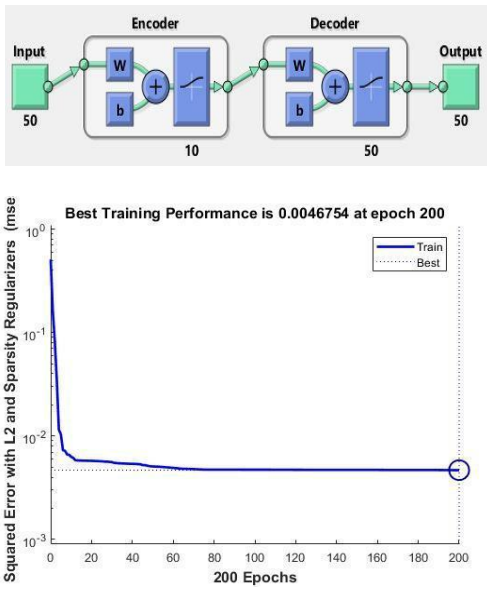

Fig.7. Sparse Auto-Encoder 2 (Output of View (Network 2)) and Performance Plot for Sparse Auto-encoder 2

The best performance validation for sparse auto-encoder at level 2 of 0.0046754 was evaluated at 200 epochs.

C. Logistic Classifier:

The logistic classifier takes the output of the level 2 encoders and classifies them to 2 feature classes as shown in Fig 8&9.

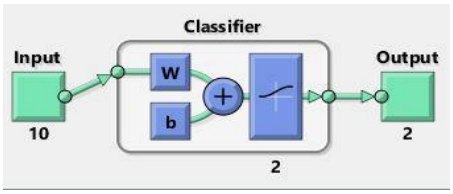

Fig.8. Logistic Classifier (Output of View (Network3))

D. Fully Connected Layer

The fully connected network of the stack of all the networks (auto-encoder 1, auto-encoder 2, logistic regression) is constructed as shown in Fig 10. The dataset is taken as an input and classified into 2 outputs (normal = 0; intruder = 1). The internal weights are derived from previous auto-encoders and logistic classifier.

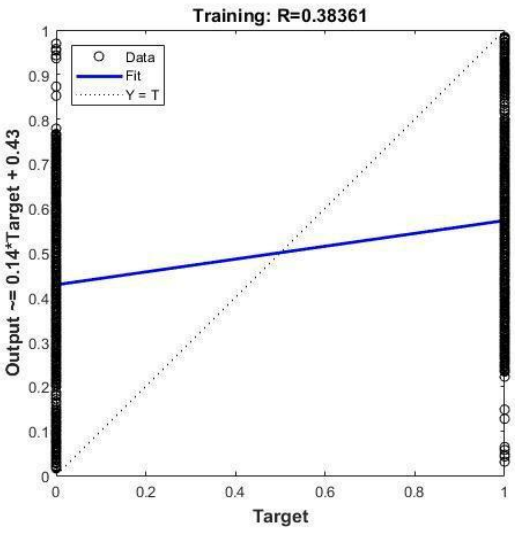

Fig.9. Regression plot for Logistic Classifier.

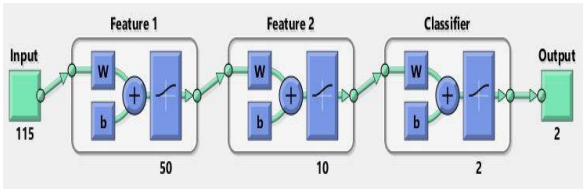

Fig.10. Fully Connected Layer (Deep-net)

The network was tested with 2401 sample inputs and the confusion matrix formed is shown below in Fig 11.

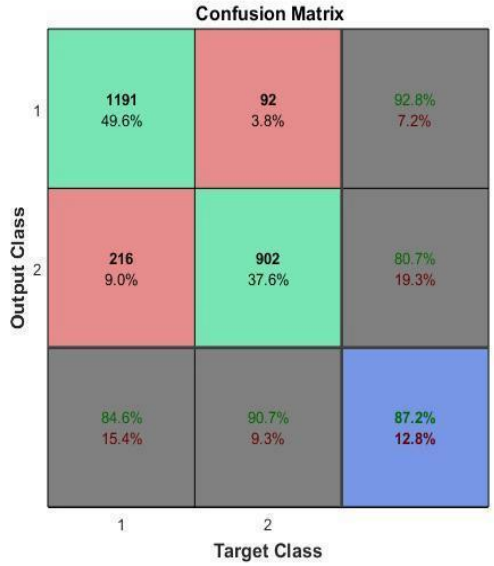

Fig.11. Confusion Matrix

From a dataset of 1283 anomalies, 1191 were identified successfully as True Positives and the remaining 92 were recognized incorrectly as False Negatives. From 1118 normal patterns as an input, 216 were identified as intrusions as False Positives whereas 902 were classified as normal as True Negatives.

The performance of the system was calculated using Equations (2) to (5). The precision score obtained for the model was 84.6% and the recall score was 92.8% whereas the specificity and the negative predictive values were 80.7% and 90.7% respectively. The overall accuracy of the model was 87.2%.

VI. Conclusion And Future Work

A deep learning based approach for Network Intrusion Detection System is an anomaly based technique used to detect any possible intrusion of any type in the network. The system gives a higher accuracy rate in comparison to Signature-Based Intrusion Detection approaches and also reduces the chances of False Positives and Negatives. The network learns and adjusts itself to patterns which were not defined previously. The system can be implemented on any server which monitors the network activity of any organization in real time. The deep-net can identify any intrusion and adjust itself with the newer data to classify an intruder. A variation on the encoders can be carried out to make the system more robust and also increase the accuracy of detection to a higher degree.

References

[1] Behrouz A Forouzan, Debdeep Mukhopadhyay (2017), Cryptography and Network Security, (Third Edition), McGraw Hill Education (India) Private Limited.

[2] D. E. Denning, "An Intrusion Detection Model," IEEE Transactions on Software Engineering, SE-13, pp. 222-232, 1987.

[3] S. Revathi, Dr. A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278- 0181 Vol. 2 Issue 12, December - 2013.

[4] KDD Cup 1999 Data, Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, August 2003.

[5] "Nsl-kdd data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/KDD/NSL-KDD.html, March 2009.

[6] Q. Niyaz, W. Sun, AY Javaid, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System" in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies, ser. BICT'15. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 21–26. [Online]. Available: http://dx.doi.org/10.4108/eai. 3-12-2015.2262516.

[7] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "The architecture of a network level intrusion detection system", Technical report, Computer Science Department, University of New Mexico, August 1990.

[8] N. Shone, T. N. Ngoc, Vu D. Phai, "A Deep Learning Approach to Network Intrusion Detection", IEEE Transactions on Emerging Topics in Computational Intelligence (Volume: 2, Issue: 1) Feb. 2018.

[9] Shyu, M-L., S-C. Chen, K. Sarinnapakorn, and LW. Chang, "A Novel Anomaly Detection Scheme Based on Principal Component Classifier", ICDM Foundation and New Direction of Data Mining workshop, 03-1221.1-2312, 2003.

[10] K. Alrawashdeh and C. Purdy, "Toward an Online Anomaly Intrusion Detection System Based on Deep Learning," in 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). Anaheim, California, USA: IEEE, dec 2016, pp. 195–200.

[11] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, Xinzheng He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks".

Authors' Profiles

Sandeep Gurung received his M. Tech and Ph.D degree Sikkim Manipal University, India in the year 2009 and 2017. He is working as Assistant Professor (Selection Grade) at Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Sikkim.

His research interests include Visual Cryptography and Steganography, Soft Computing and Distributed System.

Prof.(Dr.) Mrinal Kanti Ghose, is currently working as a visiting faculty at Sikkim University, Sikkim, India. He was a former Dean (Academics), Professor & HOD at CSE Department Sikkim Manipal Institute of Technology, Majhitar, Sikkim. Prior to his academic career he was formerly a Sr. Scientist at Vikram Sarabhai Space Centre & RRSSC (E), ISRO, India.

How to cite this paper: Sandeep Gurung, Mirnal Kanti Ghose, Aroj Subedi,"Deep Learning Approach on Network Intrusion Detection System using NSL-KDD Dataset", International Journal of Computer Network and Information Security(IJCNIS), Vol.11, No.3, pp.8-14, 2019.DOI: 10.5815/ijcnis.2019.03.02

