The Indonesian Journal of Computer Science
www.ijcs.net
Volume 13, Issue 4, August 2024
https://doi.org/10.33022/ijcs.v13i4.4180

Detection of DDoS Attack Based on Deep Neural Network with various Number of Features

Suhad Shakir Jaber[1], Rasim Azeez Kadhim[2]
suhad.jaber@atu.edu.iq[1], rasimazeez@uobabylon.edu.iq[2]
[1]Technical College/ AL-Mussaib, Al-Furat Al-Awsat Technical University, Babylon, Iraq,
[2] College of Information Technology,University of Babylon, Babylon, Iraq

Article Information

Received : 23 Jun 2024
Revised : 27 Jul 2024
Accepted : 1 Aug 2024

Keywords

Distributed Denial of Service, Deep Neural Network, Deep learning.

Abstract

Distributed Denial of Service (DDoS) attacks have become an effective threat to the reliability and availability of the services of the internet in last decades. The effectiveness of utilizing Deep Neural Networks (DNNs) for DDoS attack detection is investigated in this paper. We implemented a reliable detection system that analyzes network traffic data to spot any DDoS activities. A multi-layer perceptron model trained on a dataset containing five different forms of DDoS attacks and normal traffic is used in this method. Also, three cases of varous number of features was investigated to extract the optimal number of features that can be used for detection of DDoS attacks. To improve accuracy, a great deal of testing was done on the model's architecture using various hyperparameters and training procedures. With a 96.5% detection rate, the DNN results showed a high degree of accuracy. This demonstration highlights the ability of deep neural networks to identify DDoS attacks in the midst of regular traffic. The six-category classification enhances detection granularity and facilitates the application of more specialized and successful mitigation techniques. Given the great precision attained, DNNs have the potential to be an essential part of real-time detection systems, providing a major advancement over conventional techniques.

A. Introduction

A distributed denial-of-service attack is characterized as an attack that takes advantage of the inherent vulnerabilities in the Internet's infrastructure to disrupt services that are delivered over it. In the modern, interconnected world, cyberattacks have grown to be a significant issue that worries a wide range of stakeholders, including people, businesses, and governments [1].

Therefore, the main goal of a DDoS assault is to overload and degrade online services in order to confuse and disrupt them, preventing regular users from reaching the host that provides the required services. Because the distributed denial-of-service attack primarily relies on taking advantage of a network of electronic devices, represented by cameras, smartphones, and desktop or labtop computers, the iregular user launches a coordinated attack that predominate the resources of the application layer or the processing power and the bandwidth. This allows the attacker to overwhelm the target servers or network with an enormous volume of data traffic. The target's resources are depleted by this massive influx of data, making it impossible for the systems to handle and react to requests that have been granted authorization [2].

DDoS attacks can be in different types, including volumetric attacks, which highly increase traffic of the network under attack, application layer attacks, which spot on a specific service or application, and TCP-based attacks, which use the statefulness of the TCP connection to exhaust server resources. One of the most prevalent kinds of DDoS assaults is TCP-based [3]. In order to mimic the first authentic connection phase of the procedure of three-ways handshake, the attacker sends a large number of TCP SYN packets to the victum network in these attacks, which result in an enormous number of TCP connections [4].

The attacker fails to deliver the decision packet, which is the final connection phase, so this process is not entirely finished. Half-open connections flood the target system as a result, using up system resources and preventing it from processing valid requests. Additionally, DDoS assaults can be categorized into groups based on their origin, such as application-layer attacks that exploit vulnerabilities in certain programs or network-layer attacks, which leverage traffic from several IP addresses. However, there are numerous approaches for identifying, stopping, or lessening these attacks. Network-based detection techniques are one method of doing this, which involves monitoring network traffic and identifying anomalous patterns in it. One more that is regarded as a network-based technique is traffic analysis. Additionally, modern distributed denial-of-service attack detection methods heavily rely on machine learning and artificial intelligence approaches. Diverse methods and strategies are used in various networks to identify and counteract DDoS attacks. These consist of various security technologies such as machine learning, and statistical techniques. The network architecture and possible attack types determine which approach is best to employ [5].

Many techniques have been proposed by the researchers for addressing the DDoS attacks on networks. In [6], the research focused on the DDoS attacks detection by employing sequential neural networks (NNs) to classify packets, separate TCP datagrams, determine the type of TCP packet and detect port scans.

This approach achieved high accuracy of 99% recognition rate of TCP. In [7], the authors integrated the Google's Word2vec with Global Vectors for Word Representation. This hybrid model was compared with some machine learning models. Based on real dataset, the approach showed an efficient detection of isider threats.

In [8], a detection method of DDoS attack have been adopted by monitoring the entropy of variation the IP address of the desination. This work invstment the flexibility of the OpenFlow controller (POX) and OpenFlow protocol's. The proposed method that computed the distributed features of DDoS attack based on the entropy showed the capability to detect the user datagram protocol flooding attack with 0.445 sec from the starting of attack [8].

In [9], the authors developed a model to prevent the DDoS attack at the application layer by decentralized platform and provide IoT system security by tackling the single point failure problem. Also, in order to prevent malicious users from connecting with the IoT networks, the authentication and verification of IoT devices are carried out.

B. Research Method

The proposed system utilizes the deep neural network for detecting and mitigating the effect of DDoS attacks on the victum. The framework is showing in Figure 1.

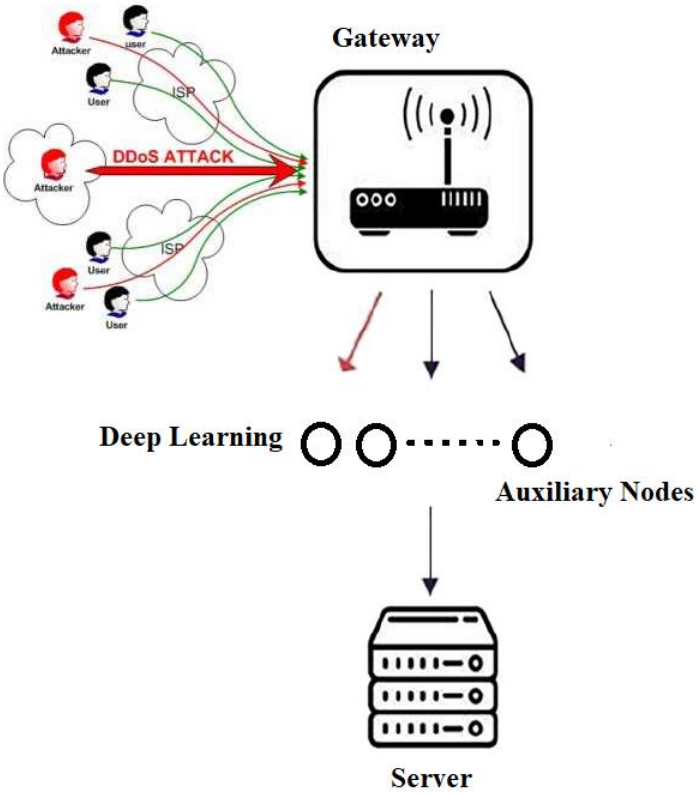


Figure 1. The Proposed System Framework.

The architecture of the system has two types of nodes:

1- The Server is the main node which receives and responds to the requests of the clients and monitor the traffic of the network espcially when the processing capacity is overloaded.
2- The second type of node is the auxiliary nodes represented by smart contracts that have direct connections to the server. These smart nodes contracts represents specialized nodes used to distinguish between potential attackers and normal clients by using the deep neural network. All requests must be pass through these nodes for checking and taking action agaist the sender either allowing the requests to the server or blocking the senser.

Figure 2, illustrates the flowchart of the proposed system. The system consists of multiple phases, including data collection, preprocessing, model training, and testing.

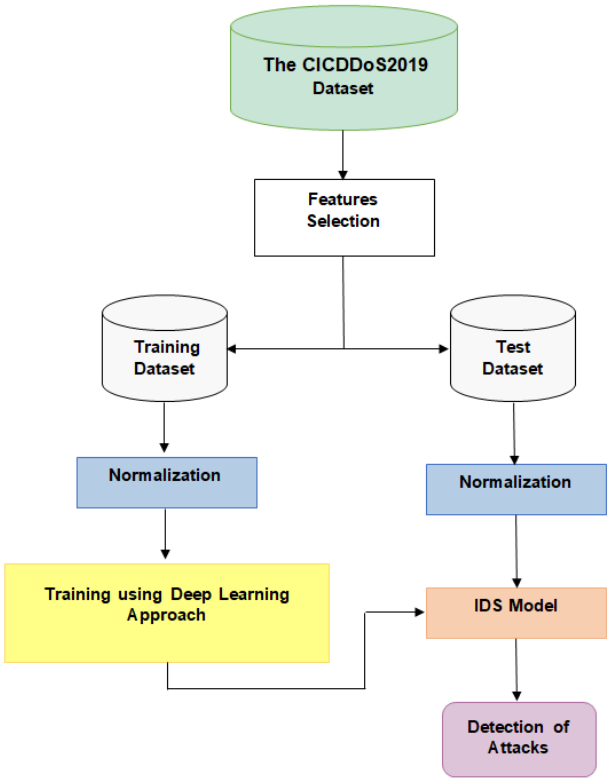


Figure 2. The Proposed DDoS Attack Detection Flowchart

The system Architecture is:

1) **Data Collection and Preprocessing**
   - **Data Sources:** The widely used dataset is CICDDoS2019 that contains network traffic data of DDoS attacks. It contains a set of 88 features and the corresponding labels, where the label information indicates either the type of DDoS attack or normal network traffic. This dataset equips comprehensive data about the various types of

attacks, such as UDP flood and SYN flood system that collects network traffic data from different sources, firewalls, network routers, and intrusion detection systems (IDS).

- **Feature Extraction:** the dataset has 88 features, where ten features are selected represented by: Fwd Pkt Len Std, Pkt Len Std , Pkt Len Max, Fwd Seg Size Min, Init Fwd Win Bytes, Fwd Pkt Len Max, Fwd Pkt Len Min, Pkt Size Avg, Pkt Len Min, and Fwd Header Len.

2) **Model Training**

- **Deep Neural Network Architecture:** The employment of the system is accomplished by using deep learning model, specifically a feature input layer and fully connected layer and classification layer. The structure of the DNN is shown in figure 3.

```
7×1 Layer array with layers:

   1   ''   Feature Input           3 features with 'zscore' normalization
   2   ''   Fully Connected         50 fully connected layer
   3   ''   Batch Normalization     Batch normalization
   4   ''   ReLU                    ReLU
   5   ''   Fully Connected         6 fully connected layer
   6   ''   Softmax                 softmax
   7   ''   Classification Output   crossentropyex
```

Figure 3. The structure of the deep neural network.

- **Training Process:** The labeled dataset is split into training and testing sets. The training set is used in training phase, where it learns to differentiate between malicious and benign traffic. Table 1, contains the hyperparameters such as learning rate, batch size, and the number of layers are optimized to improve model performance.
- **Testing Process:** The final system performance is evaluated using the test set, measuring metrics such as accuracy, precision, recall, and F1-score

Table 1. The Deep Neural Network Parameters

Parameter	Value
Input Layer size	3, 5, 10 inputs features
Number of layers	7
Hidden Layers Activation function	'relu'
Output layers activation function	'sigmoid'
Optimizer Type	'adam'
Epoch	2
batch_size	1000
Loss	'binary_crossentropy

C. The Evaluation

The confusion matrix is used to evaluate the proposed model as shown in Table 2. The confusion matrix's parameters of the two classes contains True Positive (TP), which represents normal traffic correctly recognized, and True Negative (TN), which performs benign traffic correctly identified. False Positive (FP) performs attack traffic recognized as normal traffic, and False Negative (FN) performs benign traffic mistakenly recognized as attack traffic. This evaluation methodology considers as an important aspect of the results and aims to measure the effectiveness of the model in distinguishing between attack and normal traffic.

Table 2. The Confusion Matrix

Predictive Class	True Class: Positive	True Class: Negative
Positive	TP	FP
Negative	FN	TN

The evaluation metrics that commonly used to evaluate the performance of the system are: accuracy, precision, recall, and F1 score, as illustrated in Table 3. The precision metric measures the proportion of correct positive recognitions out of all positive recognitions, while recall refers to the ratio of correctly recognized actual positives. The accuracy metric assesses the proportion of correctly classified cases, whereas the F1 score metric provides a balance between recall and precision.

Table 3. The Performance Evaluation Metrics

Metrics	Formula
Recall	$Recall = \frac{TP}{FN + TP}$
Precision	$\text{Precision} = \frac{TP}{TP + FP}$
Accuracy	$\text{Accuracy} = \frac{TN + TP}{TP + FP + TN + FN}$
F1-score	$F1 - \text{Score} = \left(\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}\right) \times 2$

D. Results and Discussion

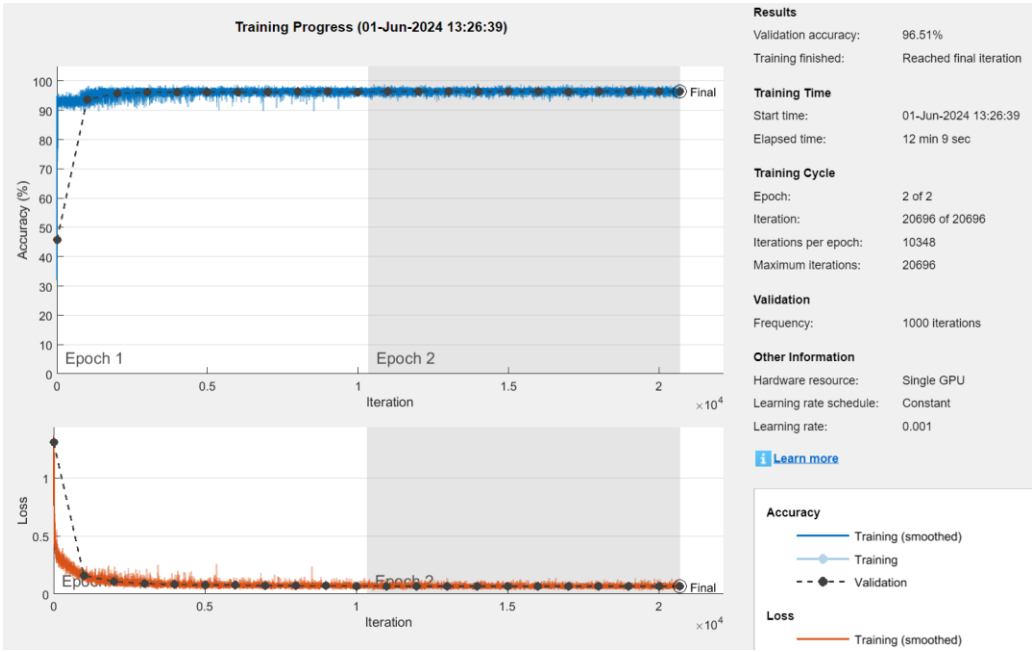
The MATLAB programming language v.2021a, and an operating system of (Microsoft 2010) with the processor (I7), and memory size of (16GB) are used to build the system model. The DNN model of classification is trained and tested based on the CICDDoS2019 dataset. The information of the selected features was normalized, reorganized, and any missing or infinite values were replaced with the mean values. To evaluate the effectiveness of the proposed deep learning-based DDoS attack detection system, we used the CICDDoS2019 dataset [10], which includes a comprehensive collection of network traffic data labeled for normal and various attack types, including DDoS attacks. The dataset was preprocessed to extract relevant features as mentioned above, forming the input for our deep learning model. The dataset was split into training (85%), and testing (15%) sets. The model was trained on the training set with hyperparameters tuned to optimize performance. After training, the model was tested on the testing set, and the results were evaluated using standard performance metrics, including accuracy, precision, recall, F1-score, and the confusion matrix. The DNN was trained on this dataset to classify six classes represented by the Benign, LDAP, NetBIOS, PortMAP, MSSQL, and UDP. The training progress of the DNN using three features is shown in figure 4.a, and the confusion matrix is shown in figure 4.b. Also, figure 5 illustrates the training progress and the confusion matrix using five features. Finally, the training progress and the confusion matrix of the DNN using ten features is shown in figure 6. Clearly, the accuracy of the system about 96.5% for all three cases.

Table 4, demonstrates the evaluation metrics for three cases of different number of input features. The stability in accuracy across different feature sets suggests that the model's capability to distinguish between normal and malicious traffic is strong, regardless of the number of features. However, accuracy alone does not fully capture the performance nuances, as evidenced by the variations in precision and recall. The Trade-offs in Precision and Recall:

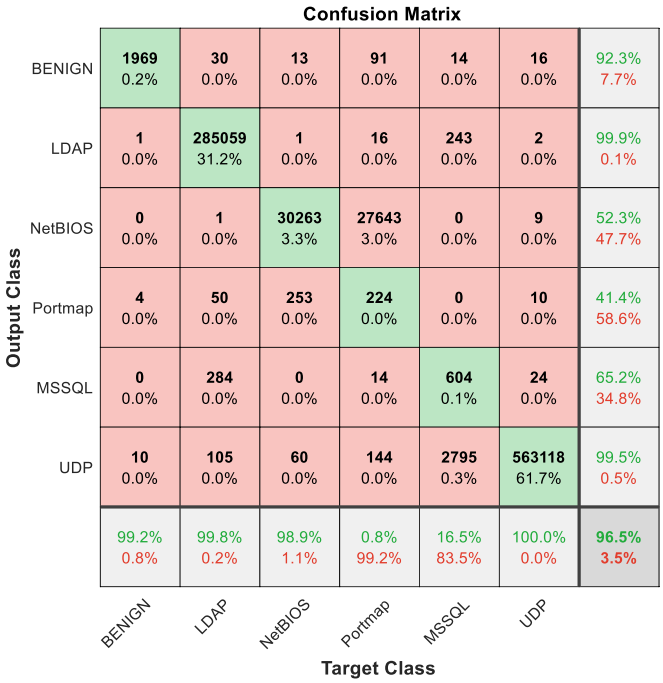
- The highest precision with 5 features indicates that this configuration minimizes false positives, which is crucial for reducing unnecessary alerts in a practical scenario.
- The highest recall with 10 features indicates that this configuration is better at detecting more true positives, which is critical for identifying as many attacks as possible.

Optimal Feature Set: in comparison among these three cases, the case of 10 features have the maximum recall, but there may be an overfitting problem as indicated by the sharp decline in precision and F1-score. On the other hand, the case of three features, that achieved the best F1-score, is a more dependable option in real-world applications where it is necessary to reduce both false positives and false negatives since it better balances precision and recall. The significance of feature engineering in machine learning models is emphasized by these findings. Adding more features does not always result in improved performance; in fact,

overfitting or the inclusion of unnecessary data may worsen performance. Achieving the optimum performance requires choosing an ideal feature set

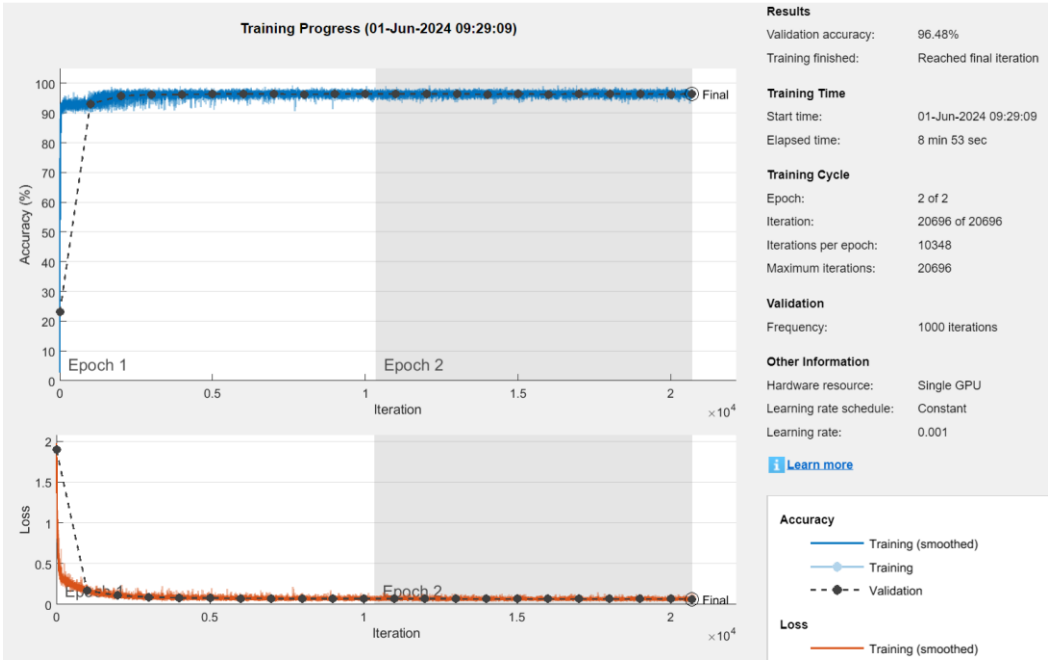


(a)

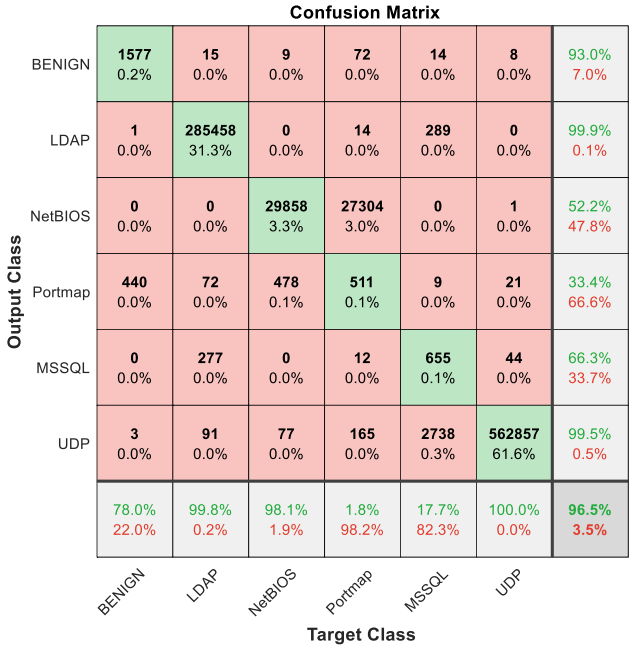


(b)

Figure 4. The DNN training of three features a) the accuracy (b) the confusion matrix

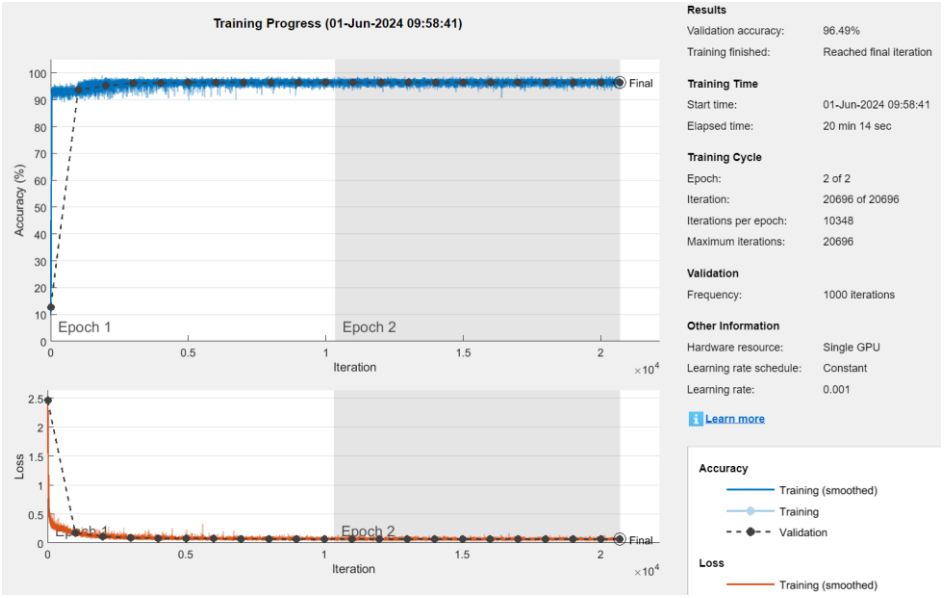


(a)

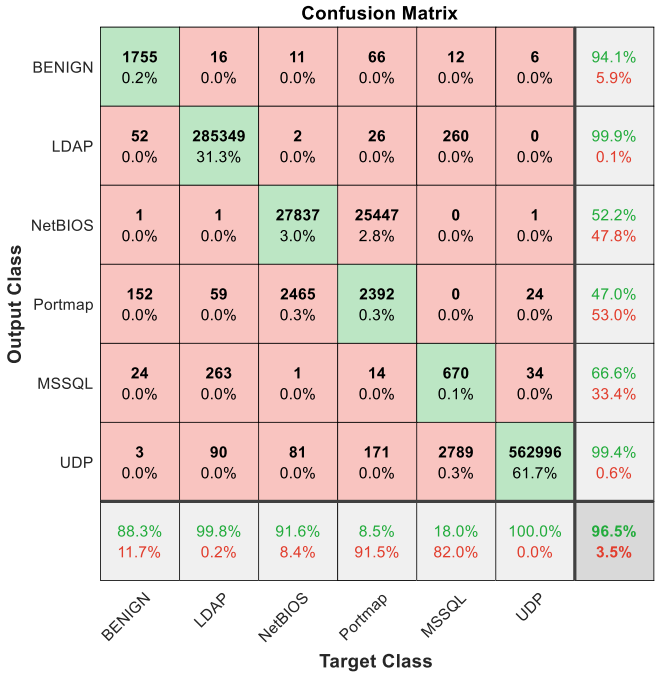


(b)

Figure 5. The DNN training of three features a) the accuracy (b) the confusion matrix



(a)



(b)

Figure 6. The DNN training of three features a) the accuracy (b) the confusion matrix

Table 4. The Evaluation metrics for three cases.

No. of features	Accuracy (%)	Precision (%)	Recall (%)	F1-score(%)
3	96.48	70.85	69.25	70.04
5	96.48	74.05	65.9	69.73
10	96.49	61.83	67.7	64.63

E. Conclusion

By adding to the expanding corpus of work on the use of machine learning techniques to cybersecurity, this study lays the groundwork for future developments in the detection and mitigation of DDoS attacks. This study shows how deep neural networks may be used to detect DDoS attacks with six different classes, with considerable potential. With an astounding accuracy of 96.5%, our detection system successfully discerns between different kinds of DDoS attacks and regular traffic by utilizing the sophisticated pattern recognition powers of DNNs. This high detection rate demonstrates how well neural networks perform to recognize subtle and complicated patterns linked to criminal activity, offering a strong answer to one of the most important cybersecurity concerns. Future research will concentrate on expanding this methodology to encompass an even wider spectrum.

F. References

[1] B. A. Khalaf et al.(2021) "An adaptive protection of flooding attacks model for complex network environments," Security and Communication Networks, vol. 2021, pp. 1–17, http://dx.doi.org/10.1155/2021/5542919.

[2] B. J. S. Kumar and K. R. K. Gowda (2022) "Detection and Prevention of TCP SYN flooding attack in WSN using Protocol dependent Detection and Classification System," 2022 IEEE International Conference on Data Science and Information System (ICDSIS), http://dx.doi.org/10.1109/ICDSIS55133.2022.9915949

[3] B. Ramkumar and T. Subbulakshmi (2021). "Tcp Syn Flood Attack Detection and Prevention System using Adaptive Thresholding Method," International Conference on Innovative Technology for Sustainable Development (ICITSD-2021), vol. 37, p. 01016, http://dx.doi.org/10.1051/itmconf/20213701016

[4] R. S. D. W. Banu, T. N. Jyothi, M. Amulya, K. N. Anju, A. Raju, and S. N. Kashyap (2019). "MONOSEK – A Network Packet Processing System for Analysis & Detection of TCP Xmas attack using Pattern Analysis," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), http://dx.doi.org/10.1109/ICCS45141.2019.9065325

[5] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang (2017). "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," International Journal of Distributed Sensor Networks, vol. 13, no. 12, p. 155014771774146 , http://dx.doi.org/10.1177/1550147717741463

[6] B. Hartpence and A. Kwasinski,(2020). "Combating TCP port scan attacks using sequential neural networks," 2020 International Conference on Computing, Networking and Communications (ICNC), http://dx.doi.org/10.1109/ICNC47757.2020.9049730

[7] Haq, M.A.; Khan, M.A.R.; Alshehri, M. Insider Threat Detection Based on NLPWord Embedding and Machine Learning. Intell. Autom. Soft Comput. **2022**, 33, 619–635

[8] M. I. Kareem and M. N. Jasim (2022). "Entropy-based distributed denial of service attack detection in software-defined networking," Indonesian Journal of Electrical Engineering and Computer Science, vol. 27, no. 3, p. 1542, http://dx.doi.org/10.11591/ijeecs.v27.i3.pp1542-1549

[9] H. M. Mohammad and A. A. Abdullah (2023) "DDoS attack mitigation using entropy in SDN-IoT environment," Nucleation and Atmospheric Aerosols, http://dx.doi.org/10.1063/5.0123465.

[10] DDoS Evaluation Dataset (CICDDoS2019). https://www.kaggle.com/datasets/krisshibu/cicddos-2019full-dataset