The First International Conference on
Machine Learning and Knowledge Discovery (MLKD 2024)
Amirkabir University of Technology, December 18-19, 2024

A One-Dimensional Convolutional Neural Network Intrusion Detection System on the CICDDoS2019 Dataset in Cloud Environments

Elmira Majidi Hatkehloui* Payam Mahmoudi Nasr †

Abstract

In the modern era, cloud computing has become a cornerstone of computer science and networking. As a result, security concerns and the risk of intrusions have emerged as critical challenges for individuals and organizations managing cloud networks. To address these issues, developing an effective intrusion detection system is crucial. Attackers continuously attempt to breach cloud networks through various forms of attacks, which can lead to significant data breaches and potentially devastating consequences. Hence, identifying vulnerabilities and detecting attacks within cloud environments is of paramount importance. This paper presents a deep learning architecture utilizing a one-dimensional Convolutional Neural Network (1D-CNN) for detecting network attacks in cloud systems. This research identifies nine different types of attacks on cloud networks. The evaluation was conducted using the CICDDOS2019 dataset, yielding an accuracy of 99.92%. These outcomes, as confirmed through practical experiments, highlight the model's high effectiveness.

Keywords: one-dimensional Convolutional Neural Network(1DCNN), cloud computing (CC), CICDDOS2019, Deep Learning (DL), Intrusion Detection System (IDS)

1 Introduction

Over the past few decades, the rise of sophisticated cyber threats has significantly escalated, posing a range of challenges for organizations [1]. Figure 1 depicts the growing trend of attacks on cloud networks. This scenario has driven the advancement of Intrusion Detection Systems (IDS). Implementing IDS is essential both in academic settings and in practical network environments, as every cyberattack can result in financial losses, harm to reputation, and potential legal repercussions [2]. Protecting cloud networks from unauthorized access, ensuring secure user interactions, and safeguarding user data are critically important [3]. It's essential to continuously address security vulnerabilities and employ the latest technologies to counteract emerging threats [4]. Distributed Denial of Service (DDoS) attacks represent a major threat by disrupting access to cloud services and causing significant issues. The vast amount of malicious data that infiltrates cloud environments is often varied and inconsistent, making it a major cybersecurity challenge. In recent years, the frequency and severity of DDoS attacks have increased substantially, leading to serious repercussions for internet and cloud service providers. Figure 2 . illustrates

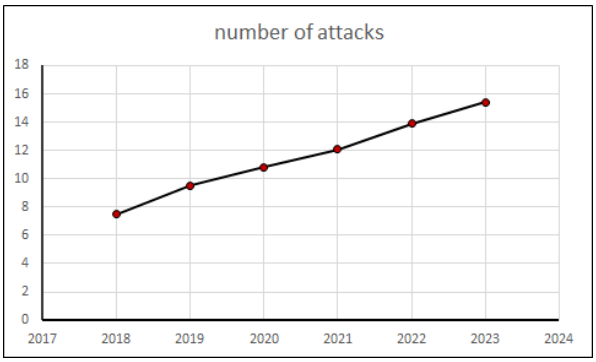

Figure 1: The number of DDOS attacks in cloud environments`https://www.stationx.net/cloud-security-statistics/`

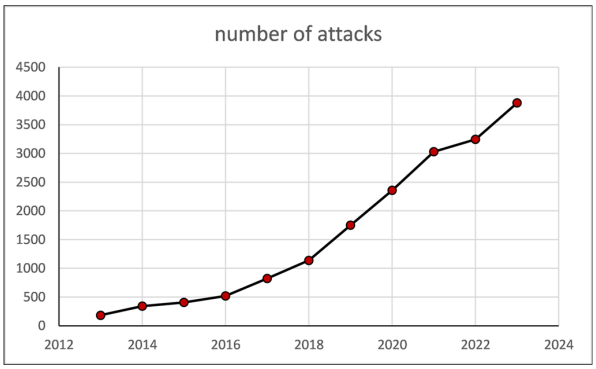

Figure 2: The number of attacks in cloud environments has significantly increased over the past 10 years`https://www.stationx.net/cloud-security-statistics/`

*Department of Computer Engineering, Faculty of Engineering and Technology, University of Mazandaran, Babolsar, Iran, `e.majidi22@umail.umz.ac.ir`

†Department of Computer Engineering, Faculty of Engineering and Technology, University of Mazandaran, Babolsar, Iran, `p.mahmoudi@umz.ac.ir`

the rising trend of DDoS attacks in cloud environments. DDoS attacks have the potential to inflict serious dam-

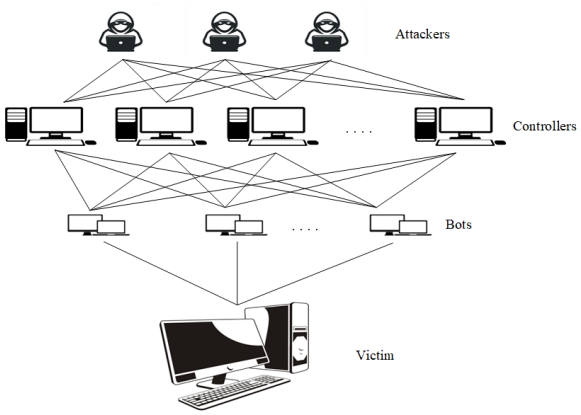

Figure 3: the architecture of a DDoS attack

ageon cloud networks, as they can illegitimately access different parts of the system [5]. Figure 3 shows the architecture of a DDoS attack, which is comprised of four elements: the attackers, controllers, bots, and the victim. These attacks overwhelm the server by infecting devices and creating a botnet, generating high levels of traffic that saturate the system and strain its capacity [6]. Given the significant risk that DDoS attacks present to service providers, tackling this issue has become a necessity. To counter these threats, researchers have designed a range of strategies to help reduce, detect, and even prevent these attacks effectively. One effective approach to prevent these attacks is to create an intrusion detection system [7]. Intrusion detection systems are valuable for boosting security against attacks in cloud environments [8]. These systems monitor cloud network traffic to identify unusual behaviors and protect the network, helping to minimize financial and operational impacts. By analyzing network patterns, they can detect anomalies and alert administrators to potential threats, enabling timely intervention to prevent attacks. Essentially, IDS serve as virtual watchdogs, spotting irregular intrusions and enhancing overall network protection [6]. Intrusion detection systems are typically categorized into three types: anomaly-based, signature-based, and hybrid systems [9]. Anomaly-based systems focus on identifying unusual activities within a network. They use statistical models, machine learning, deep learning, and other methods to analyze and record normal patterns, which allows them to detect deviations from these patterns [7]. A notable benefit of these systems is their capability to identify zero-day and previously unknown threats in cloud environments. Signature-based intrusion detection systems, often referred to as rule-based systems, detect attacks by recognizing established traffic patterns. These systems excel at identifying threats for which they have specific information. However, they have limitations: they need frequent updates to their databases and are unable to detect new or zero-day attacks [6]. The optimal solution would be to develop a system that can process data in real-time while consuming fewer resources. This approach would not only cut down on costs but also enhance performance and responsiveness in rapidly changing cloud environments. This study aims to create an anomaly-based intrusion detection system capable of precisely identifying and categorizing different types of attacks using the CICDDOS2019 dataset. By leveraging deep learning techniques, the system has achieved notably successful outcomes. This study employs a one-dimensional convolutional neural network (1D-CNN) to identify and classify various types of attacks. An advanced deep learning model built on 1D-CNN has been crafted for analyzing network traffic. The key contributions of this research include:

- Development of a highly effective and efficient intrusion detection system for cloud environments using deep learning techniques
- Thorough review of current deep learning methods
- Detailed analysis of the CICDDOS2019 dataset for improved detection and updating
- Comprehensive comparison with existing network intrusion detection systems
- Presentation of a novel classification approach based on a one-dimensional convolutional neural network

This thesis is organized into several key sections. Section 2 offers an in-depth analysis of significant research related to network intrusion detection systems that use deep learning models. Section 3 focuses on the data and preprocessing methods employed in the study. In Section 4, the proposed approach for designing an intrusion detection system (IDS) using deep learning techniques is discussed. Section 5 presents and analyzes the results of the model. Finally, Section 6 summarizes the findings and provides recommendations for future research.

2 Related Work

In 2023, Ramzan and his team explored various deep learning models, including Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU), for detecting DDoS attacks. They evaluated these models using the latest dataset, CICDDoS2019, and compared their performance with the earlier CICIDS2017 dataset. This

comparison aimed to advance the development of efficient and accurate DDoS detection methods while reducing execution time and complexity. Their findings revealed that while the models achieved similar accuracy rates of 0.99 on the CICDDoS2019 dataset, GRU exhibited faster execution times compared to RNN and LSTM [10]. In 2024, AlSaleh and his team developed a Bayesian Convolutional Neural Network (BaysCNN) for detecting DDoS attacks with the CICDDoS2019 dataset. Their model, consisting of 19 layers, achieved an accuracy of 99.66%. By enhancing this model to BaysFusCNN, which features 27 layers, they improved the accuracy further, reaching an average of 99.79% [11]. In 2023, Elubeyd and his team utilized SDN to implement and evaluate multiple algorithms on the CICDDoS2019 dataset. They experimented with three different algorithms and found that the CNN approach achieved an accuracy of 95%. By integrating these three algorithms, they managed to significantly improve their results, reaching an accuracy of 99.88% [12]. In 2024, Shaikh and his team implemented a deep learning-based system for detecting DDoS attacks using a CNN-LSTM model with the CICDDoS2019 dataset. Their approach achieved a high accuracy rate of 99.89%. They also applied autoencoders for dimensionality reduction and utilized SMOTE to tackle class imbalance issues [13]. In 2023, Shieh and team presented an intrusion detection system utilizing a CNN algorithm on the CICDDoS2019 dataset. By integrating geometric metric features, they were able to enhance the algorithm's effectiveness. Their system achieved a notable accuracy of 99.8% and was capable of identifying zero-day attacks [14].

3 Data preprocessing and analysis

3.1 Dataset

Researchers required a robust and up-to-date dataset with a range of features and criteria, essential for thoroughly testing, evaluating, and validating the model [15]. In response, numerous datasets have been created and released in recent years, providing researchers with the resources they need to select and apply the most data for their specific research objectives [16]. In 2019, Ali Gasm and his colleagues created this dataset to advance research in the development of intrusion detection systems, and it was published by the Canadian Institute for Cybersecurity (CIC) [14]. The dataset includes common and up-to-date attacks that closely resemble real-world data (PCAP). It also features network traffic analysis results using CICFlowMeter, which includes labeled flows based on time, source and destination IPs, source and destination ports, protocols, and attacks [17]. This dataset is, in fact, an updated version of the CICIDS2017 dataset, specifically focused on DDoS attacks.The features of this dataset are outlined in Table 1. The CICDDoS2019 dataset is extensively utilized in research, especially for studying cloud network attacks. Despite its comprehensive coverage, the dataset has a critical issue that must be addressed: class imbalance. This imbalance can lead to inaccuracies in the classification process. In the research, a selection of the primary features (totaling 88) was utilized, while certain features were omitted. These omitted features included data like source and destination IP addresses, timestamps, flow IDs, and similar attributes. This left around 78 features in the dataset. Of the remaining features, the final one (feature 78) indicates the traffic classification within the bidirectional flow and is used as the label. This dataset includes two types of consumption profiles as well as multi-stage attacks such as Heartbleed and various DoS and DDoS attacks. The generated data is in CSV format and contains records of traffic features. The CICDDoS2019 dataset comprises nearly fifty million sixty-three thousand one hundred and twelve records, with fifty million six thousand two hundred and forty-nine rows related to DDoS attacks and approximately fifty-six thousand eight hundred and sixty-three rows corresponding to normal traffic. Each row contains 88 features. The attacks in this dataset involve various protocols, including Network Time Protocol (NTP), Microsoft SQL Server (MSSQL), Domain Name System (DNS), and 12 types of DDoS attacks such as UDP-Lag, LDAP, NetBIOS, SSDP, SNMP, SYN, UDP, WebDDoS, and TFTP. In the test dataset, there are seven different types of attacks, including MSSQL, SYN, PortScan, LDAP, NetBIOS, UDP-Lag, and UDP. Table 2 presents the names of the attacks and the count of each attack recorded in the CICDDoS2019 dataset [18]

Table 1: Details of the CICDDOS2019 dataset

Dataset	CICDDOS2019
Type	Multi-class
Year	2019
Class	8
Number of Data	50,603,112
Number of Features	88

3.2 preprocess

During the data preprocessing stage, the first step involved removing 10 columns where all the data values were zero. Following this, any 'NaN' (Not a Number) and 'infinity' values were removed from the entire dataset. After cleaning the data, logarithms with a base of 10 were applied to large numerical values to prevent potential issues during the training phase. After the log-

Table 2: The names and counts of attacks in the CICD-DoS2019 dataset

Attacks	Number
UDP-Lag	3,415,526
LDAP	3,064,952
MSSQL	2,047,784
NetBIOS	2,037,197
SSDP	5,259,204
SNMP	3,307,012
SYN	10,059,013
UDP	5,897,745
WebDDoS	3,940,867
TFTP	2,976,949
NTP	3,229,524
DNS	5,770,476

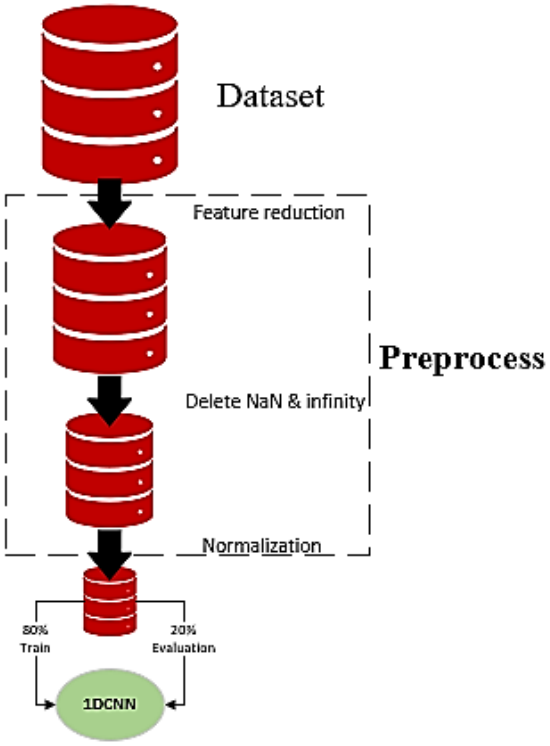

Figure 4: Data Preprocessing Steps

arithm transformation, the numbers were normalized to a range between -1 and 1. With these steps completed, the data is now ready for training. The preprocessing steps are illustrated in Figure 4.

4 Proposed Method

In this research, a one-dimensional convolutional neural network (1D-CNN) has been utilized for detecting

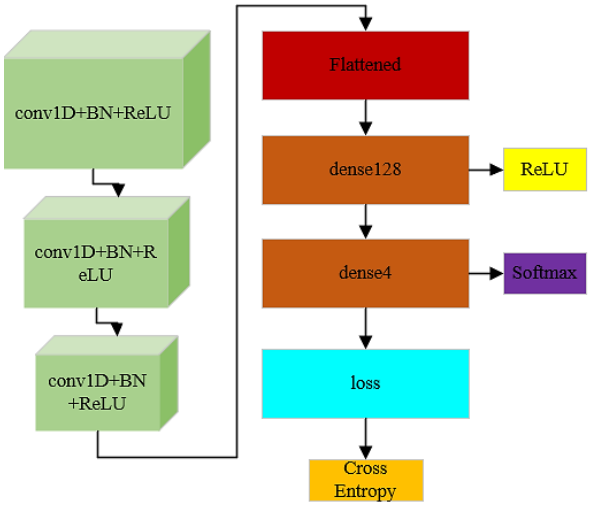

Figure 5: Data Preprocessing Steps

network attacks. The 1D-CNN is specifically designed to operate on one-dimensional data. A 5-layer intrusion detection system has been developed using a CNN model. This system consists of three one-dimensional convolutional layers and two dense layers with sizes 128 and 4. To address data imbalance, a weighting method has been applied. For the CICDDoS2019 dataset, due to the large data volume and system memory constraints, two separate intrusion detection systems were used: one with four classes and the other with three classes. Finally, for the CICDDoS2019 dataset, the average of the evaluation metrics was considered. Figure 5 illustrates the architecture implemented in the CNN algorithm. In convolutional neural networks, the convolutional layer is responsible for filtering inputs and processing data. This layer utilizes repeated filtering operations to extract important features. During this process, the inputs are multiplied by kernel weights to produce new values known as feature maps. After this stage, the feature map is passed through the ReLU activation function. ReLU transforms negative inputs to zero and passes positive inputs unchanged as output. This function helps the model learn more quickly and reduces issues related to vanishing gradients. The formula 1 defines this activation function.

Formole 1:

$$\mathrm{ReLU}(x) = \max(0, x)$$

Here, x represents the input to the activation function, and Relu(x) is its positive output. Convolutional layers are typically followed by additional blocks of convolutional layers. Pooling layers are used to reduce dependency on specific features and to prevent overfitting in the model. These layers transform prominent features into aggregated features. The choice between maximum

pooling and average pooling depends on the filter size and pooling type; in maximum pooling, the maximum value of the feature is taken, while in average pooling, the average value is computed. To mitigate overfitting in deep networks, dropout layers are employed, which prevent overfitting by randomly ignoring some neurons during training. This process helps the model perform better with test data (unknown data). After applying dropout, the active inputs are scaled by a specific factor so that the sum of all inputs remains constant using the following formula 2:

Formole 2:

$$z = \frac{1}{1 - \text{rate}}$$

Therefore, assigning responsibility to some of the nodes can introduce noise into the training process, and this technique is applicable only during training. The use of dropout increases the weights of the network. Next, fully connected layers are connected to the previous layers, and an activation function is applied to transform the results into the final output. In this dataset, the distribution of attacks and non-attacks is not balanced, with attacks constituting only about 2% of the total data [19]. To address the issue of data imbalance, various algorithms are available, which can be categorized into three main groups: preprocessing methods, in-model methods, and post-processing methods [18]. One well-known algorithm in the preprocessing category is SMOTE (Synthetic Minority Over-sampling Technique). This algorithm helps balance the dataset by generating synthetic samples for minority classes. In addition to SMOTE [13], other preprocessing algorithms include ADASYN, which focuses on generating synthetic samples near the decision boundary to enhance the quality of minority class samples. Methods like Tomek Links and Edited Nearest Neighbors are also employed to improve data balance and remove problematic samples. In this study, instead of using preprocessing methods, an in-model algorithm has been utilized. This algorithm includes weighting and normalization techniques to assist in balancing the classes. By applying a weighting function, higher weights are assigned to minority classes and lower weights to majority classes. This approach helps maintain data balance and improve model performance, as indicated by the following formula 3:

Formole 3:

$$w_i = \frac{N}{K \times n_i}$$

In this formula, wi denotes the weight for class i. The total number of samples in the dataset is represented by N , while K refers to the total number of classes. The variable ni stands for the number of samples in class i. The weight wi is calculated by dividing N by the product of K and ni. This method helps to balance the influence of each class on the model by adjusting for the class distribution in the dataset.

4.1 Proposed Method Architecture

The proposed architecture features three convolutional layers, a dropout layer, pooling layers, and two dense layers.

Convolutional Layers: The first convolutional layer uses 32 filters with a kernel size of 5 and applies the ReLU activation function. This layer processes the input, transforming it into a vector with dimensions of (32, 64). Following this, another convolutional layer is added, also with 32 filters and a kernel size of 5.

Dropout Layer: After these two convolutional layers, a dropout layer with a rate of 0.2 is introduced. This means that during each training step, about 20% of the neurons are randomly turned off. The dropout rate, which is set at 0.5 or 50% in this case, determines the proportion of neurons that are deactivated. This rate usually falls between 0 and 1.

Max-Pooling Layer: The dropout layer is followed by a max-pooling layer with a pooling size of 2. The purpose of the max-pooling layer is to reduce the number of parameters that the model needs to learn, thereby lowering its computational cost. The output is then flattened into a one-dimensional vector by a flattening layer.

Dense Layers: The final part of the architecture includes two dense layers that utilize the ReLU and SoftMax activation functions to make the final predictions.

The dataset was divided into two separate parts due to memory constraints. Each part was further split into two sections, with 80% used for training and 20% for testing. Finally, the results obtained are averaged. The classification performance of the 1DCNN was evaluated using various metrics, including accuracy, precision, recall, and F1 score, as defined by the following formulas from 4 to 7:

Formole 4:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Samples}}$$

Formole 5:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Formole 6:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Formole 7:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table 3: Parameters of the Proposed Model

Parameter	Value(%)
Epoches	50
Optimizer	Adam
Dropout Rate	0.2
Activation Function	ReLU

Table 4: Results Obtained

Parameter	Value(%)
Accuracy	99.92
Precision	99.94
Recall	99.93
F1 Score	99.93

Articles	Model	Dataset	Accuracy(%)
[11]	LSTM	CICDDOS2019	99
[12]	BaysFusCNN	CICDDOS2019	99.79
[13]	CNN	CICDDOS2019	99.88
[14]	CNN-LSTM	CICDDOS2019	99.89
[15]	CNN	CICDDOS2019	99.8
Proposed Method	1D-CNN	CICDDOS2019	99.92

Table 5: Comparison of the Proposed Method with Other Studies

5 Experimental Results and Discussion

The classification results from the CNN are detailed in Table 3. The results, obtained using the specified parameters, include an accuracy of 99.92%, precision of 99.94%, recall of 99.93%, and an F1 score of 99.93% across 50 epochs, as shown in Table 4. Additionally, there is a thorough comparison of the proposed method with other studies on the dataset. Figures 6 and 7 illustrate the loss and accuracy graphs for this dataset, divided into two sections. Figure 8 presents the confusion matrix for the model. Additionally, Table 5 offers a comparative analysis of the proposed method against other existing approaches, enhancing the presentation and conclusions.

6 Conclusion and Future Work

In this research, a one-dimensional convolutional neural network (1D-CNN) was utilized to identify DDoS attacks. The preprocessing stage involved data cleaning, applying logarithmic transformation, and removing irrelevant features. To handle class imbalance, weighting methods were implemented. Future research could consider alternative deep learning approaches and develop novel strategies to address class imbalance and memory limitations.

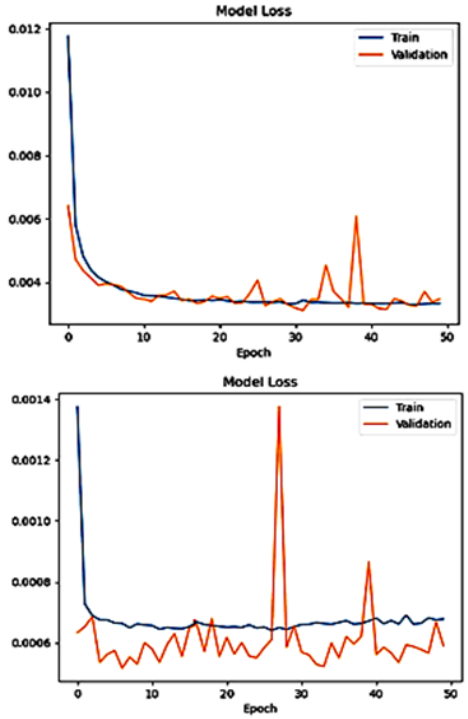

Figure 6: Plot of Loss Over 50 Epochs

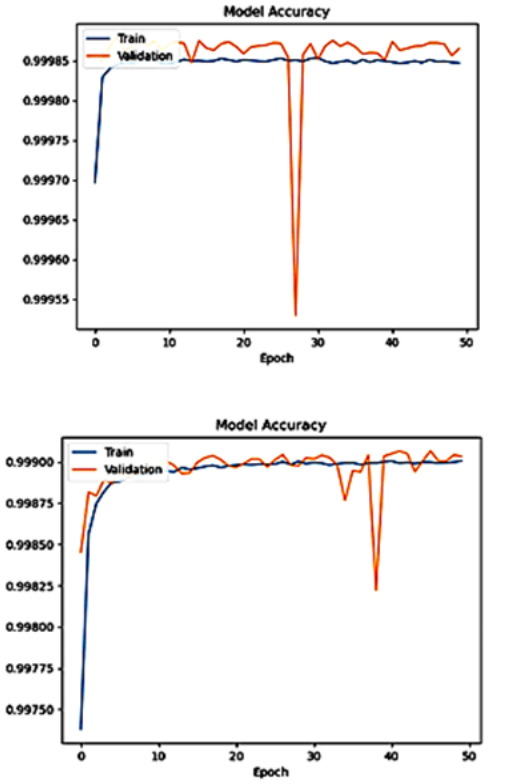

Figure 7: Plot of Accuracy Over 50 Epochs

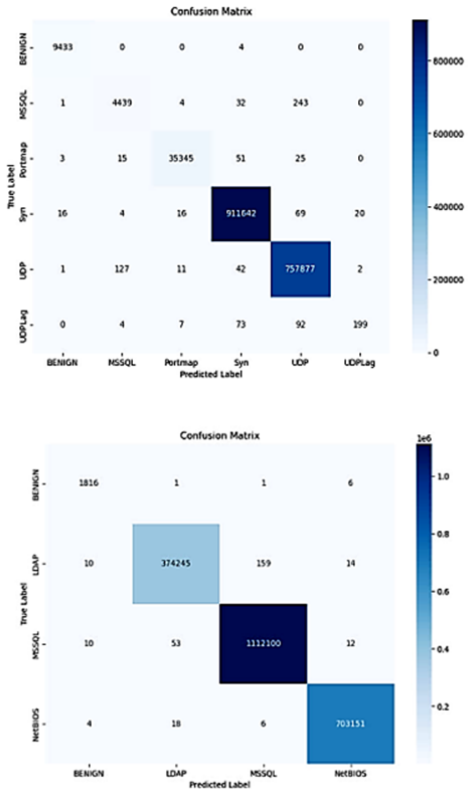

Figure 8: Confusion Matrix

References

[1] M. Premkumar and T. Sundararajan DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks. *Microprocessors and Microsystems*, vol. 79, p. 103278, 2020.

[2] E. U. H. Qazi, A. Almorjan, and T. Zia A one-dimensional convolutional neural network (1d-cnn) based deep learning system for network intrusion detection. *Applied Sciences*, vol. 12, no. 16, p. 7986, 2022.

[3] Y. Liao and V. R. Vemuri Use of k-nearest neighbor classifier for intrusion detection. *Computers & security*, vol. 21, no. 5, pp. 439-448, 2002.

[4] A. Mohiyuddin, A. R. Javed, C. Chakraborty, M. Rizwan, M. Shabbir, and J. Nebhen Secure cloud storage for medical IoT data using adaptive neuro-fuzzy inference system. *International Journal of Fuzzy Systems*, vol. 24, no. 2, pp. 1203-1215, 2022.

[5] Q. Yan, F. R. Yu, Q. Gong, and J. Li Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE communications surveys & tutorials*, vol. 18, no. 1, pp. 602-622, 2015.

[6] B. B. Gupta and O. P. Badve STaxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. *Neural Computing and Applications*, vol. 28, pp. 3655-3682, 2017.

[7] S. Mahdavi Hezavehi and R. Rahmani An anomaly-based framework for mitigating effects of DDoS attacks using a third party auditor in cloud computing environments. *Cluster Computing*, vol. 23, no. 4, pp. 2609-2627, 2020.

[8] M. Ali, S. U. Khan, and A. V. Vasilakos Security in cloud computing: Opportunities and challenges. *Information sciences*,vol. 305, pp. 357-383, 2015.

[9] M. Ouhssini, K. Afdel, E. Agherrabi, M. Akouhar, and A. Abarda DeepDefend: A comprehensive framework for DDoS attack detection and prevention in cloud computing. *Journal of King Saud University-Computer and Information Sciences*, p. 101938, 2024.

[10] M. Ramzan et al Distributed denial of service attack detection in network traffic using deep learning algorithm. *Sensors*, vol. 23, no. 20, p. 8642, 2023.

[11] I. AlSaleh, A. Al-Samawi, and L. Nissirat Novel Machine Learning Approach for DDoS Cloud Detection: Bayesian-Based CNN and Data Fusion Enhancements. *Sensors*, vol. 24, no. 5, p. 1418, 2024.

[12] H. Elubeyd and D. Yiltas-Kaplan Hybrid deep learning approach for automatic DoS/DDoS attacks detection in software-defined networks. *Applied Sciences*, vol. 13, no. 6, p. 3828, 2023.

[13] J. Shaikh, Y. A. Butt, and H. F. Naqvi Effective Intrusion Detection System Using Deep Learning for DDoS Attacks. *The Asian Bulletin of Big Data Management*, vol. 4, no. 1, pp. Science 4 (1)-183, 2024.

[14] C.-S. Shieh, T.-T. Nguyen, and M.-F. Horng Detection of unknown ddos attack using convolutional neural networks featuring geometrical metric. *Mathematics*, vol. 11, no. 9, p. 2145, 2023.

[15] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE access*, vol. 9, pp. 22351-22370, 2021.

[16] M. Al-Sharif and A. Bushnag Enhancing cloud security: A study on ensemble learning-based intrusion detection systems. *IET Communications*, 2024.

[17] T. G. Gebremeskel, K. A. Gemeda, T. G. Krishna, and P. J. Ramulu DDoS attack detection and classification using hybrid model for multicontroller SDN. *Wireless Communications and Mobile Computing*, vol. 2023, no. 1, p. 9965945, 2023.

[18] V. Jyothsna, C. Manisha, and B. NanduSri ntrusion Detection System for Detection of DDoS Attacks in Cloud Environment. , 2023.

[19] T. Althobaiti, Y. Sanjalawe, and N. Ramzan Securing Cloud Computing from Flash Crowd Attack Using Ensemble Intrusion Detection System. *Computer Systems Science & Engineering*, vol. 47, no. 1, 2023.