International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 5, October 2020, pp. 5514~5525
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i5.pp5514-5525

An intrusion detection system for packet and flow based networks using deep neural network approach

Kaniz Farhana[1], Maqsudur Rahman[2], Md. Tofael Ahmed[3]
[1]Department of Computer Science and Engineering, Port City International University, Bangladesh
[2]Faculty of Science and Engineering, Department of Computer Science and Engineering,
Port City International University, Bangladesh
[3]Faculty of Engineering, Department of Information and Communication Technology, Comilla University, Bangladesh

Article Info

Article history:

Received Jan 26, 2020
Revised Apr 13, 2020
Accepted Apr 26, 2020

Keywords:

Big data
Deep neural networks
Intrusion detection system (IDS)
Keras
Tensorflow

ABSTRACT

Study on deep neural networks and big data is merging now by several aspects to enhance the capabilities of intrusion detection system (IDS). Many IDS models has been introduced to provide security over big data. This study focuses on the intrusion detection in computer networks using big datasets. The advent of big data has agitated the comprehensive assistance in cyber security by forwarding a brunch of affluent algorithms to classify and analysis patterns and making a better prediction more efficiently. In this study, to detect intrusion a detection model has been propounded applying deep neural networks. We applied the suggested model on the latest dataset available at online, formatted with packet based, flow based data and some additional metadata. The dataset is labeled and imbalanced with 79 attributes and some classes having much less training samples compared to other classes. The proposed model is build using Keras and Google Tensorflow deep learning environment. Experimental result shows that intrusions are detected with the accuracy over 99% for both binary and multiclass classification with selected best features. Receiver operating characteristics (ROC) and precision-recall curve average score is also 1. The outcome implies that Deep Neural Networks offers a novel research model with great accuracy for intrusion detection model, better than some models presented in the literature.

Corresponding Author:

Kaniz Farhana,
Department of Computer Science and Engineering,
Port City International University (PCIU),
Chittagong, Bangladesh.
Phone +8801857979984
Email: kanizbips@gmail.com

1. INTRODUCTION

Information and communication technology (ICT) now poses a great challenge for network engineers because of its growth over the years. Along with the advancement in the technologies the amount of threat is also increasing and handling big datasets has also become an important factor when it comes to security. For years, research on network security has been followed by a great focus and interest by the researchers. Threats can bring huge damage to systems and organizations. The challenge is to detect those intrusions over the big data and network. To detect these attacks, the network intrusion detection system (IDS) has shown great significance on the platforms that need constant security monitoring and we can say that intrusion detection is mandatory for protecting any system from malicious activities and keep the system secure to perform precisely. An intrusion detection system is a type of system that scans through the traffic of networks and looks for potential suspicious activities, analyzes those activities, alerts the system

immediately. Cyberattacks that appeared from 2001 to 2013 are discussed in [1]. Intrusion detection systems (IDSs) are the combination of hardware and software which keep tracks of the flow of data through networks and computers, also analyze the data flow to detect the threats and anomalies. It is significant for any critical network that includes security administration to get alarmed from some unauthorized activities.

There are two kinds of intrusion detection systems based on location, called host-based and network-based intrusion detection systems. Host-based intrusion detection systems work on collected information from an individual computer system while a network-based intrusion detection system collects raw data packets from the network to look for vulnerabilities. There are also two kinds of the detection system is in use based on the techniques, named as Anomaly detection and Misuse detection. Anomaly detection finds out changes in patterns or behavior on the original system or from normal traffic. The activities different from the usual activities are looked for and observed. This approach gives a high detection rate but, if the traffic is not profiled correctly or fully, this scheme can show a high false-positive rate. While misuse detection technique works by finding out abnormal behavior from both known normal and attack signatures. Then real network traffics is compared with the already collected data. This system is less productive for attacks without certain and untold patterns. Raw data packets containing the information of the network data flow, patterns are collected from networks and computers, used as big data for IDSs. An IDS classifies a network activity as 'normal' or ‘attack’ (abnormal). In this paper, we exert on the multi-class and binary classification for intrusion detection.

As time passed the amount of real-time data has increased immensely which makes it difficult for traditional intrusion detection approaches such as, support vector machines (SVM), ruled based system, data mining, fuzzy logic, machine learning (ML) [2] to handle those datasets. Also using traditional machine learning approaches for classification on training data available from large scale networks is not productive. However, deep learning organize arranges learning algorithms in layers, which can learn and make decisions by itself [3-7]. It is also being used for security measures, identifying threats and attacks. Latterly, deep neural network (DNN) techniques are showing improvement in handling big datasets with real-time computing cost and predictive accuracy. Several works can be found in the literature which discusses the intrusion detection systems thoroughly and applies current trends and techniques. Also, due to the lack of relevant datasets and limitations of traditional methods dedicated to revealing threats over the network shows despondent accuracy results.

Kim et al. [8] proposed an intrusion detection system based on Deep Neural Network (DNN). They preprocessed the data set thoroughly. The classification model was trained and tested on kdd-CUP'99 data set. The DNN model contained four hidden layers and a hundred layer units. Approximately 99% accuracy is obtained, the model also gave relatively small false rate scores. A deep belief network (IDBN) using the kdd-cup'99 data set is proposed by Liu and Zhang [9]. A deep belief network is created and the proposed model is implemented to train the network data. 92.4% accuracy is gained within 47.2s using the belief network and 91.8% accuracy is by 22.7s using IDBN. An LSTM model utilizing a rmsprop optimizer is proposed by Sara, Eric, and Kaushik [10]. The model constructed a multi-class intrusion detection system. The LSTM model is applied to the CIDDS-001 dataset and showed 0.8483 accuracies.

A RandonForest based anomaly detection model is presented and assessed on the kdd-cup'99 data set by Zhang and Zulkermine [11]. They considered two attack types and gained a detection rate of 94.7% on binary attack types. Gao et al. [12] proposed an IDS using the KDD-CUP-1999 dataset. The classification used Deep Belief Networks and performance evaluation is done by comparing the model with Support Vector Machines and Artificial Neural Networks. Using ANN obtained accuracy is 82.30% and SVM model accuracy is 86.82%. The highest accuracy is achieved with DBN at 93.49%. An intrusion detection system using Support Vector Machine has been applied on the CICIDS2017 dataset is presented by Vijayanand, Devaraj, and Kannapiran [13]. Several SVM models have been trained on multiple attack types. 99% accuracy rate is obtained.

In [14] Bipraneel and Hon introduced a deep learning technique named bi-directional long short-term memory recurrent neural network (BLSTM). After feature extraction data set training and testing were implied. The experiment result shows over 95% accuracy for the UNSW-NB15 dataset. They performed for binary classification problems with 100% precision. MD. Moin et al. [15] implemented a deep learning method for intrusion detection on the kdd and nsl-kdd dataset. They trained a deep CNN structure for feature extraction, the training also continued as a 1-NN classifier to detect the attack and achieved over 94% accuracies for both datasets.

Serpil, Zeynep and Muhammed [16] presented an intrusion detection system. They applied random forest for recursive feature elimination, also used deep learning classifiers on the CICIDS2017 data set. They achieved 91% accuracy by implementing a deep multilayer perceptron structure on features obtained from recursive feature elimination. Ustebay et al. [17] presented different ANN models to detect malicious activities. To reduce features of the CICIDS2017 dataset they employed Deep Neural Network, Shallow

Neural Network, and Auto Encoder. They compared the accuracy among these models. Their study showed several accuracies with 98.45% accuracy rate. An IDS system proposed by G. Watson [18] using the CICIDS2107 dataset. For their study the applied the model on 27 features. They gained 94.5% accuracy only for MLP, while 95.2% accuracy is obtained by using MLP and Payload model together.

Deep neural networks techniques focus on learning from attributes and perform better on imbalanced big datasets. This paper propounds a simulation of deep neural network model on CICIDS2017 publically available dataset [19] for intrusion detection and performance is evaluated for binary and multiclass with selected best features [20]. The model is executed using the Python environment, Keras and Google TensorFlow. Google developed TensorFlow, open-source software for machine learning and Deep Learning practices. Keras is a high-level application programming interface for learning, which is CPU and GPU enabled. It supports programming language python, which can run on TensorFlow.

The contents of this study are disposed of as follows. The Dataset section explores the used dataset in this study. The next section includes data preprocessing. The model implementation section explains the deep neural network model. Then the model test result and result analysis section show the experimental analysis and section conclusion deduces the paper.

2. DATASET

For intrusion detection, it is important to have certain properties in network-based datasets, such as the format and the labeling of data. For both supervised and unsupervised intrusion detection methods these properties are significantly explained at Markus et al. [21]. This section discusses the background of the CICIDS2017 dataset that is used as an intrusion detection dataset for this study. The dataset introduced by the Canadian Institute for Cybersecurity, it is wide open for all researchers [19]. It is one of the latest datasets in the literature for network intrusion detection that contains 2830743 records with 79 network traffic features and 15 attack types are available at [22]. The records in the dataset are the collection of real-world data [19, 20], spread over eight files containing five-day benign and attack activity. The format of the records is mainly packet-based and bifacial flow-based, including additional metadata [21]. Also, the dataset is fully labeled. The intention of generating the dataset is for network intrusion detection, therefore it focuses on the attack types mentioned in Table 1. For binary classification all attack types are considered as '1' and benign attacks are considered as '0'. For multi-classification, each attack type is considered as they are given.

The attacks are comprised of seven common attack families as Botnet, DoS attack, DDoS attack, Brute force attack, Web attack, Infiltration attack, and Heartbleed attack. Brute force attack is used to crack passwords, locate hidden contents, hit and try an attack. Botnet attacks perform attacks through internet-connected devices and send spam. DoS attack makes the system unavailable for some time, it overloads the system networks. DDoS attack occurs when multiple systems fall into a victim. Web attacks are software programs written to attack the user system, look for vulnerabilities. In an infiltration attack, after exploiting the user system a backdoor will be created to execute attacks on the system. Heartbleed attack works by deceiving servers, gaining private encryption key and leaking their information.

Table 1. Dataset and attack types

File Names (.csv file format) + Activity Day	Instances	Attack Types
Friday Working Hours Afternoon DDos	225745	Benign, DDoS
Friday Working Hours Afternoon PortScan	286467	Benign, Port Scan
Friday Working Hours Morning	191033	Benign, Bot
Monday Working Hours	529918	Benign
Thursday Working Hours Afternoon Infiltration	288602	Benign, Infiltration
Thursday Working Hours Morning WebAttacks	170366	Benign, Web Attack Brute Force, Web Attack Sql Injection, Web Attack XSS
Tuesday Working Hours	445909	Benign, FTP Patator, SSH Patator
Wednesday Working Hours	692703	Benign, DoS-GoldenEye, DoS-Hulk, DoS Slowhttptest, DoS-Slowloris, Heartbleed

3. DATA PREPROCESSING

The phase of data preprocessing is significant to transform raw data into the penetrable format as real-world data is often imperfect, incompatible in certain ways. It minimizes the ambiguity present in the used dataset and to bring out the accurate statistics of the detection system. The data preprocessing phase differs from data to data and study by study. In this section data preprocessing is discussed through sub-sections as feature selection, removing duplicates, class imbalance, normalization, and label encoding. Figure 1 manifests the process of data-preprocessing.

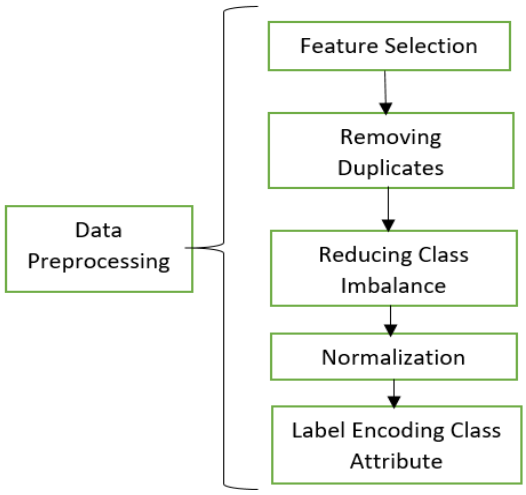

Figure 1. Data preprocessing

3.1. Feature selection

When working with high dimensional data, models can retch because of the increased training time. Training time is influenced by the number of features, also there is a chance of overfitting if too many features are used while training the model. So, feature reduction is important for building a model with high dimensional data without much loss of total information. The creator of the CICIDS2017 dataset explained the optimal features set depending on each attack type in [19]. Random Forest Regressor has been used as a classifier and observes the best features for each attack type among 79 features according to the weight of the features. In this study, 24 features including class label [19, 20] have been considered as the best-selected features for detection. Table 2 describes the selected features [23, 24].

Table 2. Best features

Feature Name	Description
Flow duration	Flow duration
Total length of the forward packets	Combined size of the packets in forward direction
Subflow forward_bytes	Average number of the bytes present in a subflow to the forward direction
Backward packet length std	Standard deviation length of packets in the backward direction
Flow IAT Min	Minimum Time between two packets passed through the flow
Flow IAT Mean	Mean time across two packets passed in the flow
Flow IAT std	Time of standard deviation between two packets
Forward IAT Min	Minimum time across two packets passed in the forward direction
Active Min	Before turning into idle the minimum activation time of a flow
Active Mean	Before turning into idle the mean active time of a flow
Average_Packet_Size	Average measurement of packet
Backward Packet Length Min	Minimum packet length in the backward direction
Forward_IAT_Mean	Mean time across two packets passed into forward direction
Backward_IAT_Mean	Mean time across two packets passed into backward direction
Initial Win bytes forward	Total number of bytes passed through initial window into forward direction
ACK flag _Count	Number of packets on acknowledgment
SYN flag_Count	Number of packets on synchronization
Forward push_flag	Numbers of how many times push flag set in packets to travel into forward direction
Flow_packets	Number of packets carried per second
Initial win_bytes backward	Total number of bytes passed through initial window into backward direction
Forward Packet Length Mean	Mean packet length into forward direction
Backward Packets	Number of packets carried per second
Push Flag_Count	Packet numbers on PUSH

3.2. Removing duplicates

Some of the records in the dataset have duplicate values. For precise evaluation, it was necessary to remove duplicate values from the dataset. After removing duplicate records number of instances has reduced to 2286604 for multi-class attack types. Figure 2 shows the distribution of each class label for multi-classification.

3.3. Class imbalance

It is observed that the size of the dataset combined is huge, also the class imbalance situation can be seen from Figure 2. If the combined dataset is used for training the model, then the model will likely show results biased towards the majority class type. Because of this class imbalance issue, the model can give a high false alarm rate with low accuracy. For this study, to reduce the effect of class imbalance problem 'Monday-WorkingHours.pcap_ISCX.csv' dataset which contains only benign records has been taken out of consideration. The amount of benign record is more than 80% which can make the classifier over-fitted towards the normal attack. Without the mentioned dataset the total number of instances for the multi-class dataset is 1835569.

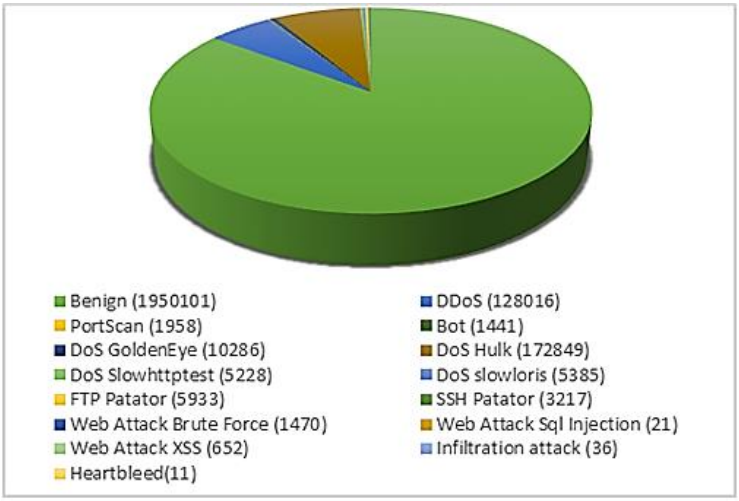

Figure 2. Attack instances number for multi-class

3.4. Normalization

Normalization is important for data preparation in neural networks because the layers in the model act sensitive depending on the weights of the data. The features of the CICIDS2017 dataset have values of different ranges which make the dataset incomparable. So, to keep the values in a common range normalization is applied, which makes the learning process easier for neural network layers. All the values are normalized within a range of 0 to 1 using max-min normalization.

$$x' = (x - min)/(max - min) * (new\_max – new\_min) + new\_min$$

The status of each record before and after normalization is shown below.

Before Normalization (snippet of dataset)
3, 12, 6, 0, 0, 3, 0, 3, 3, 3, 0, 0, 666666.7, 0, 0, 0, 1, 9, 12, 33, -1, 0, 0, BENIGN

After Normalization (snippet of dataset)
0, 0.000066, 0.001552, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.222222, 0, 0, 0, 1, 0.00356, 0.000066, 0.000519, 0, 0, 0, BENIGN

3.5. Label encoding

For binary classification benign and attack are considered as '0' and '1' respectively. And for multi-class classification Label encoding has been applied on the dataset because the multi-class labels have string names. It is vital to binarize those string names into numeral values before passing the dataset through the classification model. Label encoding is used rather than randomly numbering the strings to avoid any biases in the model towards the hierarchy of the numbers of class type. After encoding the class type into binary codes it is used as the output layer for the multi-classification with n classes. Label encoding is summarized in Algorithm 1. After label encoding, the class attribute for each record forms an array with binary values assigned to every 15 classes randomly. Each record looks like the below array fragment.

[1, 0, 0, ..., 0, 0, 0]
[1, 0, 0, ..., 0, 1, 0]
[1, 0, 0, ..., 0, 0, 0]
.....
[1, 0, 0, ..., 0, 0, 0]
[1, 0, 0, ..., 1, 0, 0]
[1, 0, 0, ..., 0, 0, 0]

Algorithm 1 : Label Encoding for multi-class classification

Input: Dataset D with features f_0, f_1, ……, f_{n-1} and attack classes (1…….C) for each records in the dataset
Output: label encoded class feature
1: LabelBinarizer()
2: lb.fit_transform()
3: y ← label_binarize() #binary coded class label
4: end

After encoding, class distribution is shown in Table 3.

Table 3. Class distribution

Class number	Class name
Class-0	Benign
Class-1	Bot
Class-2	DDoS
Class-3	DoS Goldeneye
Class-4	DoS Hulk
Class-5	DoS Slowhttptest
Class-6	DoS Slowloris
Class-7	FTP-Patator
Class-8	Heartbleed
Class-9	Infiltration
Class-10	PortScan
Class-11	SSH-Patator
Class-12	Web Attack-Brute Force
Class-13	Web Attack-Sql Injection
Class-14	Web Attack-XSS

4. RESEARCH METHOD

For the implementation process, we used Jupyter notebook, python programming environment through TensorFlow and Keras library. After data preprocessing all the datasets are separated into two files for binary and multi-class classification. Figure 3 depicts the methodology of this study. First, the dataset is preprocessed so that the model can be applied to the dataset thoroughly. The dataset with selected 24 features (including the class type) has been divided into 65% of the training set and 35% of the test set. The datasets are divided with the same proportion for both binary and multi-class classification.

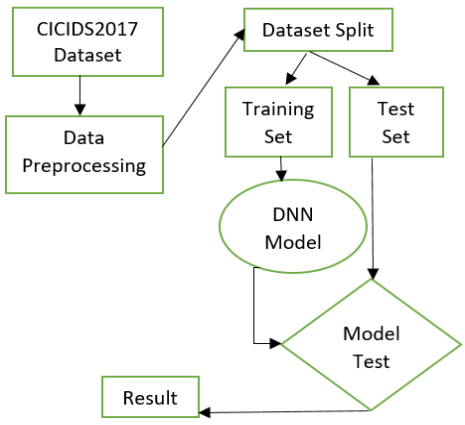

Figure 3. Methodology

4.1. Proposed model

The proposed deep neural network model is built with four layers, where all the nodes in the layers are fully connected uses backpropagation for the learning model. DNN produces output values by calculating the hidden layer neuron weights. So, the model constructs with one input layer, two hidden layers, and one output layer. The number of hidden layers is not increased to avoid the vanishing gradient problem, also many hidden layers can produce results with very high sensitivity. Figure 4 gives an idea of the proposed model.

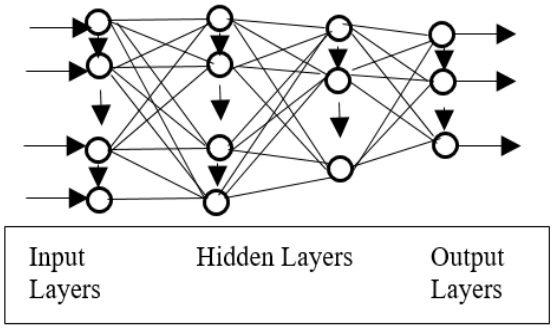

Figure 4. Deep neural network

As activation function, ReLu is used in both input and hidden layer for easier mathematical operation, while sigmoid function is activated for binary classification and activation function softmax is employed on the final layer for multi-class classification respectively. Input dim is set as 23 as the input features, excluding the class type feature. The first layer is set for 128 nodes, two hidden nodes layers are set to 64 and 32 respectively while the output layer node is set as 1 for binary class model and for the multi-class problem it is set according to the N class number. N classes are the count of benign and attack types. For both scenarios training batch size is set as 10 thousand (for a single batch of the sample) and the epoch is set to 150 (total pass number over the complete training set). The proposed model is summarized in Algorithm 2.

Algorithm 2: Proposed DNN model

Input: Dataset D with features f_0, f_1, ……, f_{n-1} and attack label encoded (1…….C) for each record
1. Train test split
2. Build the model on training dataset - Sequential()
3. Adding layers to the neural network - model.add()
4. Compile the model and calculate the accuracy
   model.fit( )
   for i to n - 1 Each layer according to the batch size and epochs
   | calculate the performance (lose, accuracy) of each layer
5. End
return ← accuracy
6. ROC and Precision-recall calculation
return ← roc and precision-recall curve

Here, step 2 defines the Keras model for classification. On step 3 new layers are added to the model. Then, the model is trained over 65% of the dataset, calculating the accuracy, loss score for each layer. After step 4, the average value of accuracy and loss is counted. Finally, roc and precision-recall scores are calculated to understand the performance of the proposed model on both multi-class and binary classification problems.

5. TEST RESULT AND PERFORMANCE ANALYSIS

We measured the values of the loss function and accuracy on the test dataset for binary and multi-class classification with 24 best-selected features including the class label. The performance of the model is also estimated under the receiver operating characteristic (ROC) curve and precision-recall curve. ROC refers to a graphical illustration to show the potential of a classifier. True positive and false positive rates are the two parameters used in roc. Metrics defined below are used to measure accuracy, roc and precision-recall curve [25].

- True Positive-TP -represents attack data which is correctly classified as an attack type.
- False Positive-FP -refers to the data that is not correctly identified as an attack type.
- True Negative-TN -alludes to data that is correctly identified as a normal type.
- False Negative-FN -refers to attack data that is not correctly categorized as a normal type.

Precision, recall score delivers the success of prediction on the imbalanced classes. Precision refers to the result of relevancy, where recall calculates true relevant events that have been found. Precision refers to tp/tp+fp and Recall refers to tp/tp+fn. The precision-recall curve illustrates the trade-off relation between recall and precision for non-identical thresholds. The top area under the curve means peak values for recall and precision, but top precision means the lowest false-positive rate and top recall means lowest false-negative rate. Peak scores show that classifier giving precise results as high point precision. Table 4 gives the result, where the DNN model with two hidden layers achieved high accuracy rates. For binary classification Table 5 shows the precision, recall scores. And, the plot is shown in Figure 5 shows the ROC curve value as 1 for binary classification.

Table 4. Test result 1

	Binary Classification	Multi-class Classification
Accuracy	99.13%	99.29%
Loss	0.0232	0.0289

Table 5. Test result 2

	Binary classification
Precision	0.99
Recall	0.96

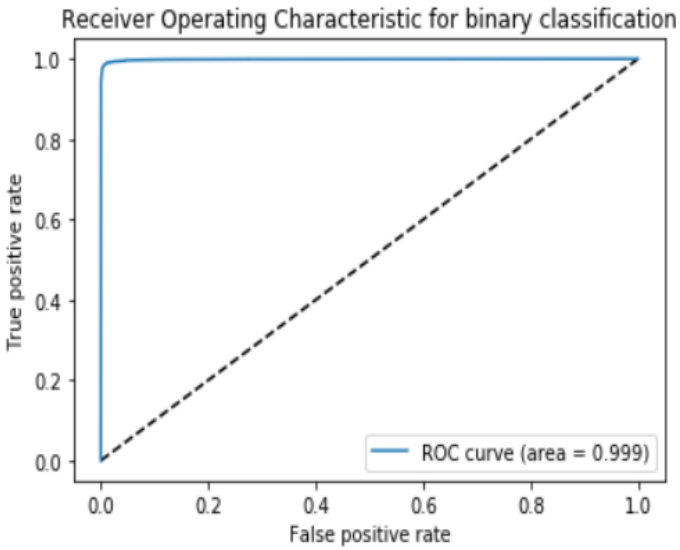

Figure 5. ROC for binary classification

The plot is shown in Figure 6 is the ROC for multiclass, which gives ROC area under the curve value as 1 for all 15 classes. ROC AUC value varies between 0 and 1, while 1 is considered as a perfect performance. Figure 7 depicts the Precision-recall curve for multi-class attacks. Table 6 shows the precision-recall curve scores concerning the attack name and it is observed from Table 6, that the precision-recall curve score for attack classes 8, 9 and 13 is 0. Class 8, 9 and represents the attacks 'Heartbleed', 'Infiltration' and 'Web Attack Sql injection' respectively. From Figure 2, we can see that these attack types have a very small percentage compared to other attack types. This class imbalance problem and a small number of attack instances are the reason why the model cannot give the precision-recall curve score for those specific attack types. Percentage and precision-recall curve scores for other Web attacks are also relatively low.

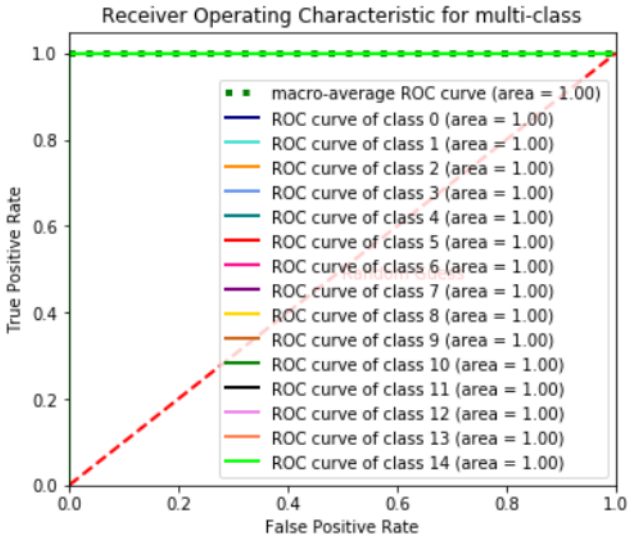

Figure 6. ROC for multi-classification

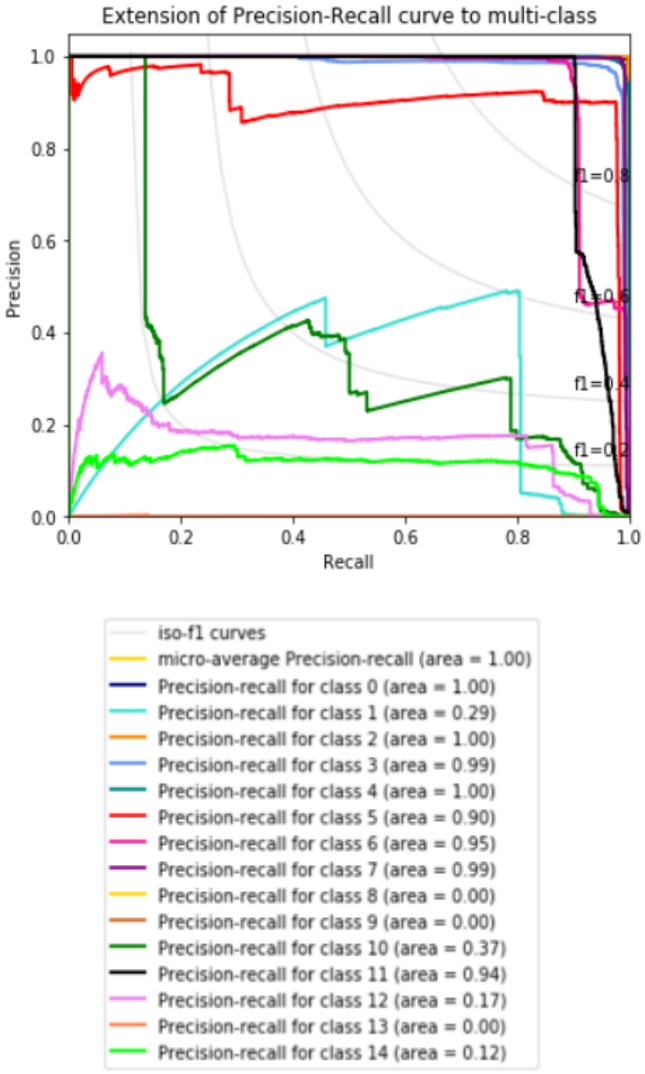

Figure 7. The precision-recall curve for multi-class

Table 6. Test result 3

Class name	Precision-recall curve score
Benign (Class 0)	1
Bot (Class 1)	0.54
DDoS (Class 2)	1
DoS Goldeneye (Class 3)	0.99
DoS Hulk (Class 4)	1
DoS Slowhttptest (Class 5)	0.82
DoS Slowloris (Class 6)	0.97
FTP-Patator (Class 7)	0.99
Heartbleed (Class 8)	0
Infiltration (Class 9)	0
PortScan (Class 10)	0.69
SSH-Patator (Class 11)	0.94
Web Attack Brute Force (Class 12)	0.24
Web Attack Sql Injection (Class 13)	0
Web Attack XSS (Class 14)	0.19

Figure 8 shows the average returned for the precision-recall curve score as 1. CICIDS2017 is one of the inclusive data set used in literature. From Table 4 we have seen that the accuracy of the model for binary and multi-class classification is relatively better in contrast with the studies [16, 17] that used the same dataset for deep neural networks. Table 7 shows a performance comparison between the proposed model and other studies.

It is also visible that feature selection and data-preprocessing effects the performance of models used in the classification. In this study, the selected features and data-preprocessing results over a 99% accuracy rate that makes the DNN model better than the previously mentioned studies [16, 17] of the DNN model.

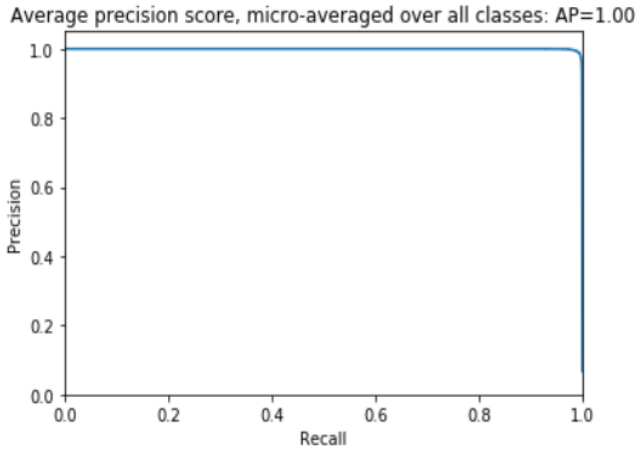

Figure 8. Averaged precision-recall curve for multi-class

Table 7. Comparison

	The proposed method		The study [16]	The study [17]			
	Binary Classification	Multi-class Classification	Binary Classification	Multi-class Classification			
				All features	10 features	20 features	30 features
Accuracy	99.13%	99.29%	89%	98.40%	94.72%	95.92%	96.71%

6. CONCLUSION

An IDS model is proposed in this study using Deep neural networks and the model is trained and tested on the CICIDS2017 dataset, contains real network traffic data. The huge size of the dataset was a challenge to handle. Label Encoding is performed on the data preprocessing stage to ensure that the model does not show biasness for any attack classes. The deep neural network is used so that the dataset can produce results with good accuracy and speed. In this study, the proposed neural network model gives

99.13% accuracy for binary classification and 99.29% for multi-class classification. The model also scores 100% for the precision-recall curve and roc curve. Also, the model gives faintly better accuracy results for multi-class classification results than binary classification.

However, the model could not classify 'Heartbleed', 'Infiltration' and 'Web Attack Sql injection', due to the low number of records presented in the data set itself. This study offers a contribution to the literature over the performance measure of IDS using DNN on the packet and flow-based dataset, as the CICIDS2017 dataset has been also updated by creators. This study provides an overview for the researchers to focus on the limitations of detecting the low numbered instances and also help to design, analyze and understand the intrusion detection model using other relevant deep learning techniques over the dataset. In the future, we are planning to work on a detection model to detect the low numbered instances.

REFERENCES

[1] T. Vaidya, "2001-2013: Survey and analysis of major cyberattacks," *arXiv preprint arXiv: 1507.06673*, 2015.

[2] H. Azwar, et al., "Intrusion Detection in secure network for Cybersecurity systems using Machine Learning and Data Mining," *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, Bangkok, Thailand, pp. 1-9, 2018.

[3] S. Naseer, et al., "Enhanced Network Anomaly Detection Based on Deep Neural Networks," in *IEEE Access*, vol. 6, pp. 48231-48246, 2018.

[4] G. Karatas, et al., "Deep Learning in Intrusion Detection Systems," *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, Ankara, Turkey, pp. 113-116, 2018.

[5] F. A. Khan, et al., "A Novel Two-Stage Deep Learning Model for Efficient Network Intrusion Detection," in *IEEE Access*, vol. 7, pp. 30373-30385, 2019.

[6] N. Shone, et al., "A Deep Learning Approach to Network Intrusion Detection," in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, Feb. 2018.

[7] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," *2018 20th International Conference on Advanced Communication Technology (ICACT)*, Chuncheon-si Gangwon-do, Korea (South), pp. 1-1, 2018.

[8] J. Kim, et al., "Method of Intrusion Detection using Deep Neural Network," *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jeju, pp. 313-316, 2017.

[9] Y. Liu and X. Zhang, "Intrusion Detection Based on IDBM, in 2016 IEEE 14th Intl Conference on Dependable, Autonomic and Secure Computing," *14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, pp. 173-177, 2016.

[10] S. A. Althubiti, et al., "LSTM for Anomaly-Based Network Intrusion Detection," *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, NSW, pp. 1-3, 2018.

[11] J. Zhang and M. Zulkernine, "A hybrid network intrusion detection technique using random forests," in *The First International Conference on Availability, Reliability and Security, (ARES 2006)*, 2006.

[12] N. Gao, et al., "An Intrusion Detection Model Based on Deep Belief Networks," in *2014 Second International Conference on Advanced Cloud and Big Data*, pp. 247-252, 2014.

[13] R. Vijayanand, et al., "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computer and Security*, vol. 77, pp. 304-314, 2018.

[14] B. Roy and H. Cheung, "A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short Term Memory Recurrent Neural Network," *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, NSW, pp. 1-6, 2018.

[15] M. M. U. Chowdhury, et al., "A few-shot deep learning approach for improved intrusion detection," *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, New York, NY, pp. 456-462, 2017.

[16] S. Ustebay, et al., "Intrusion Detection System with Recursive Feature Elimination by Using Random Forest and Deep Learning Classifier," *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, Ankara, Turkey, pp. 71-76, 2018.

[17] S. Ustebay, et al., "Cyber Attack Detection by Using Neural Network Approaches: Shallow Neural Network, Deep Neural Network and AutoEncoder," in *International Conference on Computer Netwoks*, pp. 144-155, 2019.

[18] G. Watson, "A Comparison of Header and Deep Packet Features when Detecting Network Intrusions," University of Maryland, 2018.

[19] I. Sharafaldin, et al., "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *4th International Conference on Information Systems Security and Privacy*, pp. 108-116, 2018.

[20] I. Sharafaldin, et al., "A Detailed Analysis of the CICIDS2017 Data Set," *International Conference on Information Systems Security and Privacy*, pp. 172-188, 2019.

[21] M. Ring, et al., "A Survey of Network-based Intrusion Detection Data Sets," *arXiv: 1903.02460*, 2019.

[22] UNB, "Intrusion Detection Evaluation Dataset (CICIDS2017)," University of New Brunswick. Available: https://www.unb.ca/cic/datasets/ids-2017.html.

[23] A. H. Lashkari, et al., "Characterization of Tor Traffic using Time based Features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, vol. 1, pp. 253-262, 2017.

[24] Arash H. L., et al., "CICFLOWMETER," *UNB, Canadian Institute for Cybersecurity*, [Online], Available: www.netflowmeter.ca.
[25] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2016.

BIOGRAPHIES OF AUTHORS

Kaniz Farhana, Graduate Student at Port City International University (Bangladesh), completed B.Sc. Engineering at University of Science & Technology, Chittagong (USTC) in the year of 2018. Research interest includes Deep Learning Approaches, Cyber Security, Performance Analysis. Working as an Information and Communication Technology (ICT) lecturer at South Point School and College (Bangladesh).

Maqsudur Rahman, Senior Lecturer at Department of Computer Science and Engineering, Port City International University, (Bangladesh). Teaching area includes Computer Programming, Computer Networks, Data Communication & Mobile Communication. Research interests lie at Digital Communication, IoT and Network Security.

Md. Tofael Ahmed, Associate Professor of ICT, Comilla University (Bangladesh). Research area includes Cyberbulling Detection, Data Mining, Text Analytic, Big Data, Online Social Network analysis.