Smart Intelligent Computing and Communication Technology
V.D. Ambeth Kumar et al. (Eds.)

doi:10.3233/APC210005

An Effective Comparative Analysis of Data Preprocessing Techniques in Network Intrusion Detection System Using Deep Neural Networks

Aravind Prakash M[a,1], Indra Gandhi K[b], Sriram R[b] and Amaysingh[b]
[a1,b]*Dept of Information Science and Technology, College of Engineering, Chennai, India*

Abstract. Recently machine learning algorithms are utilized for identifying network threats. Threats otherwise called as intrusions, will harm the network in a stern manner, thus it must be dealt cautiously. In the proposed research work, a deep learning model has been applied to recognize and categorize unanticipated and unpredictable cyber-attacks. The UNSW NB-15 dataset has a vital number of features which will be learned by the hidden layers present in the suggested model and classified by the output layer. The suitable quantity of layers, neurons in each layer and the optimizer utilized in the proposed work are obtained through a sequence of trial and error experiments. The concluding model acquired can be utilized for estimating future malicious attacks. There are several data preprocessing techniques available at our disposal. We used two types of techniques in our experiment: 1) Log transformation, MinMaxScaling and factorize technique; and 2) Z-score encoding and dummy encoding technique. In general, the selection of data preprocessing techniques has a direct impact on the output performed by any machine learning process and our research, attempts to prove this concept.

Keywords. Intrusion detection System (IDS), Attacks, Deep Neural Networks (DNN), Host-based intrusion detection system (HBID), Machine Learning.

1. Introduction

Deep Neural Networks (DNN), comes under Machine Learning that enables machines to study samples and predict using the learnt features. The process of feature engineering is not needed in DNN as the features are implicitly learned by the hidden layers which leads to efficiency. An attack is defined as the stealing of information or causing damage to the user's system without the consent of the user. Currently, users are freely sharing sensitive data over the networks without any security awareness and hence it is very essential to protect such data. In this paper, we made a systematic

[1]Aravind Prakash M, Dept of Information Science and Technology College of Engineering, Chennai, India.
E-mail: aravind.2k.135@gmail.com

to cyber-attacks. There are varieties of the attacks in any dataset which includes Fuzzers, Exploits, Generic etc., that can not only damage servers but also make use of the sensitive information of the users [1]. Thus there is a need of an efficient and useful Intrusion Detection System. For creating that, a novel preprocessing technique must be used so that data are properly learned by the system. This way, the prediction accuracy increases. In this paper, Section 2 reviews related recent literature and Section 3 briefly describes the problem description and the proposed method with a simple architecture diagram; Section 4 gives the implementation details of the suggested research work have been narrated; Section 5 briefs the result analysis and its related discussions; and finally Section 6 concludes the research and recommends the direction for future work.

2. Literature Review

Intrusion detection system is very essential to ensure information security [2] and the major challenge is to correctly identify different attacks in the network. The process of identifying different types of attacks and accurately classifying the malicious network traffic are posing a great challenge [2]. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), are two main types of DNN architectures that are widely explored to enhance the performance of intrusion detection system [2]. Vinayakumar et al. [3, 1] recommended a combination of NIDS and HIDS (Host based Intrusion Detection System) [1]. Binary and multi-class classification are performed on several Intrusion datasets including the UNSW NB-15. Jing et al. [4] have used the same UNSW NB15 [1] dataset, but the machine learning algorithm applied was a Support Vector Machine]. Zhang et al. [5] proposed a NIDS by combining the Improved Principal Component Analysis (IPCA) and Gaussian Naïve Bayes (GNB) and achieved a desirable good accuracy. Zegeye et al. [6] recommended the ideas of intrusion detection system with Hidden Markov Model. The curse of dimensionality has been fixed by this approach i.e., the errors that happen while applying HMM (Hidden Markov Model) to IDS. Asheret et al. [7] put forth a statement that knowledge plays a very essential role in classifying events. The study investigated how knowledge in network operations and information security influenced the detection of intrusions in a simple network [8]. Zhang et al. [9] proposed an intrusion detection system with an algorithm called Synthetic Minority Oversampling Technique combined with Edited Nearest Neighbors (SMOTE-ENN) for balancing network. Zhang et al. [10] proposed an auto encoder-based method for the NSL-KDD dataset which compresses the less important features and extract key features without decoder. Samrin and Vasumathi [11] made investigations on the KDDCup 99 dataset about different techniques and intrusion classifications on the dataset. Meftah et al. [12] proposed a two-stage anomaly-based network intrusion detection process using the UNSW NB-15 dataset and achieved accuracy up to 74%.

3. Proposed Work

The overall system architecture is illustrated in Fig.1. In Data preprocessing of UNSW-NB15 dataset, the total number of instances used in the experiment is 2,57,673 out of

which in the first method of data preprocessing where the numerical features are preprocessed by log transformation and then scaled to similar scale by MinMaxScaler. In the second method of data preprocessing, z-score encoding, and dummy encoding is performed for numerical and categorical features respectively. In both the methods, the columns 'id' and 'attack_cat' are dropped; the column 'label' contains 0 for normal and 1 for attack which will be used as the dependent variable for classification. For multi-class classification, the 'attack_cat' column can be used by dummy encoding. In Hierarchical Feature Representation module, the features are analyzed and categorized. The variables or features that are categorical in UNSW NB-15 dataset are service, state and proto. All the other features are numeric variables. In Experimenting the dataset with different DNN layers module, the deep learning model must possess definite number of layers, definite number of neurons in each layer, and the appropriate activation functions for the best accuracy and these are found out through a number of experiments. The activation functions used include ReLu, Sigmoid, Softmax [1]. In Training the finalized DNN model module, test-train split is done and the model is trained using the training data. In the first method, it is found that ReLu in the input layer and sigmoid in hidden and output layer performed well. In the second method, the model in which the input layer and hidden layer containsReLu and the output layer contains softmax performed well. The details about the number of layers and neurons are given in Table 1, 2 and 3. In Loss Function Optimization module, for the first method, binary cross-entropy loss function is used since the label contains binary values. In the second method, categorical cross-entropy is used since the label is preprocessed using dummy encoding. In Intrusion Detection module, after test train split and training the model, the test data can be utilized for prediction. Confusion matrix is constructed and evaluation metrics like accuracy, precision, recall etc., are calculated and tabulated.

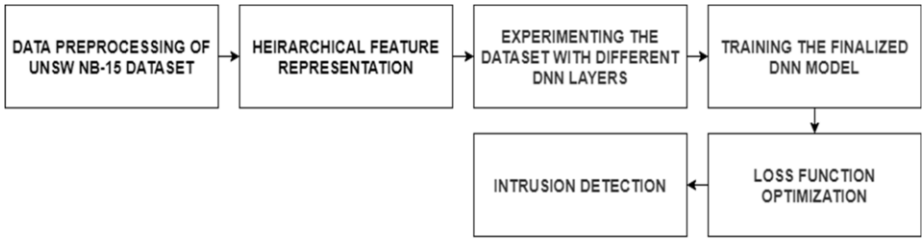

Figure 1. The overall system architecture.

4. Implementation

The experiment is done on an Intel Core i5 8th generation processor machine with 8GB RAM @ 2.30GHz. The IDE used are Spyder and JupyterNoteBook which are installed under the Anaconda Python 3.7 environment. As the dataset has huge number of instances and in order to speed up the computation process, Tensorflow has been used as the backend in JupyterNoteBook. Deep learning model has been deployed using the Sequential, Dense functions from keras. Several vital python libraries namely Numpy, Pandas, Scikit-learn are utilized for the effective processing of data.

5. Results and Discussion

From the results tabulated in Table 1, 2 and 3, the second method in which the data are preprocessed using Z-score encoding and dummy encoding yields better accuracy [1] and have lower false alarms than the earlier suggested method in [3]. The time taken to train the model is also reasonably lower than [3, 1] since there are a smaller number of neurons in each layer than in [3] and thus it is considered as an effective one.True Positive -True Normal (TN); True Negative -True Attack (TA); False Positive -False Attack (FA); False Negative -False Normal (FN). The best accuracy acquired are organized in Tables 1, 2 and 3. All the accuracy related metrics are calculated by the creation of a confusion matrix which is imported from *sklearn.metrics*package[1]. The following are the various basic standard evaluation metrics to rule out the best model in this proposed work and the calculations are shown in Table 1 and 2. Fig 2 depicts the comparative accuracy obtained by the following both the methods.

$$\text{Accuracy} = \frac{TN + TA}{TN + TA + FN + FA} \quad (1) \qquad \text{False Positive Rate} = \frac{FA}{FA + TA} \quad (2)$$

$$\text{False Negative Rate} = \frac{FN}{FN + TN} \quad (3) \qquad \text{Precision} = \frac{TN}{TN + FA} \quad (4)$$

$$\text{Recall} = \frac{TN}{TN + FN} \quad (5) \qquad \text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Pre}\,cision + \text{Re}\,call} \quad (6)$$

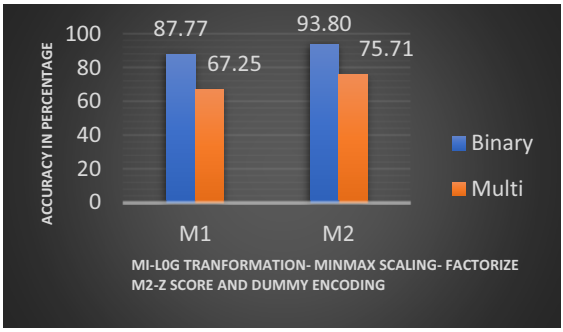

Figure 2. Combined analysis of M1 & M2 Preprocessing Techniques. M1- Log Transformation, MinMaxScaling and Factorize M2- Z-score encoding and dummy encoding

Table 1. Implementation Results for log transformation, MinMaxScaling and factorize method of data preprocessing (M1) – Binary classification

No. of layers	No of neurons	FPR	FNR	Accuracy	Precision	Recall	F1
3	43, 23, 1	0.1263	0.1984	0.8391	0.8533	0.8015	0.8265
4	43, 23, 11, 1	0.1017	0.2283	0.8325	0.8911	0.7716	0.8270
5	43, 23, 11, 5, 1	0.1214	0.2195	0.8297	0.8642	0.7804	0.8201
6	43, 23, 11, 5, 2, 1	0.2417	0.1036	0.8020	0.6326	0.8963	0.7417
6	**256, 128, 64, 32, 16, 1**	**0.1239**	**0.1200**	**0.8777**	**0.8428**	**0.8799**	**0.8609**

Table 2. Implementation results for z-score and dummy encoding method [13] of data preprocessing – Binary classification

No. of layers	No of neurons	FPR	FNR	Accuracy	Precision	Recall	F1
3	**43,23, 2**	**0.0414**	**0.0968**	**0.9380**	**0.9278**	**0.9031**	**0.9152**
4	43,23,11,2	0.0453	0.0961	0.9359	0.9206	0.9038	0.9121
5	43,23,11,5, 2	0.0491	0.0881	0.9367	0.9130	0.9118	0.9123
6	43,23,11,5, 2, 2	0.0538	0.0808	0.9366	0.9039	0.9191	0.9114
6	256, 128, 64, 32, 16, 2	0.0502	0.0830	0.9379	0.9107	0.9169	0.9137

Table 3. Implementation Results for M1- log transformation, MinMaxScaling and factorize method [13] of data preprocessing and M2 – z-score and dummy encoding method of data preprocessing – Multi-class classification

No. of layers	No of neurons	Accuracy – M1	Accuracy – M2
3	200, 100, 10	**0.6725**	0.7531
4	200, 100, 100, 10	0.6662	**0.7571**
5	200, 100, 100, 100, 10	0.6668	0.7528
6	200, 100, 100, 100, 100, 10	0.6532	0.7443

6. Conclusion and Future Work

In this proposed research, the attempt is to tune the dataset with various effective data preprocessing techniques to obtain the desired accuracy. It is experimentally proved that preprocessing techniques do have major impacts on any datasets used for any machine learning process. In future, the proposed "Network Intrusion Detection System" paves a way to develop efficient and Intelligent Intrusion Detection systems by the way of incorporating several other prominent machine learning algorithms.

References

[1] Aravind Prakash M, Sriram R, Amay Singh. Intelligent Intrusion Detection Systems using Deep Neural Networks (Thesis Report DOT 2020-11). Department of Information Science and Technology, College of Engineering, Guindy, Chennai, Tamil Nadu, India.

[2] Cui, Jianjing& Long, Jun & Min, Erxue& Liu, Qiang& Li, Qian. (2018). Comparative Study of CNN and RNN for Deep Learning Based Intrusion Detection System: 4th International Conference, ICCCS 2018, Haikou, China, June 8-10, 2018, Revised Selected Papers, Part V. 10.1007/978-3-030-00018-9_15.

[3] R, Vinayakumar&Alazab, Mamoun&Kp, Soman&Poornachandran, Prabaharan& Al-Nemrat, A. &Venkatraman, Sitalakshmi. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2895334.

[4] Dishan, Jing & Chen, Hai-Bao. (2019). SVM Based Network Intrusion Detection for the UNSW-NB15 Dataset. 1-4. 10.1109/ASICON47005.2019.8983598.

[5] Zhang, Bing & Liu, Zhiyang&Jia, Yanguo&Ren, Jiadong& Zhao, Xiaolin. (2018). Network Intrusion Detection Method Based on PCA and Bayes Algorithm. Security and Communication Networks. 2018. 1-11. 10.1155/2018/1914980.

[6] Zegeye, Wondimu& Dean, Richard &Moazzami, Farzad. (2018). Multi-Layer Hidden Markov Model Based Intrusion Detection System. Machine Learning and Knowledge Extraction. 1. 265-286. 10.3390/make1010017.

[7] Ben-Asher, Noam & Gonzalez, Cleotilde. (2015). Effects of cyber security knowledge on attack detection. Computers in Human Behavior. 48. 51-61. 10.1016/j.chb.2015.01.039.

[8] Gandhi, K &Janarthanan, S &Sathish, R &Surendar, A. (2020). Dominant Feature Prediction By Improved Structural Similarity Computation. 1-5. 10.1109/ICITIIT49094.2020.9071534.

[9] X. Zhang, J. Ran and J. Mi, An Intrusion Detection System Based on Convolutional Neural Network for Imbalanced Network Traffic, 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 2019, pp. 456-460, doi: 10.1109/ICCSNT47585.2019.8962490.

[10] Zhang.C, Ruan.F, L. Yin, X. Chen, L. Zhai and F. Liu. A Deep Learning Approach for Network Intrusion Detection Based on NSL-KDD Dataset. 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, China, 2019, pp. 41-45, doi: 10.1109/ICASID.2019.8925239.

[11] Samrin.R and Vasumathi.D . Review on anomaly based network intrusion detection system. 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, 2017, pp. 141-147, doi: 10.1109/ICEECCOT.2017.8284655.

[12] Meftah S, Rachidi T, Assem N. Network based intrusion detection using the UNSW-NB15 dataset. International Journal of Computing and Digital Systems. 2019;8(5):478-87.

[13] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi and M. Ghogho. Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks. 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, 2018, pp. 202-206, doi: 10.1109/NETSOFT.2018.8460090.