Prospect of Machine Learning Scheme for Efficient Detection of DDoS Attacks in IoT Networks

[1]Kazeem B. Adedeji, [2]Sammy O. Oladiran, [1]Segun V. Abokede, and [1]Omojola Ogunlade

[1]Department of Electrical and Electronics Engineering, Federal University of Technology Akure, Ondo State, Nigeria

[2]Department of Information and Communication Technology, Federal University of Technology Akure, Ondo State, Nigeria

***Abstract*—The Internet of Things (IoT) has seen widespread deployment due to its capability to offer seamless connections. However, because of its innate security architecture, the IoT is currently experiencing an increase in attacks. One of the most prevalent attacks against IoT systems is the distributed denial of service (DDoS) attack. In this paper, we investigate DDoS attack detection for the IoT using machine learning and deep learning. Three classifiers; multi-layer perceptron (MLP), k-nearest neighbors (k-NN), and deep neural network (DNN) were developed for attack detection. Unlike previous studies, we investigated these models considering both binary and multiclass classifications. The performance of these models in both classifications was verified using the CICIDS 2017 and Bot-IoT datasets. Experimental results show that the performance of the models is incredible for multiclass classification compared to binary classification. We are able to show that by using such multiclass classification, the accuracy of the models can be improved and effective detection for IoT networks can be achieved. For the CICIDS2017 dataset, all the models recorded accuracy close to 100%. The MLP has 99.984% accuracy, while k-NN and DNN record 99.994% and 99.987% accuracy, respectively. The detection accuracy for binary classification is also superb for both MLP and k-NN. However, we noticed that the DNN doesn't seem to be an excellent model for DDoS prediction when binary classification is considered.**

Keywords—attack detection; cyber attack; DDoS; IoT; machine learning

I. INTRODUCTION

Over the last few years, the use of the internet and internet-enabled applications has increased dramatically and has become more indispensable to today's generation. Many modern smart devices are now internet-enabled and linked to the internet via IoT. The IoT is a platform that enables a network of linked devices to connect to the internet and communicate with each other. IoT has become more popular today due to the increase in the number of devices linked to the internet. It is predicted that this trend will continue in the years to come. Since 2015, billions of IoT devices have been connected globally [1]. By 2025, there will be more than 75 billion linked IoT devices in use according to the forecast published in [2]. The IoT and other internet-enabled networks have recently emerged as one of the enabling technologies that have been implemented in a variety of applications [3, 4]. Due to the inherent security issues connected with IoT devices, the use of IoT in many of these applications has generated a great deal of controversy. Most IoT devices feature web interfaces that do not demand the use of secure passwords. Some of them continue to provide access to people who have repeatedly failed to log in. As a result, these interfaces are vulnerable to several attacks. Most IoT devices lack access control, have insecure default passwords, and use unprotected credentials. As a result, an attacker could take advantage of this to compromise data integrity and privacy.

DDoS is a set of well-organized attacks launched remotely using distributed botnet computers in a network. A botnet is a massive network of hundreds or thousands of hacked machines that can be remotely controlled and that are used to attack a specific server or network [5]. In a DDoS attack, numerous devices attack a single server or network. These attacks are executed with networks of internet-connected devices—including PCs and other devices (such IoT devices) that have become infected with malicious software and are thus susceptible to remote manipulation. These devices are known as bots. Once a botnet has been established, the attacker can direct the attack by sending remote commands to each bot. Each of the bots in the botnet sends queries to the IPs of the victim's server while it is being targeted by the botnet which may overwhelm the network and disrupt legitimate traffic. Each bot is a real internet device which makes it challenging to differentiate between attack and legitimate traffic. Since a DDoS attackers initiate an attack via a botnet; therefore, the architecture of a DDoS attack will consist of an attacker, botnet, and the target network or server. In a decentralized DDoS architecture shown in Fig. 1, the bots establish a peer-to-peer (P2P) network. An attack query is sent to a certain bot to start the DDoS attack.

The commands are then forwarded by this bot through P2P to other bots in the network.

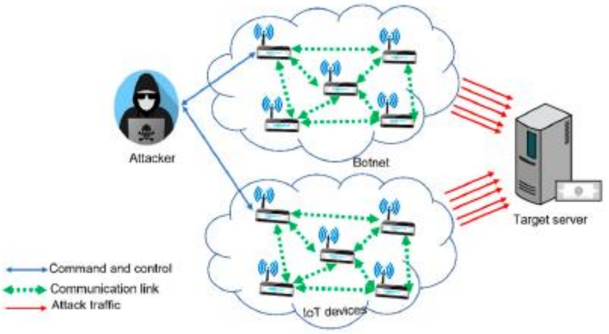

Fig. 1. *Architecture of a DDoS attack [6].*

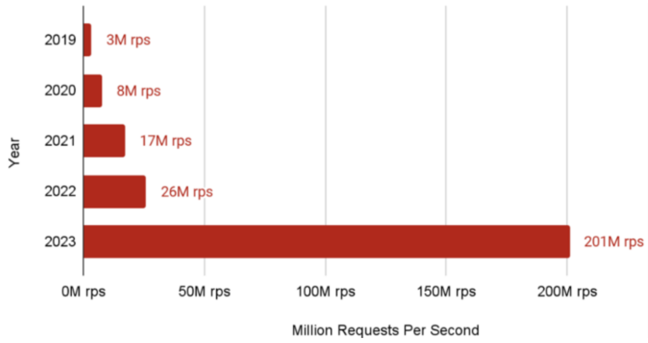

Fig. 2. *Largetr HTTP DDoS attack recorded by year [7].*

DDoS attacks are frequently launched on target networks with high volumes of traffic. A report presented in [7] revealed that during the 4th quarter of 2023, more than 200 million request per second has been sent by attackers using the HTTP request to flood the victim server (see Fig. 2).

Despite persistent efforts to prevent, detect, and mitigate DDoS attacks on IoT networks, these destructive attacks remain prevalent. Therefore, finding solutions to this problem continues to be a critical challenge in the field of network security. Due to the significant effect of these attacks, it is crucial to develop an efficient attack detection method. In the past, several methods have been proposed with varying degrees of effectiveness. Among these methods, machine learning are major participants and are currently being employed due to their ability to learn traffic and attack features and have been used to predict DDoS attacks. In this paper the potential of three machine learning models (MLP, k-NN, and DNN) for predicting DDoS attacks in IoT networks is presented. The performance of these models was demonstrated using the CICIDS 2017 and Bot-IoT datasets. The key contributions of this research are as follows:

- Unlike other studies [8-12], experiments and analyses were performed with a focus on both binary and multiclass classification and compared the potential of these models under each classification scenario.
- In this study, we verify the effect of classification scenarios on the performance of these models, and we are able to show that the DNN doesn't seem to be a good model for DDoS prediction when binary classification is considered.

The rest of the paper is organized as follows: Section 2 presents the attack detection methods and a review of related research studies. Section 3 shows the methodology of this research, while Sections 4 and 5 present the results and conclusion, respectively.

II. Literature Review

There are several methods used for the detection of DDoS attacks in IoT networks. These attack detection methodologies are classified as traditional methods, signature-based detection, and anomaly-based detection. The Traditional methods concentrate on measuring the traffic volume. When the measured traffic volume exceeds a predetermined level, a DDoS attack is identified [6]. The traditional detection methods suffer from high false alarm rate and low detection accuracy, thus, are seldom used. For the signature-based detection, attack signatures stored in a database are utilized to find attacks. This is achieved by tracking traffic patterns and comparing them to pre-existing signatures. Once a disparity is detected between the previously recorded patterns and the incoming pattern, this is indicated as indicated as malicious traffic. The method has high accuracy in detecting known attacks, provided the database is updated. The major challenge with this method is that only attacks whose signatures have been previously stored in the database can be detected. The anomaly based method are mostly used for attack detection in IoT due to their ability to detect unknown and zeroday attacks. This is achieved by identifying anomalous circumstances caused by the attack.

Statistical methods such as entropy analysis [13-15] and machine learning (ML) methods [8-12] are typically utilized in the anomaly detection approach. Entropy-based DDoS attack detection methods strongly depend on the use of thresholds to achieve the desired detection results. In most cases, it is hard to choose the right detection threshold in various attack environments due to the changing nature of network traffic patterns and rising attack intensities. To improve on this limitation, machine learning techniques are currently being used because they can learn the characteristics of traffic and create a very precise model for identifying anomalous traffic features.

Moore and Zuev [16] utilize Bayesian techniques to classify internet traffic patterns for DDoS attack detection. The method records a relatively low detection accuracy of 60%. In [17], network traffic samples were collected via sflow protocol from network devices. These samples were classified using a random forest (RF) classifier. The network traffic was compared to signatures collected earlier from network traffic samples to achieve attack detection. The method was via datasets that comprised of the CIC-DoS, CICIDS2017, and CICIDS2018. The reults obtained revealed that a 96% detection rate, a relatively high level of precision, and a low false alarm rate were achieved. In [18], the authors employ SVM for DDoS attack detection in an SDN-based IoT network. In this study, a routine collection of network packets was conducted. From this, 24 features were extracted. The SVM is then used to categorize these features to detect abnormalities. The method was validated through the NSL–KDD dataset, and its performance was compared to that of the J48 and Naïve Bayes (NB) classifiers. The result obtained record a detection accuracy of 99.4%, compared to 99.75% and 95.87% for the J48 and NB algorithms, respectively. It can be shown that the J48 classification method continues to perform better in terms of accuracy than the suggested approach. The method also has a significant processing overhead. Similarly, the authors [19] use SVM to classify additional traffic features that are periodically obtained from a flow table. These are aggregated features that pertain to DDoS attacks. They include the speed of the source IP and port, the speed of flow entries, the standard deviation of the flow bytes and packets, and the ratio of pair–flow. The validity of the approach was verified by simulation. Evaluation results show that a detection rate of 95.24% was achieved, even with a small amount of flow data. The method does, however, record some false alarms. The average false alarm rate generated was 1.26%.

In Chen et al. [20], a DT classifier was employed for DDoS attack detection in a multi-layer IoT environment. Experimental results show that ICMP flooding, SYN flooding, and UDP flooding were detected with 97.39% accuracy and an F1-score above 97%. Mihoub et al. [21] proposed attack detection and mitigation architecture for IoT networks using machine learning. In this study, a multi-class classifier was developed using DT, RF, k-NN, multi-layer perception (MLP), RNN, and LSTM to classify the extracted features from the BoT–IoT dataset. This classifier follows the looking-back idea, where the sub-categories of the attacks are also localized. Evaluation results show that looking-back-enabled RF has the highest accuracy, while the lowest is observed with the k-NN under the same concept. The authors of [22] implemented k-NN, SVM, NB, DT, RF, and LR machine learning algorithms in WEKA tools to analyse their detection performance using the CICDDoS2019 datasets. Evaluation results show that both DT and RF record the highest accuracy, while the NB has the lowest detection accuracy. Nevertheless, the DT has superior performance in terms of processing time. The DT classifier requires 4.53 s to process the data, whereas the RF classifier needs roughly 84.2 s.

The authors of [23] analysed the potential of SVM, MLP, DT, and RF classifiers for attack detection in a simulated SDN environment using Scapy tool with a list of valid IPs. Results show the superiority of the RF over other classifiers in terms of detection accuracy. The DT, however, has a quicker processing time. The primary drawback of this study is that all traffic was generated artificially and that some traffic characteristics, including IP, protocols, and packet size, were randomly selected. The choice of these features was not discussed. Additionally, these features were insufficient to provide successful detection performance.

In [10], RF, C5.0, NB an SVM classifiers were used for attack prediction in IoT network using the CICIDS2017 dataset. A detection accuracy of 86.8%, 86.5%, 80% and 79.9% for RF, C5.0, NB and SVM respectively was achieved. In [11], the authors proposed the use of DT classifier for attack detection in IoT network using the CICIDS2017 dataset. In this study, a detection accuracy of 96.36% and processing time of 16.58 secs were achieved. In [13], a DT and SVM classifier were used for attack prediction using the same dataset used in previous studies [10, 11] with 98.98% and 97.97% accuracies for DT and SVM classifiers respectively. These studies, however, focused only on binary classification. Similarly, a binary classification study was presented in [12] with the same dataset to train a RF classifier for attack detection in IoT network. The result obtained show that an accuracy of 99.79% was achieved with the RF classifier. Other studies [24-26] considered multiclass classification with varying accuracy level. The study in [24] considered multiclass classification with the use of DNN. When validated using the CICDDoS 2019, an accuracy of 94% was recorded. In [26], SVM, NN, J-48 and RF were used to predict DDoS attacks in IoT network. In this study, SVM record 88.5%, NN with 99.4%, J-48 with 99.7% and RF with 99.7% accuracies.

III. Research Methods

Fig. 3 represents the block diagram of the methodology used in this paper. In this study, we have used two datasets: the CICIDS2017 [27], and Bot-IoT dataset [28]. These data were pre-processed using a variety of methods. This included the removal of null values from the datasets, and then balancing and normalization techniques were applied to scale and balance the dataset. At the feature ranking stage, the best features were extracted from the dataset. The data was then divided into 30% for testing and 70% for training sets. The testing set is used to evaluate the models, while the training set is used to train the k-NN, MLP, and DNN classifiers.

A. *Data pre-processing*

- CICIDS2017 data processing

The CICIDS2017 dataset contains benign (normal) and the most updated attacks in pcap and csv file formats. It includes the result of network traffic analysis using CICFlowmeter, with the flows labelled based on the time stamp, source and destination IPs, source and destination ports, protocols, and attack. The dataset is available in eight different csv files, and each file contains a different attack. The 8 files were concatenated using the Python Pandas data frame. Pre-processing is done on the dataset to improve its suitability for the classifier. Null entry removal, data balance, label encoding, and normalization are all steps in this process. The dataset consists of 2,830,743 rows with 78 attributes and a column for

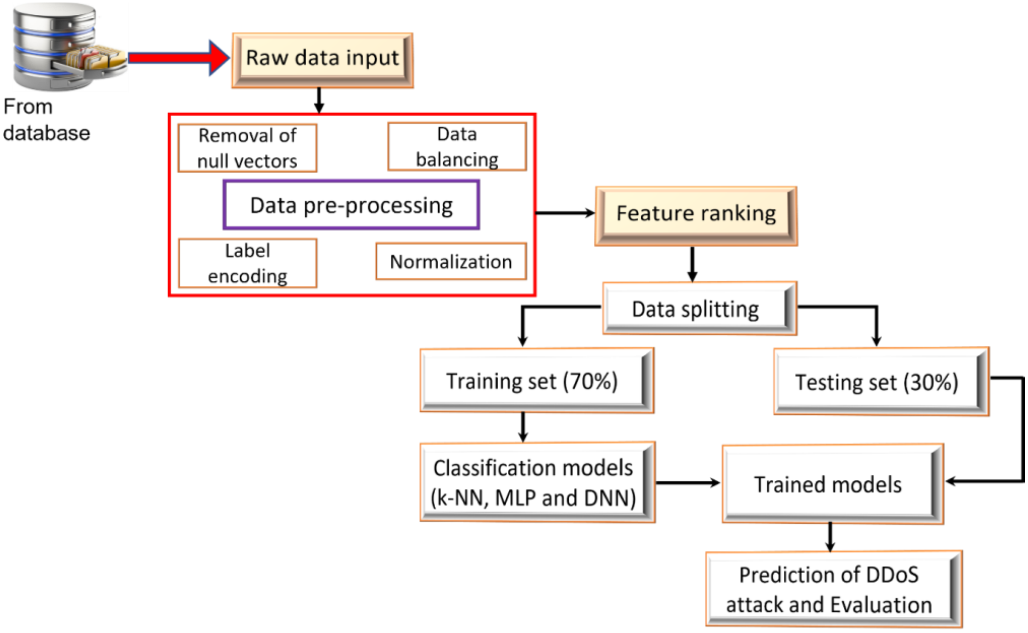

Fig. 3. *Block diagram of the research methodology.*

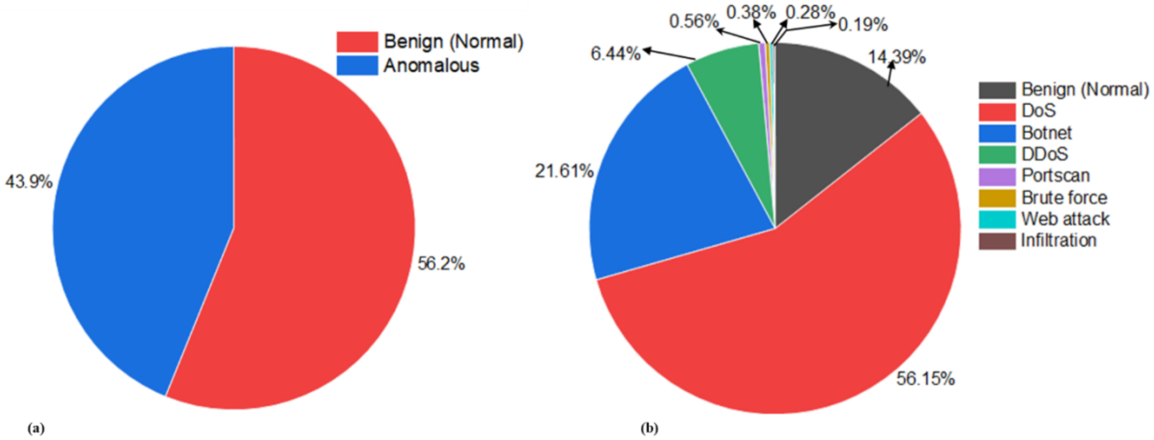

Fig. 4. *Target data distribution in the CICIDS2017 dataset for (a) binary classification, (b)multiclass classification.*

class label. First, the Pandas libraries were used to import and concatenate the dataset csv files. The dataset was condensed to 890,353 entries in order to increase computational capacity, and after that, it was verified for null entries that might have an impact on the models' performance. The dataset's null entries were removed, leaving 889842 rows and 78 attributes in the final dataset. The dataset columns were found to have some unidentified characters, so these unidentified characters were eliminated from the column names. Due to the dataset's imbalance, several attack types, particularly those from minority classes, were combined to create new attack types. This is to minimize the imbalance in the dataset. In this study, we categorize the attack features in the dataset using both binary and multiclass classification. The dataset contains attack types such as Dos_Hulk, PortScan, DDoS, DoS_GoldenEye, FTPPatator, SSHPatator, DoS_Slowloris, DoS_Slowhttptest, Heartbleed, Bot, Web_Attack_Brute_force, Web_Attcack_XSS, Web_Attcack_Sql_Injection, and Infiltration. For multiclass classification, these attack types were grouped together and classify each entry into eight operational states, with attack types represented as either "normal (benign or no attack)", "port scan", "DoS", "DDoS", "brute force", "botnet", "web attack", and "infiltration". The binary classification categorized the features for each entry as either normal (no attack) or abnormal (attack) by creating a binary label, and the values for the column label were populated using the *NumPy* library in Python. The data is visualized, and the distribution of data for both classifications is illustrated in Fig. 4.

- Bot-IoT data processing

This dataset is one of the newest utilized for attack detection in IoT networks. It was generated by designing a realistic network environment in the Cyber Range Lab at UNSW Canberra. It contains a combination of simulated and real-world settings and more than 72 million records. It consists of benign and four categories of attacks, but most of the dataset contains packets of the DoS and DDoS types. More information about the dataset can be found in [29]. For the sake of training and validating machine learning models through a binary classification, attack instances in the dataset are labelled with a '1', whereas benign (normal) instances are labelled with a '0'. For multiclass classification, we have four major attack categories represented as DoS, DDoS, reconnaissance, and information theft, plus a benign. Each of the attack categories also has sub-categories. Both DoS and DDoS have TCP, UDP, and HTTP; reconnaissance has OS fingerprints and service scanning; and information theft has key logging and data exfiltration. Fig. 5 visualizes the distribution of the normal and attack classes in the Bot-IoT dataset for both classifications. As attack classes make up a considerable percentage of the data set while regular traffic is significantly underrepresented, this highlights the extreme imbalance in the dataset. It is important to note that, as suggested by [27], we extracted 5% of the initial data using select MySQL queries because the produced dataset is enormous, with more than 72.000.000 records and 16.7 GB for CSV and 69.3 GB for pcap. Koroniotis et al. [28] suggested the 5% subset as a more manageable and condensed form. It comprises about 3.6 million records totaling about 1.07GB. When it comes to the attack category, it is a representative sample of the entire collection.

B. Normalization

Normalization aims to scale down features to a similar scale. This is accomplished by using (1) to scale down the dataset so that the normalized data falls between 0 and 1 without affecting the normalcy of the data's behaviour.

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (1)$$

where x is the original feature value, $\hat{x}$ is the normalized feature value, and min(x) and max(x) are the minimum and maximum values of each original feature, respectively. Since both binary and multiclass classifications were considered, two data frames were created, one for each of them. The “label” attribute, which is classified into “normal” and “abnormal”, was encoded using the label encoder for binary classification data frames and is represented as “0” for normal and “1” for abnormal. For the multiclass classification data frame, the “attack_cat” attribute, categorized into the eight operational states (for the CICIDS2017 dataset) and five (BoT-IoT), is encoded using labelEncoder() and also one-hot encoded.

C. Feature ranking

Feature extraction was done for both binary classification and multiclass classification, which had 80 and 88 columns, respectively, including the class label for the CICIDS2017 dataset. A correlation matrix is used as a tool for feature extraction, whereby a subset of features (i.e., variables) are chosen from a larger set of features in a dataset to enhance the performance of a machine learning model. A correlation coefficient for each pair of features in the dataset was estimated using the Pearson correlation coefficient (PCC) illustrated in (2).

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})}} \qquad (2)$$

In (2), r_{xy} denotes the correlation coefficient between variables x and y, x_i and y_i denote the values of the x-variable and y-variable in a dataset, $\bar{x}$ and $\bar{y}$ denotes the mean of the values of the x and y variables while n is the size of the dataset.

In this study, the PCC was adopted because it’s very fast and one of the widely used for estimating the relationship between variables. The idea is that the features with the lowest correlation coefficient would introduce less ambiguity in the dataset. A correlation matrix was created and visualized as a heatmap, which highlights the pairs of features that are highly correlated with each other. A correlation matrix that shows the correlation coefficient between the features of the data frame used for binary classification and that used for multi-class classification is shown in Fig. 6, for the CICIDS2017 dataset. The correlation coefficient has a value that ranges from -1 to 1. If the value is -1, then there is a perfect negative correlation, meaning that as one variable increases, the other decreases. When both variables increase as one does, there is a perfect positive correlation with a value of 1. In the absence of a linear relationship between the two variables, a value of 0 denotes no correlation.

The attributes that have less than a 0.5 correlation coefficient with the target attribute were selected, while the rest were dropped. After the feature extraction, the data frame for binary data contains 20 attributes and a class label, whereas the multiclass data frame has 22 attributes and a class label for the CICIDS2017 dataset. For the Bot-IoT dataset, the 5% subset has the most features of any processed set or subset of Bot-IoT, with 43 independent features and 3 dependent features. Five CSV files, each with a header row with feature names, make up the 5% subset. There are 46 features present in this dataset. After feature ranking, we have a total of 19 features, which consist of flgs, proto, pkts, bytes, state, dur, mean, stddev, sum, min, max, spkts, dpkts, sbytes, dbytes, rate, srate, drate, and class. These features are used to train the classifiers.

D. Data splitting

In this stage, the normalized data is divided into two parts. This was achieved by randomly dividing the set into training and test sets. In this study, 70% of the data was used for training and 30% for testing. The training set is used to train the model, while the testing set is used to evaluate the model's prediction.

E. Classification and prediction

Classification uses a dataset or set of observations to categorize fresh data into one of several categories. At this stage, we develop three machine learning classifiers: MLP, k-NN, and DNN.

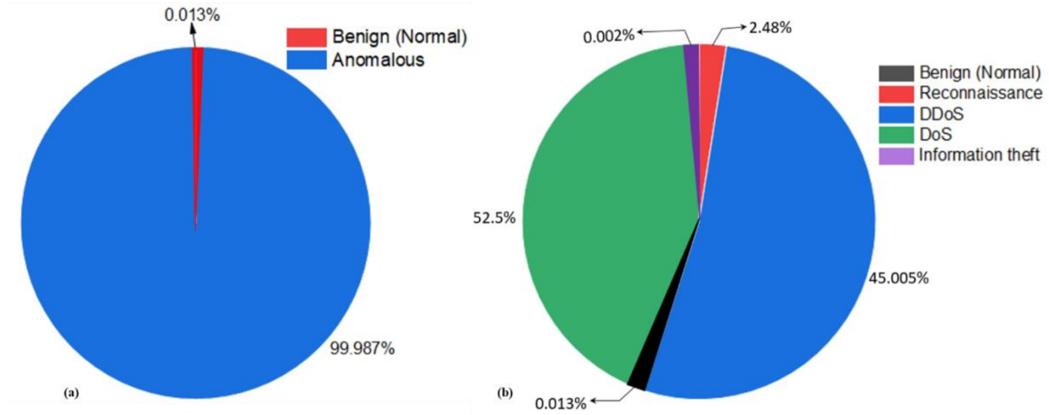

Fig. 5. *Target data distribution in the CICIDS2017 dataset for (a) binary classification, (b)multiclass classification.*

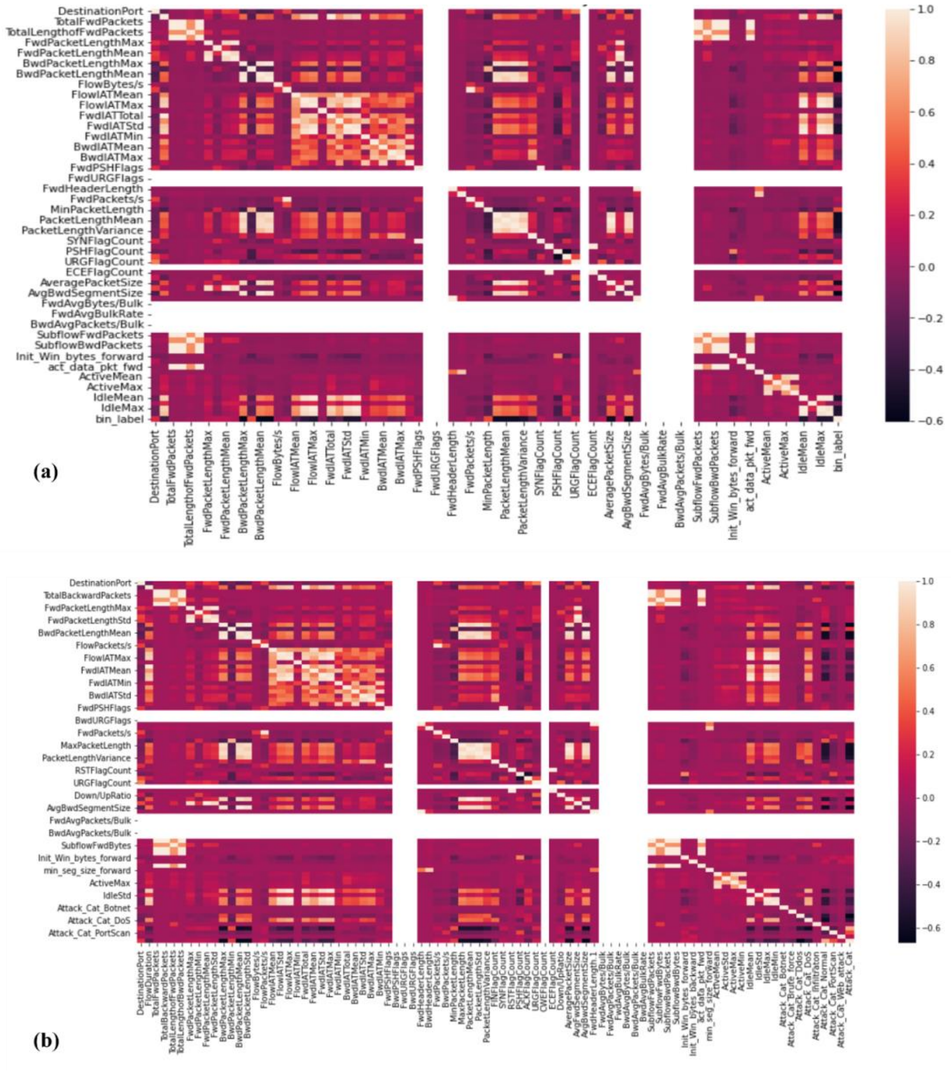

Fig. 6. *Heatmap of feature correlation matrix for the CICIDS2017 dataset for (a) binary classification (b) multiclass classification.*

70% of the dataset was applied to train these classifiers, which are used to classify and predict DDoS attacks in the dataset.

F. Experimental setup

The simulations were carried out in Python 3.0. The data processing and evaluation were implemented by extension packages including NumPy, Pandas, and Scikit-learn. For DNN implementation, the Keras library was used, while Scikit-learn was used for MLP and k-NN. The MLP and DNN have three hidden layers with ReLU activation function. The output layer is made up of a single unit that uses the Softmax activation function for multiclass classification and the sigmoid activation function for binary classification. The loss function used was cross-entropy. This involved using categorical cross-entropy for multiclass classification and binary cross-entropy for binary classification. Finally, we used the Adam optimizer to update network weights, 100 epochs, and batch size of 128, and a learning rate of 0.0011. For the k-NN classifier, we set *k* = 10. Also, the Euclidean distance is a widely used distance metric and was adopted in this study. Each of the classifiers was trained and tested to predict DDoS attacks, considering both binary and multiclass classifications. In binary classification, the value of the attack is predicted as either normal or abnormal. The results of the model predictions were compared against the actual values of the particular entry. All experiments were carried out on a HP ProBook running Windows 10, a 64-bit operating system. The processor was an Intel Core i7 3.60 GHz CPU equipped with 8 GB of RAM.

G. *Performance evaluation*

The performance of the classifiers was assessed using accuracy, precision, recall and f1-score metrics.
Accuracy: This metric determines the percentage of accurate predictions across all the cases considered. The detection accuracy (D_A) is expressed using (3).

$$D_A = \frac{T_P + T_N}{T_P + F_P + T_N + F_N} \tag{3}$$

In (3), T_N stands for true negative and signifies the number of instances of normal traffic that the detection method correctly classifies as belonging to the normal class, F_N denotes false negative indicating the number of instances of attack traffic that are identified as normal traffic, and true positives (T_P) signifies number of attack instances accurately categorized, whereas false positives (F_P) signifies number of instances of normal traffic that are wrongly classified as attack instances.
Precision: It is an estimate of the proportion of positive patterns in a positive class that are successfully predicted out of all the anticipated patterns. The precision (*P*) is expressed as

$$P = \frac{T_P}{T_P + F_P} \tag{4}$$

Recall (R): It measures the proportion of positive patterns that are classified properly. It is expressed as

$$R = \frac{T_P}{T_P + F_N} \tag{5}$$

F1-score: This is a popular metrics for imbalance data. It denotes the harmonic mean between *R* and *P* results. It is expressed as

$$F1 - score = \frac{2 \times R \times P}{R + P} \tag{6}$$

This means that a model will have a high F1-score if both the precision and recall are high and vice versa.

IV. Results And Discussions

In this section, we show the performance of the models when applied to the CICIDS2017, and Bot-IoT datasets. Thus, we discuss the evaluation results of the models under each dataset.

A. *CICIDS2017 dataset*

Fig. 7 shows the precision results of the three models for both binary and multiclass classifications using the CICIDS2017 dataset. It is observed that the three models have excellent performance for multiclass scenarios, whereas during binary classification, only the DNN model's performance is degraded. Under this classification, it records the least precision out of the three models, while the precision result for k-NN clearly shows its superiority above others. Nevertheless, the precision results of the three models improve during multiclass classification, where the DNN shows a significant improvement of 6.12%. For the multiclass case, the precision result of both k-NN (99.991%) and DNN (99.985%) is slightly better than that of MLP (99.984%), although the improvement can be considered insignificant with only a difference of less than 0.01% observed.

Fig. 8 shows the accuracy of the three models during training and testing for both binary (Fig. 8(a)) and multiclass (Fig. 8(b)) classifications for the CICIDS2017 dataset. As can be noticed in Fig. 8(a), k-NN is the most performing model with an accuracy of 99.649%, while MLP is second on the list with 97.869% accuracy during training. During testing, both models recorded an accuracy of 99.547% and 97.881%, respectively. Out of the three models, the DNN has the least accuracy during binary classification. However, when multiclass classification is considered, as shown in Fig. 8(b), there is a significant improvement in the accuracy performance of the DNN. Its accuracy of 99.987% and 99.985% is slightly greater than that of MLP and a bit lower than the accuracy recorded by the k-NN (99.994% and 99.991%) during both training and testing. The DNN achieves about 6% improvement in accuracy when multiclass classification is considered. For k-NN and MLP, only 0.35% and 2.12% improvement over the accuracy produced from binary classification were observed when multiclass classification was considered, respectively. Among the three algorithms, k-NN records the best accuracy (although just a slight improvement above MLP) during both classification scenarios. In this analysis, the DNN doesn't seem to be a good model for DDoS prediction when binary classification is considered.

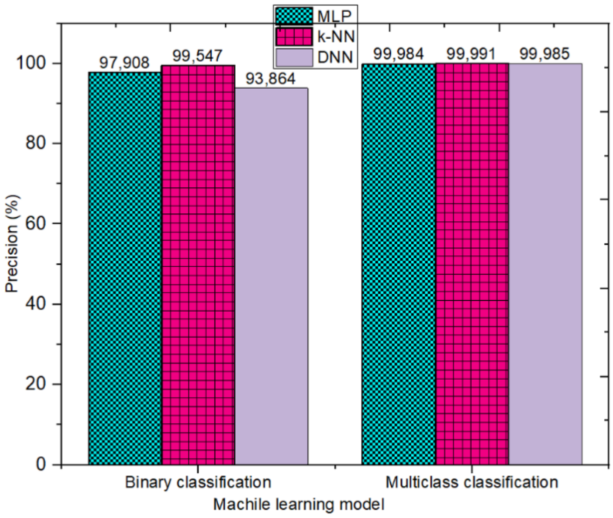

Fig. 7. *Precision of the three models using the CICIDS2017 dataset for both binary and multiclass classification.*

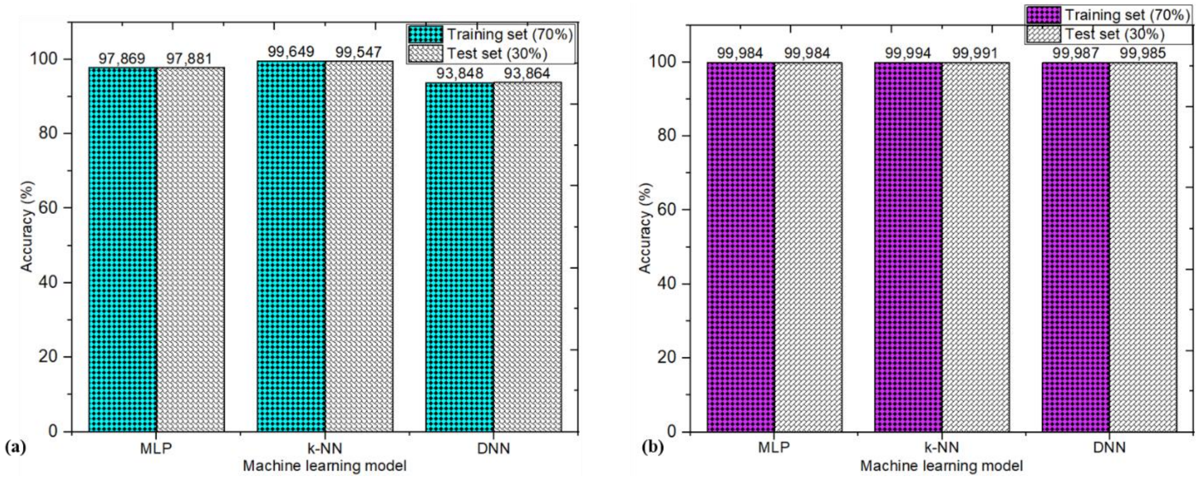

Fig. 8. *Classification accuracy of the three models using the CICIDS2017 dataset for both binary and multiclass classification.*

In Fig. 9, the result of the recall for the three models for both binary and multiclass classifications using the CICIDS2017 dataset is presented. Similar to the results presented in Fig. 7, the three models performed better for multiclass scenarios only for DNN, whose recall performance is reduced in binary classification. Under this classification, it records the lowest recall value out of the three models. Similar to Fig. 7, its performance was improved by 6.12% in the multiclass scenario.

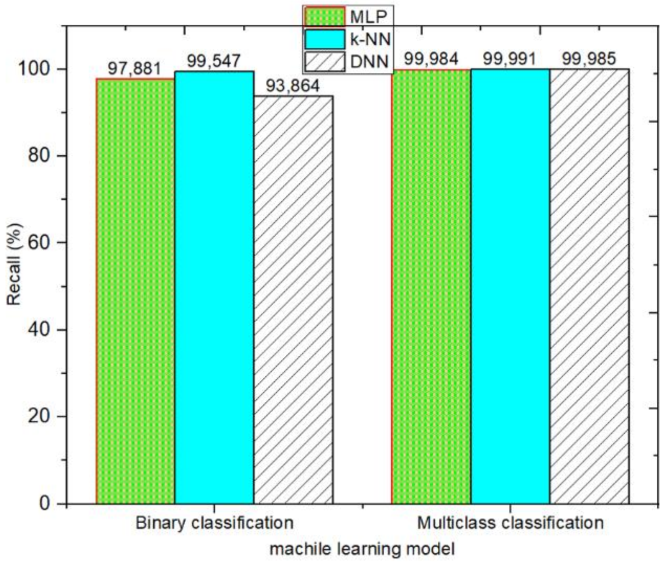

Fig. 9. *Recall of the three models using the CICIDS2017 dataset for both binary and multiclass classification.*

In Fig. 10, the result of the F1-score for the three models for both binary and multiclass classifications using the CICIDS2017 dataset is presented. This also supports the results presented in Figs. 8 and 9. For multiclass classification, these results revealed that the performance of the DNN is superb. However, during binary classification, its performance is slightly affected and may not be proposed for prediction in binary classification. Fig. 11 illustrates the ROC curve for the three classifiers using the CICIDS2017 dataset. By observing Fig. 11 and comparing the ROC curve, we can deduce that k-NN has better capability to distinguish between attack instances and normal ones with 99.523% of the AUC, while MLP also records a superb performance with 97.685% of the AUC. This result shows that k-NN can excellently predict the attack in this dataset. The capability of the DNN classifier in this regard slightly reduces with a record 92.868% AUC.

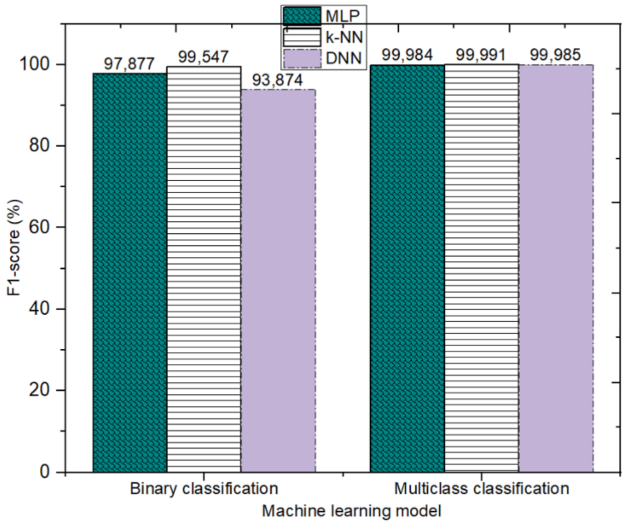

Fig. 10. *F1-score of the three models using the CICIDS2017 dataset for both binary and multiclass classification.*

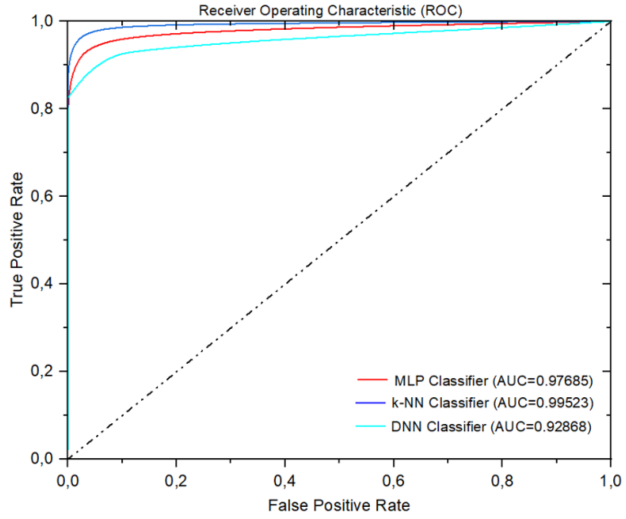

Fig. 11. *ROC curve of the three models using the CICIDS2017 dataset for both binary and multiclass classification.*

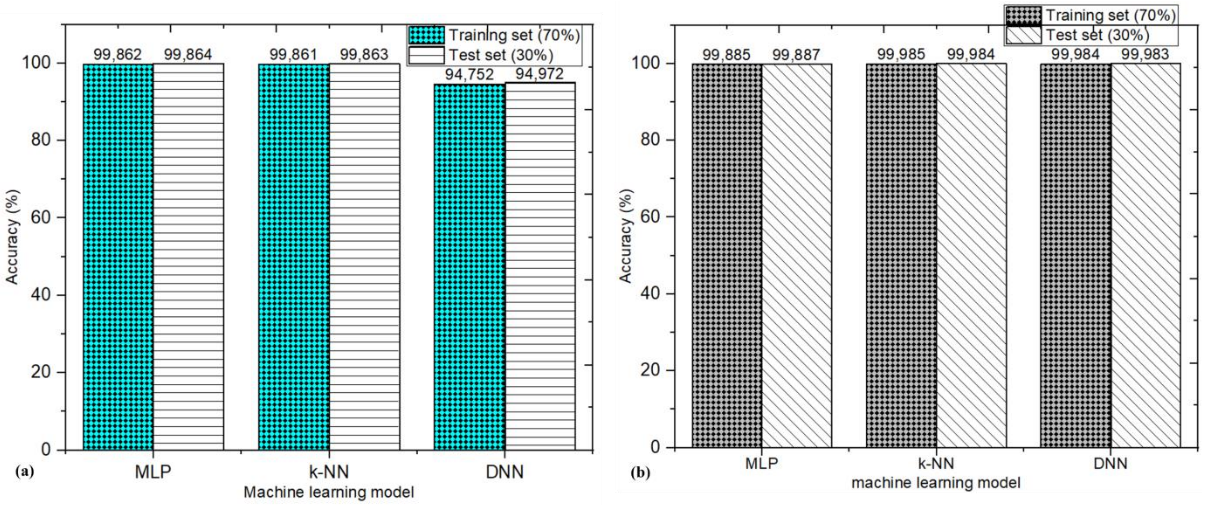

Fig. 12. *Classification accuracy of the three models using the BoT-IoT dataset for both binary and multiclass classification.*

B. BoT-IoT dataset

Fig. 12 shows the accuracy of these models during training and testing for both binary (Fig. 12(a)) and multiclass (Fig. 12(b)) classifications. As can be noticed in Fig. 12(a), both MLP and k-NN are the most accurate models with an accuracy of 99.862% and 99.861% during training, while DNN records the least performance with 94.752% accuracy. Out of the three models, the DNN has the least accuracy during binary classification. However, when multiclass classification is considered, as shown in Fig. 12(b), there is a significant improvement in the accuracy performance of the DNN. An improvement of 5.23% in accuracy is observed when multiclass classification is considered. During both training and testing, its accuracy of 99.984% and 99.983% is slightly lower than that of MLP and k-NN, though only a less significant difference can be observed. The results obtained for this dataset also revealed that the DNN doesn't seem to be a good model for DDoS prediction when binary classification is considered.

Fig. 13 shows the precision results of the three models for both binary and multiclass classifications using the Bot-IoT dataset. It is observed that the three models have excellent performance for multiclass scenarios, whereas during binary classification, the performance of DNN is reduced. Under this classification, it records the least precision. Similar to the results presented in Fig. 12, the DNN achieves about 5% improvement in precision when multiclass classification is considered. However, the precision results recorded by MLP and k-NN slightly reduce with 0.02% for MLP and 0.08% for k-NN when multiclass classification is considered. This may also be because the dataset is highly imbalanced, which may affect the performance during a multiclass scenario. This does not significantly degrade their precision results. For this dataset, the precision result of the k-NN is relatively better than that of MLP and DNN considering multiclass classification, while that of DNN is slightly better than the precision recorded by the MLP model. However, this analysis also revealed that the DNN seems not to be a good model for DDoS prediction when binary classification is considered.

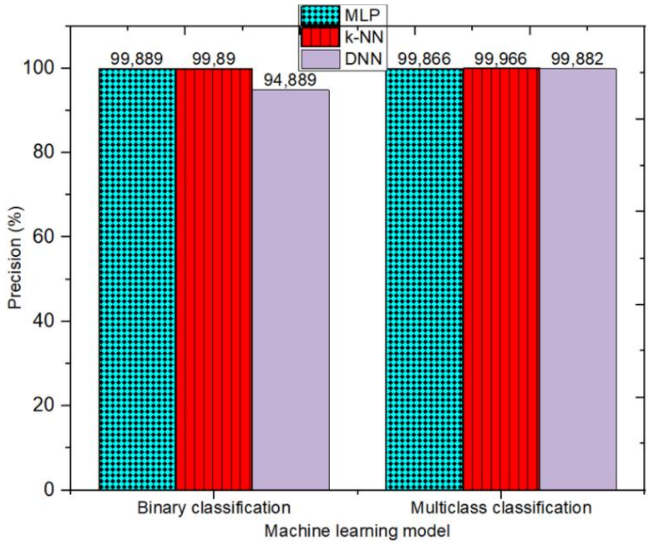

Fig. 13. *Precision of the three models using the BoT-IoT dataset for both binary and multiclass classification.*

The recall for the three models for both binary and multiclass classifications using the Bot-IoT dataset is shown in Fig. 14. Like the results shown in Fig. 13, the three models performed better in multiclass scenarios only for DNN, whose recall performance is reduced in binary classification. However, in a multiclass scenario, the performance of DNN was enhanced by 4.96%. The k-NN, with a recall value of 99.967%, and the DNN (99.881%), have relatively better recall values in a multiclass scenario than the MLP (99.867%). Even though there isn't much of a difference between them in this instance, the k-NN nevertheless records the best precision results in a multiclass scenario, although in a binary classification, MLP has a slight advantage over the k-NN in terms of the recall value. In Fig. 15, the result of the F1-score for the three models for both binary and multiclass classifications is presented. This also supports the ones presented in Fig. 13 and Fig. 14. This result further demonstrates that the DNN should not be considered for attack prediction when binary classification is used.

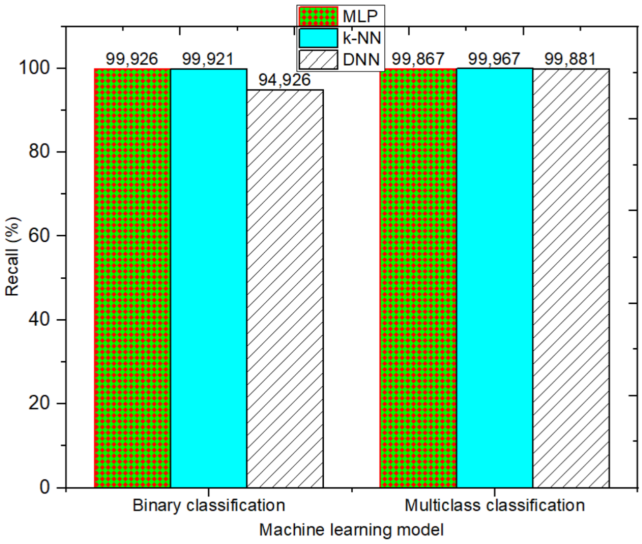

Fig. 14. *Recall of the three models using the BoT-IoT dataset for both binary and multiclass classification.*

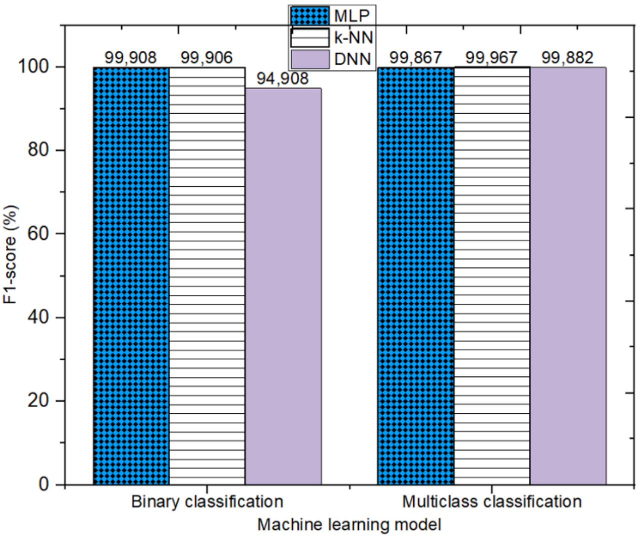

Fig. 15. *F1-score of the three models using the BoT-IoT dataset for both binary and multiclass classification.*

Fig. 16 illustrates the ROC curve for the three classifiers using the Bot-IoT dataset.

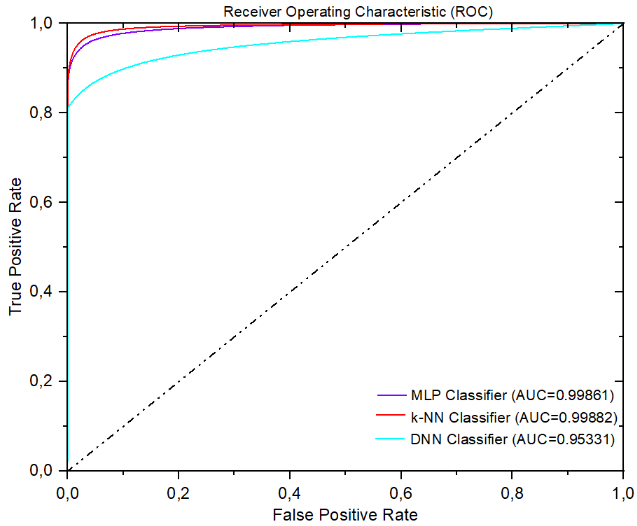

Fig. 16. *ROC curve of the three models using the BoT-IoT dataset for both binary and multiclass classification.*

By observing Fig. 16 and comparing the ROC curve, it is can be seen that both k-NN and MLP have better capability to distinguish between attack instances and normal ones with 99.888% and 99.861% of the AUC, respectively. Thus, both k-NN and the MLP can excellently predict the attack in this dataset. The capability of the DNN classifier in this regard slightly reduces with 95.33% of the AUC.

V. Conclusion And Future Studies

With technology advancing so quickly, the IoT is becoming increasingly exposed and a target for attackers. DDoS attacks differ from other types of attacks in that they are challenging to prevent because they show no evidence of device failure. Many strategies have been put forth for timely detection of this attack. Among others, machine and deep learning approaches are major participants due to their ability to learn traffic features and have been successful in accurately predicting attacks. In this paper, three classifiers (MLP, k-NN, and DNN) were developed and their potential for attack detection examined, considering both binary and multiclass classifications. The assessment findings allow us to demonstrate that multiclass classification can greatly increase the detection accuracy of these models on a balanced or nearly balanced dataset. However, for highly imbalanced datasets, the performance of the models is observed to be reduced. While the detection accuracy of these models is also good for binary classification, it has been noticed that the DNN doesn’t seem to be a good model for DDoS prediction when binary classification is considered. Overall, the results are encouraging and may open the door for the future development of an ensemble classifier using hybrid MLP and k-NN as robust detection models that will be applied to other IoT-based datasets that contain new variants of DDoS attacks. The ensemble classifier will be further incorporated into the attack mitigation algorithm to have a robust intrusion detection and mitigation system.

References

[1] Statista, “Internet of things (IoT) connected devices installed base worldwide from 2015 to 2025,” [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. [Accessed: 25 April 2023].

[2] T.A. Ahanger, A. Aljumah, and M. Atiquzzaman, “State-of-the-art survey of artificial intelligent techniques for IoT security,” Computer Networks, *vol.* 2022, pp. 1–30, 2022.

[3] Ahmad, M.S. Niazy, R.A. Ziar, and S. Khan, “Survey on IoT: security threats and applications,” Journal of Robotics and Control, vol. 2, pp. 42–46, 2021.

[4] I. Ahmad, M.S. Niazy, R.A. Ziar, and S. Khan, “Survey on IoT: security threats and applications,” Journal of Robotics and Control, vol. 2, pp. 42–46, 2021.

[5] S. Kothari, S. Joshi, and I. Tidke, “An Exhaustive comparison and analysis of botnet attacks for smartphones,” International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 2, pp.599-619, 2024.

[6] K.B. Adedeji, A.M. Abu-Mahfouz, and A.M. Kurien, “DDoS attack and detection methods in internet-enabled networks: Concept, research

perspectives, and challenges," Journal of Sensor and Actuator Networks, vol. 12, no. 4, pp. 1-57, 2023.

[7] O. Yoachimik, and J. Pacheco, "DDoS threat report for 2023 Q4," Online. Available from https://www.blog.cloudflare.com/ddos-threat-report-2023-q4. [Accessed: 01/05/2024].

[8] B. Susilo, and R.F. Sari, "Intrusion detection in IoT networks using deep learning algorithm," Information, vol. 11, pp. 1–11, 2020.

[9] A.A. Abdulrahman, and M.K. Ibrahem, "Evaluation of DDoS attacks detection in a new intrusion dataset based on classification algorithms," Iraqi Journal of Information and Communication Technology, vol. 1, pp. 49–55, 2018.

[10] M.S. Elsayed, S.N. Le-Khac, A.S. Dev, and A.D. Jurcut, "DDoSNet: A deep-learning model for detecting network attacks," In: Proceedings of the 2020 IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks, Cork, Ireland, 31 August–3 September 2020, pp. 391–396.

[11] R.B. Adhao, and V.K. Pachghare, "Performance-based feature selection using decision tree," In: Proceedings of the IEEE International Conference on Innovative Trends and Advances in Engineering and Technology, Shegaon, India, 27–28 December 2019; pp. 135–138.

[12] K. Kurniabudi, D. Stiawan, D. Darmawijoyo, M.Y.B. Idris, B. Kerim, and R. Budiarto, "Important features of CICIDS-2017 dataset for anomaly detection in high dimension and imbalanced class dataset," Indonesian Journal of Electrical Engineering and Informatics, vol. 9, pp.498–511, 2021.

[13] M.J. Santos-Neto, J.L. Bordim, E.A. Alchieri, E. and E. Ishikawa, "DDoS attack detection in SDN: Enhancing entropy-based detection with machine learning," Concurrency and Computation: Practice and Experience, p.e8021, 2024.

[14] R. Fuladi, T. Baykas, and E. Anarim, "The use of statistical features for low-rate denial-of-service attack detection," Annals of Telecommunications, pp.1-13, 2024.

[15] V.A. Shirsath, M.M. Chandane, C. Lal, and M. Conti, "SYNTROPY: TCP SYN DDoS attack detection for Software Defined Network based on Rényi entropy," Computer Networks, vol. 244, p.110327, 2024.

[16] A.W. Moore, and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," In Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modelling of Computer Systems, Banff, AB, Canada, 6–10 June 2005, pp. 50–60.

[17] F.S.D. Lima Filho, F.A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, and L.F. Silveira, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," Secur. Commun. Netw, vol. 2019, p. 1574749, 2019.

[18] A. Alshamrani, A. Chowdhary, S. Pisharody, D. Lu, and D. Huang, "A defense system for defeating DDoS attacks in SDN based networks," In: Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access, Miami, FL, USA, 21–25 November 2017, pp. 83–92.

[19] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detection method based on SVM in software defined network," Secur. Commun. Netw., vol. 2018, 9804061, 2018.

[20] Y.W. Chen, J.P. Sheu, Y.C. Kuo, and V. Van Cuong, "Design and implementation of IoT DDoS attacks detection system based on machine learning," In: Proceedings of the IEEE European Conference on Networks and Communications, Dubrovnik, Croatia, 15–18 June 2020, pp. 122–127.

[21] A. Mihoub, O.B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques," Comput. Electr. Eng., vol. 98, p. 107716, 2022.

[22] R.J. Alzahrani, and A. Alzahrani, "Security analysis of DDoS attacks using machine learning algorithms in networks traffic," Electronics, vol. 10, p. 2919, 2021.

[23] R. Santos, D. Souza, W. Santo, and A. Ribeiro, "Machine learning algorithms to detect DDoS attacks in SDN," Concurr. Comput. Pract. Exp., vol. 32, e5402, 2020.

[24] A. Chartuni, and J. Márquez, "Multi-classifier of DDoS attacks in computer networks built on neural networks," Applied Sciences, vol. 11, pp. 1–15, 2021.

[25] M. Aamir, and S.M.A. Zaidi, "DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation," International Journal of Information Security, vol. 18, pp. 761–785, 2019.

[26] M.H. Aysa, A.A. Ibrahim, and A.H. Mohammed, "IoT DDoS attack detection using machine learning," In: Proceedings of the IEEE 4th International Symposium on Multidisciplinary Studies and Innovative Technologies, Istanbul, Turkey, 22–24 October, 2020, pp. 1–7.

[27] Canadian Institute for Cybersecurity 2017. Intrusion Detection Evaluation Dataset (CIC-IDS2017). [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html. [Accessed: Feb. 11, 2023].

[28] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[29] Y. Xu, H. Sun, F. Xiang, and Z. Sun, "Efficient DDoS detection based on K-FKNN in software defined networks," IEEE Access, vol. 7, pp. 160536–160545, 2019.