Exploring the Potential of Artificial Intelligence Model to Detect Distributed Denial of Service Attacks

Prashant Kumar[1,2], Chitra Kushwaha[2], Dinesh Kumar Yadav[3], Solomon Raju Kota[2]
[1]Bennett University (The Times Group), Plot No 8-11, Techzone 2, Greater Noida-201310, Uttar Pradesh, India
[2] CSIR - National Aerospace Laboratories, Bengaluru-560017, Karnataka, India
[3] Netaji Subhas University of Technology, Delhi, India
{prashant.mnnit10, chitra.kushwaha2005, dinesh.nsut21, kotasolomonraju}@gmail.com

Abstract. DDoS attacks, which fall under the category of cybercrime in the contemporary scene, are simple to launch yet pose enormous consequences. DDoS attacks are classified into volumetric and exploitation-based kinds, which include denial of service, LDAP, MSSQL, UDPLag, Syn, NetBIOS, UDP, and others. To detect these attacks, numerous detection methods and machine learning techniques have been deployed. Current research focuses on improving machine learning approaches, with classifiers such as Decision Trees, Support Vector Machine (SVM), and Logistic Regression displaying improved outcomes. In certain cases, algorithms were coupled to attain greater accuracy. However, dealing with massive amounts of network data presents difficulties, necessitating significant execution time and resources for sustaining hybrid models. This study investigates a deep learning model, Deep Neural Network (DNN), to effectively forecast DDoS attacks. The investigation makes use of the CICDDoS2019 benchmark dataset, which has 88 features from which a subset of 22 important features is extracted and deep learning model is applied. The proposed model's results show a significant improvement over existing techniques in this domain involving machine learning models and data mining techniques. While it is not feasible to totally eliminate the possibility of DDoS attacks, implementing the measures outlined here can help minimize these attacks to some extent. Furthermore, it enables servers to prioritize legitimate user requests rather than becoming overwhelmed by requests from illegal sources. This implementation delivers testing accuracy of 99.39%.

Keywords: DDoS, Network Security, Cyber Security, Machine Learning, Deep Learning, DoS

1 Introduction

In the modern world internet plays a crucial role, benefitting various fields such as communication, education, business and shopping. However, alongside its advantages there has been rise in internet related crimes, including disseminations of misinformation, hacking and various types of attacks. If the services, we rely on were suddenly unavailable. This type of attack is known as a DoS attack. When such an attack is conducted using a single machine it is referred to as single DoS attack. On the other hand, when multiple machines are compromised and used in attack it is called a DDoS attack. These attacks are basically into three categories namely Volumetric attack, Protocol based attack, Application layer attack [23]. Attackers

IACIDS 2023, November 23-25, Lavasa, India

DOI 10.4108/eai.23-11-2023.2343336

inundate service providers with a large volume of requests, consuming network resources for an extended period. Thereby preventing legitimate user requests from being processed.
DDoS attacks have indeed been prevalent over the past two decades, targeting various high-profile organizations and causing significant disruptions. In 2000, Yahoo, eBay, Amazon websites experienced DDoS attack which impacted their availability and disrupted their services. In 2001, Attackers utilized DNS servers as reflectors to launch a DDoS attack on Register.com, a domain name registrar. The attack aimed to overwhelm the target's resources and disrupt its operations and many more are there. But recently in February 2020, AWS (Amazon Web Services) detected and mitigates a DDoS attack that reached a record breaking 2.3 Tbps in terms of bit rate. It is considered as the largest DDoS attack in history. These incidents highlight the scale and impact of DDoS attacks on various sectors, emphasizing the need for roust defense mechanisms and timely detection to mitigate their effects.
The objective of this research paper is to analyze the recent DDoS dataset using Deep learning approaches specifically DNN model because traditional statistical and machine learning methods including some deep learning techniques fails to tackle the arisen problem [25][26]. Experimental results demonstrate that proposed model (99.39 %) is having better accuracy than traditional models which is significant good accuracy.
The contribution of the paper includes-

- Selecting an appropriate benchmark dataset and performing preprocessing on the dataset.
- Designing and develop a Deep Neural Network model.
- Implementing the model on the latest dataset "CICDDoS2019" to achieve efficient accuracy.
- Providing a detailed report on the outcome of a model.

The structure of the paper consists of five sections: Section I provides an overview of DoS and DDoS attack, Section II explores previous studies and research related to this, Section III describes the methodology, Section IV presents the findings and results obtained, Section V summarizes the contributions and conclusions of the paper.

2 Literature Review

In 2020, [1] focused on using Named Data Networking (NDN) to detect and mitigate DDoS attacks. They utilized NDN routers with data structures like forwarding Information Base (FIB), Content Store (CS), and pending Interest Table (PIT) to identify DDoS attacks through collaboration with a centralized controller. This approach improved the accuracy of detecting fake name prefixes compared to previous methods. In 2019, [3] proposed a model to detect HTTP flooding attacks using the Susceptible Infective Susceptible (SIS) mathematical model. This model measured various user attributes to identify abnormal behavior and effectively detect HTTP flooding attacks.
In recent years, researchers have focused on machine learning based approached to improve the detection and mitigation of DDoS attacks. In 2022, [4] proposed a solution combining radial basis function with cuckoo search algorithm was proposed to detect application layer, achieving lower false positive rates. Addressing the absence of time-based features, in 2021, [5] utilized a model of 25 time-based features to detect DDoS attack types using binary and multiclass classification techniques. Another approach in 2021, [6] employed Artificial Neural Network to detect various types of DDoS attacks, including application layer, network layer, and transport

layer attacks, while considering time and space complexity for improved efficiency. In 2021, [7] has discussed different DDoS defense system based on ML techniques in virtualized network, cloud computing, software defined network, and IoT environments. In 2020, [8] utilized Reinforcement learning to detect and mitigate DDoS attacks, resulting in increased throughput of legitimate TCP traffic. IN 2020, [9] employed factorization machine-based approaches in software defined networks and achieved 97.85% accuracy in DDoS attack detection. In 2020, [10] introduced a solution to mitigate suspicious activities using machine learning algorithms by evaluating the model's efficiency through feature elimination and considering relevant features from successful DDoS detection works. In 2020, [11] worked on KDD-CUP dataset to perform K-NN, ID3, Naïve Bayes and C4.5 algorithms to compare the all obtained results those are experimentally verified. In 2020, [12] proposed methods to detect and exploring the DDoS using K-NN based on machine learning in software defined network that showed proposed method detects better in comparison of other methods. In 2019, [13] Various classification algorithms, such as Naive Bayes, Decision Tree and Random Forest have been proposing classification-based approaches have been explored, including the use of Robust Principal Component Analysis [14] [15]. Lastly, the utilization of Hadoop architecture and time series analysis has been investigated to increase processing speed, block suspicious IP addresses, and detect DDoS attacks effectively [16]. These machines learning based approaches offer promising potential for enhancing the detection and mitigation of DDoS attacks, providing more accurate and efficient results compared to traditional methods. Deep learning-based approaches have emerged as a more effective alternative to traditional machine learning techniques in detecting and mitigating DDoS attacks. In 2022, a lightweight deep learning models was proposed to analyze live traffic volume in resource-constrained environments, demonstrating its sustainability for DDoS detection and achieving high accuracy [17]. A hybrid approach combining Auto Encoder (AE) for feature selection and multilayer perceptron Network (MLP) for classification, called AE-MLP, was introduced in 2021. This approach achieved 98% accuracy on the CICDDoS2019 dataset [18]. In the same year, a model using machine learning and deep learning techniques was proposed to detect application layer and transport layer DDoS attack in software defined architecture. It achieved high accuracy on the CICDDoS2017 and CICDDoS2019 datasets using SVM, K-NN, MLP, CNN and GRU [19]. Using a deep convolutional neural network, a methodology was proposed in 2020 to detect DDoS attacks in software defined networks, demonstrating improved accuracy compared to existing detection approaches [21]. In 2018, a deep learning model was developed for Open Flow-Based software- defined network environments, achieving significantly better results than conventional machine learning methods [22].

It can be clearly seen that while deep learning models have shown considerable improvements in performance, there is still room for further enhancement in terms of feature retention to increase overall accuracy. Therefore, this paper explores the use of DNN for multiclass classification, which have demonstrated high accuracy in the detection of DDoS attacks.

3 Methodology

We have used subset of CICDDOS2019 dataset to train our model that contains 10 files that associated the data of LDAP, MSSQL, NETBIOS, DrDoS_NTP, DrDOS_SNMP, Port map, UDP, Syn, etc. We have concatenated the selected files (Table 1) to apply. After merging, this

dataset is having 7773039 rows and 88 columns. The dataset contains 88 features out of which we have to select only useful features.

Table 1: CICDDoS2019 Dataset Bifurcation

S.No.	Type of DDoS	No of Samples	Total Samples	Total Samples after Preprocessing	Training Samples	Testing Samples
1.	MySQL	1446845	7773039	7538825	5277177	2261648
2.	DrDoS_SNMP	1289967				
3.	DrDoS_DNS	1267758				
4.	Syn	1071104				
5.	UDP	938633				
6.	NetBIOS	914519				
7.	LDAP	478653				
8.	DrDoS_NTP	300616				
9.	Portmap	46776				
10.	BENIGN	18168				

Data Preprocessing: This step is used to clean the data so firstly we have to remove unnecessary and noisy data. Then we have found the values which are “infinity” or “missing places” type value. Here we have two options to handle this type of data. One thing is we can fill these missing values by calculating the mean of that column or delete that row if the dataset size is very large. In our dataset there is no missing value and the rows contain infinity values are deleted (Table 1). We also delete 3 columns from the dataset (Unnamed: 0, SimilarHTTP, and Label) (Figure 1). After this task the next thing is to convert the all-categorical value into numerical value so in this paper, we have used Label Encoder from Scikit-Learn library. It converts categorical values into integer value irrespective of their rankings. In our dataset many variables contain categorical values so we have applied the label encoding on “Label”, “FlowID”, “SourceIP”, “DestinationIP”, and “Timestamp” column (Figure 3). Then the most important thing is to make sure the data is of which datatypes. After converting all the dataset into one datatype we have selected some handful useful features (Figure 2). But the question is that how will we identify that which feature is more useful and which is less so that our model can be trained for accurate results. Here we have used ensemble feature selection method namely Extra Tree Classifier that combines multiple different feature selection methods, taking into account their strengths and create an optimal best subset. Then we have calculated feature importance that calculate a score for all the input feature for a given model. This score represents the importance of each feature. In this higher value of feature importance represents that the specific feature will have a larger effect on the model that is being used to predict a certain variable. Then we have selected 22 features out of 85 features excluding “Label” feature, those are having higher importance. After selecting 22 important features (Figure 2 and 3) we have standardized our dataset. To standardize our dataset, we have used standard scaler from scikit learn library and divided the dataset into training and testing data in the ratio of 70 and 30 (Table 1).

```
Index(['Flow ID', ' Source IP', ' Source Port', ' Destination IP',
       ' Destination Port', ' Protocol', ' Timestamp', ' Flow Duration',
       ' Total Fwd Packets', ' Total Backward Packets',
       'Total Length of Fwd Packets', ' Total Length of Bwd Packets',
       ' Fwd Packet Length Max', ' Fwd Packet Length Min',
       ' Fwd Packet Length Mean', ' Fwd Packet Length Std',
       'Bwd Packet Length Max', ' Bwd Packet Length Min',
       ' Bwd Packet Length Mean', ' Bwd Packet Length Std', 'Flow Bytes/s',
       ' Flow Packets/s', ' Flow IAT Mean', ' Flow IAT Std', ' Flow IAT Max',
       ' Flow IAT Min', 'Fwd IAT Total', ' Fwd IAT Mean', ' Fwd IAT Std',
       ' Fwd IAT Max', ' Fwd IAT Min', 'Bwd IAT Total', ' Bwd IAT Mean',
       ' Bwd IAT Std', ' Bwd IAT Max', ' Bwd IAT Min', 'Fwd PSH Flags',
       ' Bwd PSH Flags', ' Fwd URG Flags', ' Bwd URG Flags',
       ' Fwd Header Length', ' Bwd Header Length', 'Fwd Packets/s',
       ' Bwd Packets/s', ' Min Packet Length', ' Max Packet Length',
       ' Packet Length Mean', ' Packet Length Std', ' Packet Length Variance',
       'FIN Flag Count', ' SYN Flag Count', ' RST Flag Count',
       ' PSH Flag Count', ' ACK Flag Count', ' URG Flag Count',
       ' CWE Flag Count', ' ECE Flag Count', ' Down/Up Ratio',
       ' Average Packet Size', ' Avg Fwd Segment Size',
       ' Avg Bwd Segment Size', ' Fwd Header Length.1', 'Fwd Avg Bytes/Bulk',
       ' Fwd Avg Packets/Bulk', ' Fwd Avg Bulk Rate', ' Bwd Avg Bytes/Bulk',
       ' Bwd Avg Packets/Bulk', 'Bwd Avg Bulk Rate', 'Subflow Fwd Packets',
       ' Subflow Fwd Bytes', ' Subflow Bwd Packets', ' Subflow Bwd Bytes',
       'Init_Win_bytes_forward', ' Init_Win_bytes_backward',
       ' act_data_pkt_fwd', ' min_seg_size_forward', 'Active Mean',
       ' Active Std', ' Active Max', ' Active Min', 'Idle Mean', ' Idle Std',
       ' Idle Max', ' Idle Min', ' Inbound'],
      dtype='object')
```

Figure 1: Features after preprocessing

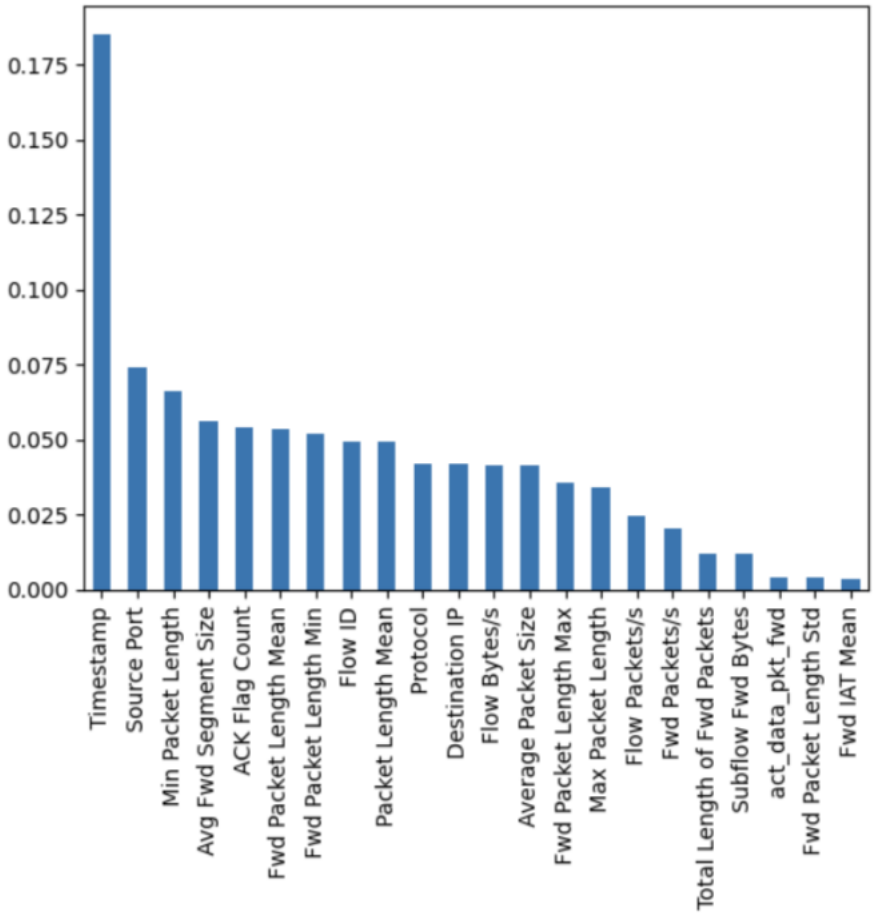

Figure 2: Feature Selection

After the data preprocessing step, we have chosen the model so in this paper we have selected Deep Neural Network model. Here a brief idea of Neural Network model is given:

	Timestamp	ACK Flag Count	Source Port	Avg Fwd Segment Size	Packet Length Mean	Fwd Packet Length Mean	Average Packet Size	Min Packet Length	Protocol	Fwd Packet Length Max	Fwd Packet Length Min	Max Packet Length	Flow Bytes/s	Flow ID	Flow Packets/s	Fwd Packets/s	Total Length of Fwd Packets	Subflow Fwd Bytes
0	3325329	0	870	211.0	211.0	211.0	316.5	211.0	17	211.0	211.0	211.0	4.220000e+08	15622001	2.000000e+06	2.000000e+06	422.0	422
1	3325330	0	871	265.0	265.0	265.0	397.5	265.0	17	265.0	265.0	265.0	1.104167e+07	15635734	4.166667e+04	4.166667e+04	530.0	530
2	3325331	0	648	229.0	229.0	229.0	343.5	229.0	17	229.0	229.0	229.0	4.580000e+08	13177109	2.000000e+06	2.000000e+06	458.0	458
3	3325332	0	872	229.0	229.0	229.0	343.5	229.0	17	229.0	229.0	229.0	4.580000e+08	15645914	2.000000e+06	2.000000e+06	458.0	458
4	3325333	0	873	229.0	229.0	229.0	343.5	229.0	17	229.0	229.0	229.0	4.580000e+08	15653823	2.000000e+06	2.000000e+06	458.0	458

Figure 3: Glimpse of the dataset after preprocessing and feature selection

Deep Neural Network Model: A neural network model with multiple fully connected layers, also known as a dense or feedforward neural network, comprises layers of interconnected neurons. Each neuron is connected to all neurons in the previous and subsequent layers. The

network begins with an input layer, followed by hidden layers, and ends with an output layer. Neurons within each layer apply an activation function to the weighted sum of their inputs, introducing non-linearity and allowing the network to capture complex relationships in the data. Weight parameters represent connection strengths and are adjusted during training to optimize performance. Each neuron also has a bias term to account for data offset. Multiple fully connected layers enable the network to learn increasingly abstract features from the input data. Deeper layers capture higher-level representations. Training involves iteratively updating weights and biases using optimization algorithms and backpropagation. Errors are propagated through the network to adjust parameters and improve performance. Neural networks with multiple fully connected layers excel at learning complex relationships, making them suitable for tasks like classification, regression, and pattern recognition. Their ability to extract meaningful features from input data makes them a powerful tool in data analysis.

Implementation: First, we have created a sequential model, which is a linear stack of layers then we added a fully connected layer (Dense) with 1024 units to the model by specifying the input dimensionality (22) for the first layer. The activation function used is ReLU (Rectified Linear Unit). Then we added a dropout layer with a dropout rate of 0.01. Dropout is a regularization technique that randomly sets a fraction of input units to 0 during training, helping to prevent overfitting. Further we added another fully connected layer with 2024 units and ReLU activation then another dropout layer like that we add 4 more layers of 2000, 1000, 500, and 200 along with ReLU activation function and dropout layer respectively. Eventually, after that we added a final fully connected layer with 10 units, which corresponds to the number of classes or output categories in the classification problem, a softmax activation layer to convert the outputs of the previous layer into probabilities, suitable for multiclass classification tasks. We defined an EarlyStopping callback to monitor the validation loss during training. It stops the training process if the validation loss does not improve for 5 consecutive epochs by a minimum change of 1e-3 (‘min_delta’). The ‘restore_best_weights’ parameter restores the weights of the model to the ones with the lowest validation loss. Overall, to tackle with attack we have used a neural network model with multiple fully connected layers, dropout layers for regularization, and an EarlyStopping callback for monitoring the training process. Then we have trained neural network model using the 'adadelta' optimizer. Then we have compiled the model that specifies the loss function as 'categorical_crossentropy', which is commonly used for multi-class classification problems. The optimizer is set to 'adadelta', which is an optimization algorithm that adapts the learning rate over time based on the past gradients. The metric chosen to evaluate the model is 'accuracy'. Then to train the model we have taken the training data and corresponding encoded labels as input. The validation data and corresponding encoded labels are provided for evaluating the model's performance during training. The ‘batch_size’ is set to 500, indicating that the model will update its weights after processing 500 samples. The ‘epochs’ parameter is set to 60, indicating the number of times the model will iterate over the entire training dataset. The ‘callbacks’ parameter is used to include the ‘monitor’ callback, which can be an instance of ‘EarlyStopping’ to monitor the validation loss and restore the best weights when training is stopped early. The object that contains the training history, including the loss and accuracy values at each epoch is used to display the available keys which typically include 'loss', 'accuracy', 'val_loss', and 'val_accuracy'. By observing the loss and accuracy (Figure 4) we can see the available metrics (Figure 5) that can be used to analyze the model's performance during training. This information is useful for further analysis and visualization of the training process.

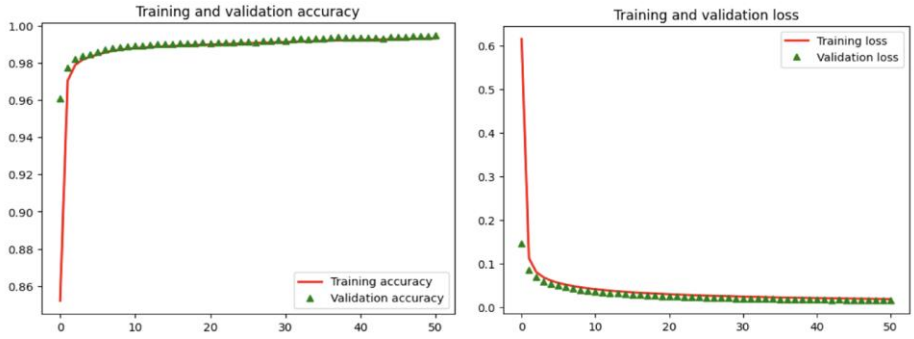

Figure 4: Training and validation plots (Accuracy/Loss)

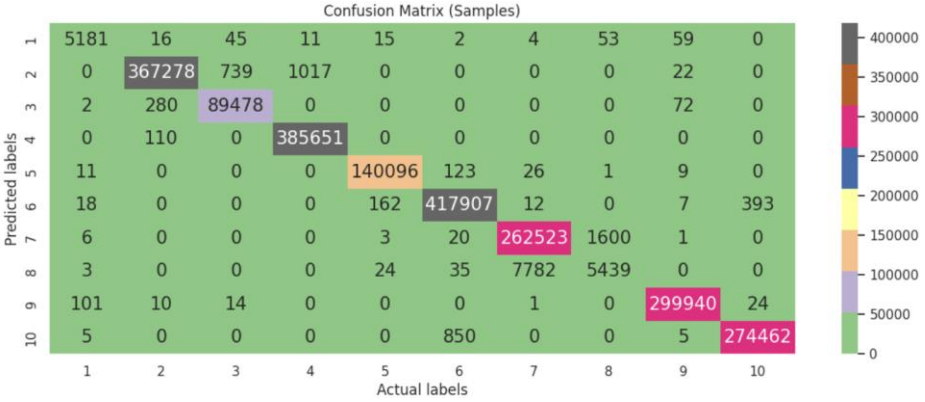

Figure 5: Confusion matrix of test data

4 Results Analysis

The Figure 4 and 5 displays the results of Deep Neural Network model used in the DDoS detection system. The model is associated with specific features that are utilized to identify and classify DDoS attacks accurately. The DNN model, being a powerful deep learning algorithm, is capable of capturing and learning patterns and attribute changes associated with DDoS attacks. By leveraging these features, the system achieves improved accuracy in detecting and classifying DDoS attacks. However, further details regarding the specific results and their corresponding features are given following:

- The developed DDoS detection system utilizes the DNN model, which has been trained and tested on various rates and input data. The best performance is achieved by this DNN model (99.39 %) on the test dataset.
- The system is capable of detecting attacks that occur within milliseconds, making it practically useful for real-time scenarios. Furthermore, the computation time is significantly lower compared to existing deep learning methods, indicating its efficiency in detecting and mitigating DDoS attacks.
- Our project has utilized 5277177 data for training and validation, with 2261648 data for testing. Increasing the size of the training dataset improves detection accuracy. A high-quality dataset

is essential for optimal performance. Our system successfully detects and classifies DDoS attacks, identifying the specific type of attack.
- Our project identified 22 key features that change during DDoS attacks and achieved comparable accuracy in detecting DDoS attacks on benchmark and real-time datasets, as shown in Table 1.
- The system analyzes network traffic features in real-time with minimal inference time. Additionally, it can send alert messages to inform users of DDoS attack occurrences.
- We utilized Wireshark and CICflowmeter software to capture and extract features from network traffic logs. Our system's enhanced accuracy contributes to the security of various networks by effectively detecting and classifying DDoS attacks.

5 Conclusion and future work

Our proposed approach utilizes a deep learning model such as a Deep Neural Network to train on a subset of the CICDDoS2019 dataset, resulting in an impressive 99.39% accuracy that surpasses other traditional methods. The analysis of results from these models suggests that the proposed DNN model is most suitable for detecting DDoS attacks, providing maximum accuracy and minimizing error rates. In future work, we plan to apply these techniques to multiple realistic datasets with varying topologies. Additionally, we aim to develop a real-time scenario for enhanced applicability of our approach.

References