DDOS ATTACK DETECTION USING AI BASED TECHNIQUES

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE

OF

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted by:

SUGANDH M

2K21/ISY/24

Under the supervision of

Dr. Jasraj Meena

(Assistant Professor)

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2023

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi

CANDIDATE'S DECLARATION

I, Sugandh M, Roll No. 2K21/ISY/24 of M.Tech. (Information Systems), hereby declare that the dissertation report titled "DDOS ATTACK DETECTION USING AI BASED TECHNIQUES" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship, or other similar title or recognition.

Place : Delhi   Sugandh M

Date:

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi

CERTIFICATE

I, hereby certify that the dissertation which is submitted by Sugandh Roll No. 2K21/ISY/24 (Information Systems), Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place : Delhi

Date:

Dr. Jasraj Meena

SUPERVISOR

Assistant Professor

Department of InformationTechnology

Delhi Technological University

ACKNOWLEDGEMENT

I am grateful to Prof. Dinesh Kumar Vishwakarma, HOD (Department of Information and Technology), Delhi Technological University (Formerly Delhi College of Engineering), New Delhi, and all other faculty members of our department for their astute guidance, constant encouragement, and sincere support for this project work.

I would like to take this opportunity to express our profound gratitude and deep regard to our project mentor Dr Jasraj Meena, for his exemplary guidance, valuable feedback, and constant encouragement throughout the duration of the project. His valuable suggestions were of immense help throughout our project work. His perspective and criticism kept us working to make this project in a much better way. Working under him was an extremely knowledgeable experience for us.

We would also like to give our sincere gratitude to all our friends for their help and support.

Sugandh M

ABSTRACT

Software Defined Networking (SDN) has emerged as a revolutionary approach to network management and configuration, yet it also presents new vulnerabilities and avenues for Distributed Denial of Service (DDoS) attacks. Effective detection and mitigation of such attacks are crucial for maintaining network integrity and reliability. In this project we use the facilities available in machine learning to classify DDos attacks in SDN based networks.

Utilizing a comprehensive dataset, generated with the Mininet emulator and containing over 100,000 instances of both benign and malicious traffic, we trained several machine learning models to classify network traffic. This dataset, uniquely tailored to SDN networks, contains 23 extracted and calculated features providing a detailed view of network events.

This research provides valuable insights into the application of machine learning techniques in the detection and classification of DDoS attacks in SDN networks. The findings contribute to ongoing efforts to enhance network security, presenting efficient and robust machine learning models that can be used to safeguard SDN environments from DDoS attacks.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

In the age of interconnected systems and the increasing reliance on digital infrastructures, the resilience and security of our networks have never been more critical. As technological advancements shape the landscape of networking, Software Defined Networking (SDN) has emerged as a transformative approach. SDN decouples the network's control plane from the data plane, allowing for more agile and centralized control over network traffic. However, with this evolution in networking paradigms, new challenges and vulnerabilities have come to light, particularly in the realm of network security. Distributed Denial of Service (DDoS) attacks, among the most disruptive and frequent cyber threats, have now found a new target in SDN environments.

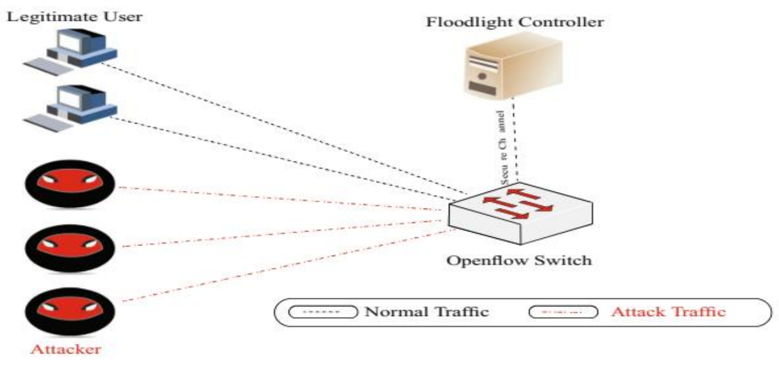


Figure 1.1: DDos Attack Schema in SDN

DDoS attacks flood networks with an overwhelming amount of traffic, thereby disrupting services and inhibiting legitimate users' access. In SDN networks, these attacks can exploit vulnerabilities in the centralized control plane, potentially crippling network operations. The pressing nature of this threat underscores the need for effective and proactive strategies to detect and mitigate DDoS attacks in SDN environments.

The burgeoning field of machine learning offers promising solutions to this problem. Machine learning algorithms can analyze network traffic, learn from it, and identify abnormal patterns indicative of a DDoS attack. Leveraging this capability to detect and classify DDoS attacks contributes significantly to fortifying SDN network security

We explore the use of several machine learning models in this study. Using a rich dataset specifically tailored for SDN networks, we aim to identify the most efficient and accurate models for this crucial task. The ensuing discussion provides insights into the performance of these models, paving the way for more secure and resilient SDN networks.

1.1 PROBLEM STATEMENT

In the current digital era, the exponential growth of internet-based services and interconnected devices has made systems increasingly vulnerable to cyber-attacks. Among these threats, Distributed Denial of Service (DDoS) attacks have been identified as one of the most potent, causing significant disruption to online services and resulting in considerable financial losses. Therefore, the problem statement for this research work can be stated as follows:

"The need for an effective and efficient system for the detection and mitigation of DDoS attacks is critical. While traditional detection methods have shown certain levels of efficiency, they often fall short when dealing with sophisticated, dynamic, and large-scale DDoS attacks. The application of machine learning models for DDoS detection promises better results; however, the performance, effectiveness, and real-time applicability of these models remain unclear and require comprehensive investigation. Thus, the challenge lies in exploring and evaluating the suitability and performance of various machine learning models in identifying DDoS attacks, and subsequently, in developing an optimized model that can handle the evolving nature of these cyber threats with high accuracy and low latency."

1.2 MOTIVATION

With the digitization of numerous aspects of our lives and businesses, network systems have emerged as the backbone of modern society. Consequently, network security has become a critical concern. Among various network threats, Distributed Denial of Service (DDoS) attacks pose a significant challenge due to their potential to cause substantial disruption and damage. These attacks flood networks with enormous volumes of traffic, disrupting services and denying access to legitimate users.

In the modern era of networking, Software-Defined Networking (SDN) is being widely adopted due to its promise of efficiency, flexibility, and centralized control. SDN separates the network's control plane from the data plane, providing improved network management and making network changes more dynamic. However, this paradigm shift also introduces new vulnerabilities, particularly to DDoS attacks, as the central controller becomes an attractive target.

Despite the advances in traditional security defense mechanisms, the increasing sophistication and evolving nature of DDoS attacks have rendered these methods less effective. This highlights the urgent need for innovative, intelligent, and adaptive techniques to protect our networks.

Machine learning, with its ability to learn and adapt from data, offers immense potential in this respect. Various machine learning algorithms can be employed to analyze network traffic patterns and detect anomalies that may indicate a DDoS attack. However, the performance of these algorithms can vary significantly based on the complexity of the problem, the nature of the dataset, and the specific use case.

Therefore, the motivation for this research lies in addressing this crucial gap in understanding. The aim is to investigate and compare the performance of various machine learning models for DDoS attack detection in SDN environments. Through this exploration, we aim to pave the way towards building more robust, secure, and resilient network systems capable of effectively tackling DDoS threats. By identifying the most efficient models, we hope to contribute to the development of improved, machine learning-driven defense mechanisms for the SDN environments of the future.

CHAPTER 2: BACKGROUND AND OVERVIEW

2.1 LITERATURE REVIEW

Distributed Denial of Service (DDoS) attacks have been a topic of research for many years, given their ability to disrupt services and compromise network security. With the advent of machine learning (ML) techniques, the focus has shifted towards utilizing these methods for DDoS attack detection and prevention.

Tavallaee et al. [1] published an exhaustive survey on network intrusion detection and data mining. Their work emphasized the importance of machine learning algorithms in the detection of network attacks and highlighted the effectiveness of these techniques when used alongside traditional intrusion detection systems (IDS).

Sommer and Paxson [2] identified the limitations of conventional signature-based intrusion detection systems in coping with evolving network threats. They suggested the use of machine learning methods to construct models capable of generalizing from known attacks to detect new ones. Their work set the foundation for future research in ML-based network intrusion detection.

Several studies have applied machine learning techniques specifically to DDoS attack detection. Bhuyan et al. [3] proposed a ML-based DDoS detection framework, utilizing multiple traffic features for detection. Their research demonstrated that the use of ML classifiers significantly improves the detection accuracy compared to traditional IDS methods.

In a similar vein, Mirsky et al. [4] introduced Kitsune, an online network attack detection system that utilizes ensemble learning. Their model was successful in detecting DDoS attacks even in the presence of adversarial noise.

More recently, with the evolution of Software Defined Networking (SDN), the landscape of DDoS attack detection has transformed. Sharafaldin et al. [5] introduced an SDN-specific dataset for evaluating the performance of ML algorithms in detecting DDoS attacks. Their research emphasized the need for SDN-specific solutions given the unique vulnerabilities of SDN architecture.

Zeb et al. [6] also recognized the challenges associated with DDoS attack detection in SDN environments. They proposed a Convolutional Neural Network (CNN)-based approach that proved to be effective in detecting DDoS attacks in SDN.

Despite these advancements, the challenge of DDoS attack detection in SDN persists due to the evolving nature of network threats. The motivation for our current research lies in this context. By comparing the performance of various machine learning models on an SDN-specific DDoS dataset, we aim to contribute to the ongoing efforts in enhancing network security in the era of SDN.

2.2 LITERATURE GAP

Especially in real-time circumstances, there is currently a gap in the literature about the development of machine learning models that can successfully detect and mitigate Distributed Denial of Service (DDoS) assaults. While some research has been done on the use of machine learning for DDoS detection, the majority of these studies have drawbacks including poor accuracy, large false positive rates, or exclusively focused on a certain kind of attack. The dynamic nature of “DDoS attacks”, which can alter in real-time and vary over time, has not been adequately researched. More research is required on the best ways to create machine learning models that can deal with huge traffic volumes in high-speed networks and operate with minimal latency to guarantee real-time detection and response. In order to increase the precision and dependability of “DDoS detection”, future research might concentrate on creating more sophisticated machine learning approaches, such deep learning, and fusing them with conventional methodologies. “Real-world DDoS attack datasets, benchmarks, and assessment metrics” are required in order to compare the effectiveness of various detection methods

2.3 OVERVIEW OF ALGORITHMS

In this project, we employed a variety of machine learning algorithms, each exhibiting distinct characteristics and performance traits. The following outlines these models:

Naive Bayes Classifier: An intuitive, probability-based model that estimates class probabilities through Bayes' theorem. The term 'naive' stems from its assumption that all features are independent. While simplistic, this model often performs surprisingly well on complex datasets, offering a baseline for comparison with more advanced models.

Decision Tree: A non-parametric model renowned for its transparency and interpretability. It generates a tree-like model of decisions based on feature values. In essence, it embodies a sequence of 'if-then-else' rules that can be easily understood and visualized. Despite its simplicity, decision trees can effectively handle non-linear relationships and interactions between features.

Deep Neural Networks (DNN): This model represents the forefront of artificial intelligence research. A DNN consists of multiple layers of nodes (neurons), with each layer learning to transform its input data into a slightly more abstract representation. Through this layer-wise abstraction, DNNs can learn complex patterns from high-dimensional data.

Stochastic Gradient Descent (SGD) Classifier: This model implements a linear approach to classification, employing SGD as an efficient optimization algorithm. The classifier is particularly suitable for large-scale and sparse datasets, given its efficiency and ease of implementation. It is, however, sensitive to feature scaling and hyperparameter settings.

K-Nearest Neighbors (KNN): An instance-based, intuitive algorithm that classifies a new instance based on the 'majority vote' of its 'k' most similar instances in the training dataset. Despite its simplicity, KNN can capture complex decision boundaries, making it a strong contender in various classification tasks.

Support Vector Machine (SVM): A strong and adaptable supervised learning technique that works well for both classification and regression. It creates a hyperplane or group of hyperplanes that may be used for classification, regression, or outlier identification in a high- or infinite-dimensional space. SVM is useful for modelling a variety of data structures and efficient in high-dimensional regions.

Random Forest: This ensemble learning technique builds several decision trees during the training phase, and then outputs the class that represents the mean of the classes (classification) or means prediction (regression) of the individual trees. It corrects the tendency of decision trees to overfit their training set.

XGBoost Classifier: eXtreme Gradient Boosting (XGBoost) is a powerful implementation of gradient boosting machines designed to prioritize both computational speed and model performance. It incorporates numerous advanced capabilities, including efficient handling of missing values, tree pruning, and regularization techniques to prevent overfitting.

The combination of these models allowed for a comprehensive and comparative study on the performance of each model with the given dataset and the unique challenges it presents. It also assisted in identifying the most effective approach(es) to detect DDoS attacks in SDN environments.

2.4 BENEFITS AND LIMITATIONS

Naive Bayes Classifier:

- *Benefits*:
    - It's easy to implement and efficient to run, making it ideal for large datasets.
    - It performs well with multi-class prediction problems.
    - It's not sensitive to irrelevant features.
- *Limitations*:
    - It assumes that all predictors (or features) are independent, which is rarely the case in real life.
    - It has a high bias when there's a small amount of data.

Decision Tree:

- *Benefits*:
    - They are easy to understand and visualize, which is valuable for interpreting model predictions.
    - They can handle both numerical and categorical data.
    - They can handle multi-output problems.
- *Limitations*:
    - Decision trees can easily overfit or underfit the data if not properly tuned.
    - They can become extremely complex, which could lead to poor prediction performance.
    - Small variations in the data can lead to a completely different tree, implying high variance.

Deep Neural Networks (DNNs):

- *Benefits*:
    - They can model complex non-linear relationships.
    - They have a high predictive power due to their flexibility and capacity.
    - They can handle large datasets and high dimensional inputs.
- *Limitations*:
    - They require a large amount of data to train.
    - They are computationally expensive and require high-end hardware resources.
    - The model decisions are usually hard to interpret (known as "black box" models).

Stochastic Gradient Descent Classifier (SGDC):

- *Benefits*:
    - It's efficient and easy to implement, suitable for large-scale and sparse machine learning problems.
    - Flexibility in modeling: it can be used for a wide range of applications as it supports different loss functions and penalties.
- *Limitations*:
    - Requires careful preprocessing of the data and tuning of the hyperparameters.
    - Sensitivity to feature scaling.

K-Nearest Neighbors (KNN):

- *Benefits*:
    - It's simple and effective, making no prior assumptions about the form of the function mapping from input features to output class labels.
    - The model doesn't need to be trained, as the classification is done based on the entire dataset.
- *Limitations*:
    - It's computationally intensive and requires a lot of memory, as the entire dataset needs to be stored.

    - The accuracy can be severely degraded by the presence of noisy or irrelevant features.

Support Vector Machine (SVM):

- *Benefits*:
    - It has a high accuracy and strong theoretical foundations.
    - This approach proves to be effective in situations where the dimensionality is high and there are more dimensions than the available samples.
- *Limitations*:
    - It's memory-intensive, slower to train, and difficult to tune due to the importance of picking the right kernel.
    - It does not provide probability estimates.

Random Forest:

- *Benefits*:
    - It has high accuracy, robustness, and ease of use.
    - It can handle missing data and maintains accuracy even when a large proportion of the data are missing.
- *Limitations*:
    - They're not as easy to visually interpret as decision trees.
    - They can overfit datasets that are particularly noisy.

XGBoost Classifier:

- *Benefits*:
    - It's robust and offers several tuning parameters that make the function fit very flexible.
    - It has built-in capabilities for handling missing data.
- *Limitations*:
    - It can be prone to overfitting if not properly tuned.
    - Computationally intensive and requires careful tuning of parameters.

CHAPTER 3: PROPOSED TECHNIQUE

3.1 RESEARCH GAP

Several machine learning models have been used for DDoS attack detection, each with their respective limitations. Here are some of them, along with relevant references:

1. **Support Vector Machines (SVMs):** While SVMs are quite efficient in binary classification tasks, they tend to perform poorly in multiclass problems, which are common in network intrusion detection. SVMs also suffer from high computational cost, especially for larger datasets. Therefore, it becomes infeasible to use SVMs for real-time DDoS detection in big data environments [7].

2. **Artificial Neural Networks (ANNs):** ANNs, including Deep Neural Networks (DNNs), have demonstrated high accuracy in DDoS attack detection. However, they require large amounts of data and substantial computational resources for training. This can be prohibitive in a real-time detection environment. Moreover, they can be susceptible to adversarial attacks, where slightly perturbed inputs can lead to misclassifications [8].

3. **Random Forests:** Random Forests have been used effectively in detecting DDoS attacks. However, they may suffer from overfitting if the number of trees is not correctly optimized. Furthermore, Random Forests can be computationally intensive and slow, particularly with large datasets, making them unsuitable for real-time detection [9].

4. **K-Nearest Neighbors (K-NN):** K-NN's performance largely depends on the choice of 'k' and the distance metric, which can be challenging to optimally determine in a dynamic network environment. It also suffers from high computational cost due to the necessity to compute the distance to all training samples, which makes it less suitable for real-time DDoS attack detection [10].

Please note that these limitations motivate the ongoing research towards improving existing methods and developing new, more efficient models for DDoS attack detection.

3.2 COMPARISON OF RELATED WORK

ML approaches were employed to address the accuracy concerns that earlier IDS had when using ANN with fuzzy clustering. They reduced the size of each training batch by making the training dataset homogenous. “J48 trees, MLP, and BN classifiers” were utilised . They don't employ feature extraction to remove unneeded, obsolete, or irrelevant attributes.

[11] utilised voting classification to combine supervised and unsupervised ML outcomes using ensemble-based ML. Improves IDS accuracy and reliability. Since Kyoto2006+ is more appealing than widely used, out-of-date datasets, it was utilised to evaluate their work. Despite a high false-positive rate, their work is accurate.

Real-time hybrid IDS was suggested [10]. Signature-based detection found known intrusions and anomaly detection found new threats. This study had a good detection rate because anomaly detection discovered threats that evaded the signature-based strategy. By the final day of the study, the algorithm's precision had improved each day to 92.65%, and the number of false negatives had fallen.

[12] found that anomaly-based IDS can increase FPR performance. The NSL-KDD dataset was used. “AdaBoost and XGBoost” When accuracy is high, hybrid or ensemble ML classifiers are used to increase IDS efficacy.

Lack of feature extraction has hampered attempts to reduce execution times and improve detection rates. ML techniques and attribute extraction approaches were used to test ML models on the NSL-KDD [13]. Due to the model's strong FPR and concentration on signature-based threats, new zero-day attacks go unnoticed. Past studies seldom compared models to diverse datasets. suggested a new IDS using feature extraction. The work improves intrusion detection by combining the ensemble classifier with specific features. This analysis used NSL-KDD, IDS2017-CIC, and AWID datasets. Characteristics were collected using CFS-BA.

Ensemble-based methods improve multi-class classification. The model's accuracy was greatest against the AWID dataset.

The utilization of scaled conjugate gradient and Bayesian regularization techniques was applied in [15] to train the artificial neural network (ANN)-based intrusion detection system (IDS). The breadth and quality of the job were evaluated using several measures. FFANN enhanced accuracy by 98,074%. Testing the model on various datasets will increase work reliability. The work of combines four different algorithms, “RNN-LSTM, Bay Classifier, Decision Tree, and Random Forest” are combined [16] in an ensemble model. By selecting the most effective characteristics for identifying intrusions and notifying system administrators of whether the traffic is lawful or illegal, the study contributes to the field by managing an imbalanced dataset. Despite the approach's ok NSL-KDD performance, an experimental research on the most recent datasets is still required.

A single machine learning classifier was utilised in the study of [17] to create an IDS. They employed DT and RF methods, which the NSL-KDD dataset was utilised to evaluate. The decision tree performs worse than the random classifier in terms of accuracy and yields inferior results. The study does not address the issues with the detection rate or the FPR.

a research on the IDS that [18] suggested entailed assessing the “NSL-KDD and UNSWN B-15” datasets using KNN and Random Committee. the usage of just the attribute subsets most relevant to the given datasets as a consequence of doing a feature extraction. The study's findings show that KNN is less efficient than the Random forest technique. The challenges of enormous data size, data imbalance, and conventional IDS algorithm efficiency must be the focus of future study.

The suggested approach by Ponthapalli et al. employed a single classifier to detect network intrusion. The algorithms employed were “SVM, LR, RF, and DT” [19]. The work was evaluated using the NSL-KDD dataset. According to the study, the random forest classifier is the most efficient way to operate the intrusion detection system. The RF algorithm has the fastest execution time, they also found. The study's drawback is that there is only one dataset that can be utilised to assess it properly.

A research employing heterogeneous datasets and a stacking ensemble approach was described in [20]. The “LR, KNN, SVM, and RF” components of the ensemble technique. This study makes use of the “UNSW NB-15 and UGR'16” databases, two recent datasets. In contrast to UGR '16, which was created in a real-world data traffic environment, UNSW NB-15 was created using a simulation [20]. The method generated the best accuracy and increased the IDS's estimation accuracy and detection speed. A lot of datasets that cover the most current assault kinds need to be the subject of additional investigation.

Combining attribute extraction, neural networks, and clustering technique was used. Additionally integrated with SVM was K-means. The outcomes demonstrated that the various ML types complement one another effectively and enhance IDS's performance. To achieve the highest accuracy, attribute extraction is combined with support vector machines and K-means. More research using enhanced hybrid ML algorithms is needed to reduce the FPR.

3.3 HIGH LEVEL ARCHITECTURE

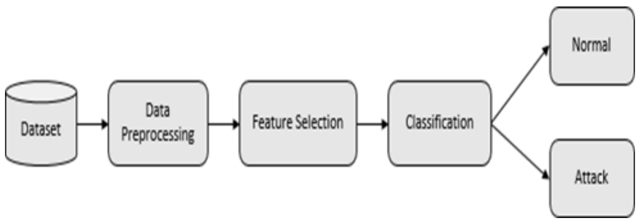


Figure 3.1: A block diagram of high level architecture of the system

Here's a high-level overview of the code's architecture:

1. **Data Loading:** The notebook begins by loading the necessary libraries and the dataset using the pandas library. The dataset is assumed to be in CSV format.
2. **Data Preprocessing:** The data is then preprocessed by converting categorical values into numerical ones using LabelEncoder from sklearn.preprocessing. Following this, any NaN values are filled using the 'ffill' method. The data is then normalized using MinMaxScaler from sklearn.preprocessing to ensure that all features contribute equally to the model. The dataset is then split into training and test sets.
3. **Model Building:** Several machine learning models are built and trained on the preprocessed data. The models include:
    - Deep Neural Network (DNN) using keras library
    - Naïve Bayes using GaussianNB from sklearn.naive_bayes
    - Decision Tree using DecisionTreeClassifier from sklearn.tree
    - K-Nearest Neighbors (KNN) using KNeighborsClassifier from sklearn.neighbors
    - Stochastic Gradient Descent (SGD) using SGDClassifier from sklearn.linear_model
    - Logistic Regression using LogisticRegression from sklearn.linear_model
    - Random Forest using RandomForestClassifier from sklearn.ensemble

    - XGBoost using XGBClassifier from xgboost

   For each model, the process is the same: the model is defined, fitted with the training data, and then used to predict the test data.

4. **Evaluation:** Each model's performance is evaluated using accuracy, confusion matrix, and classification report from sklearn.metrics. The accuracy of each model is compared and displayed in a bar plot for easy visualization.

This high-level architecture serves as the foundation for the Machine Learning-based DDoS detection system implemented in the project.

CHAPTER 4: EXPERIMENT AND ANALYSIS

Expanding on that, let's delve into the approach that will be adopted for this project work:

The system we will be using for running these models is equipped with a powerful Intel Core i5 9th Generation Processor. Accompanied by an 8GB RAM, it provides the necessary computing power to handle large-scale data processing and machine learning tasks. The system's storage capacity is robust, standing at 512GB, allowing sufficient space for dataset storage, model storage, and any additional necessary files.

Our primary programming environment will be Google Colab, a cloud-based Python development environment that offers free GPU support. This allows us to tap into the additional computational power needed to train complex models. Our code execution in Colab, with GPU settings enabled, is estimated to take around 1 hours for the entire program.

Data analysis is the crux of our methodology, a stage where we evaluate, clean, transform, and model data to extract valuable information. This process aims to discover useful information, suggest conclusions, and provide support for decision-making.

The complexity and presentation of data necessitate thorough analysis. This involves validating the accuracy of the data, ensuring that all required variables are present, and handling any outliers or missing values. By sifting through the dataset, we will distinguish valuable data from extraneous information, preserving the 'best' from the 'waste' produced by the process.

Our approach emphasizes rigorous data analysis to ensure the development of high-quality machine learning models for DDoS attack detection. In doing so, we aim to maximize the accuracy and reliability of our findings, contributing meaningful insights to the field of cyber security.

4.1 DATASET

This study utilized a specialized Software-Defined Networking (SDN) dataset designed to classify network traffic, a crucial resource for both deep learning and machine learning algorithms. This dataset was generated through the employment of the Mininet emulator, with the construction of ten unique network topologies linked to a single Ryu controller.

The generation of this dataset involved network simulation, capturing both benign (non-threatening) and malicious network traffic. The benign traffic consisted of UDP, ICMP, and TCP traffic. Conversely, malicious traffic was collected from various types of network attacks, including TCP Syn attack, ICMP attack, and UDP flood attack.

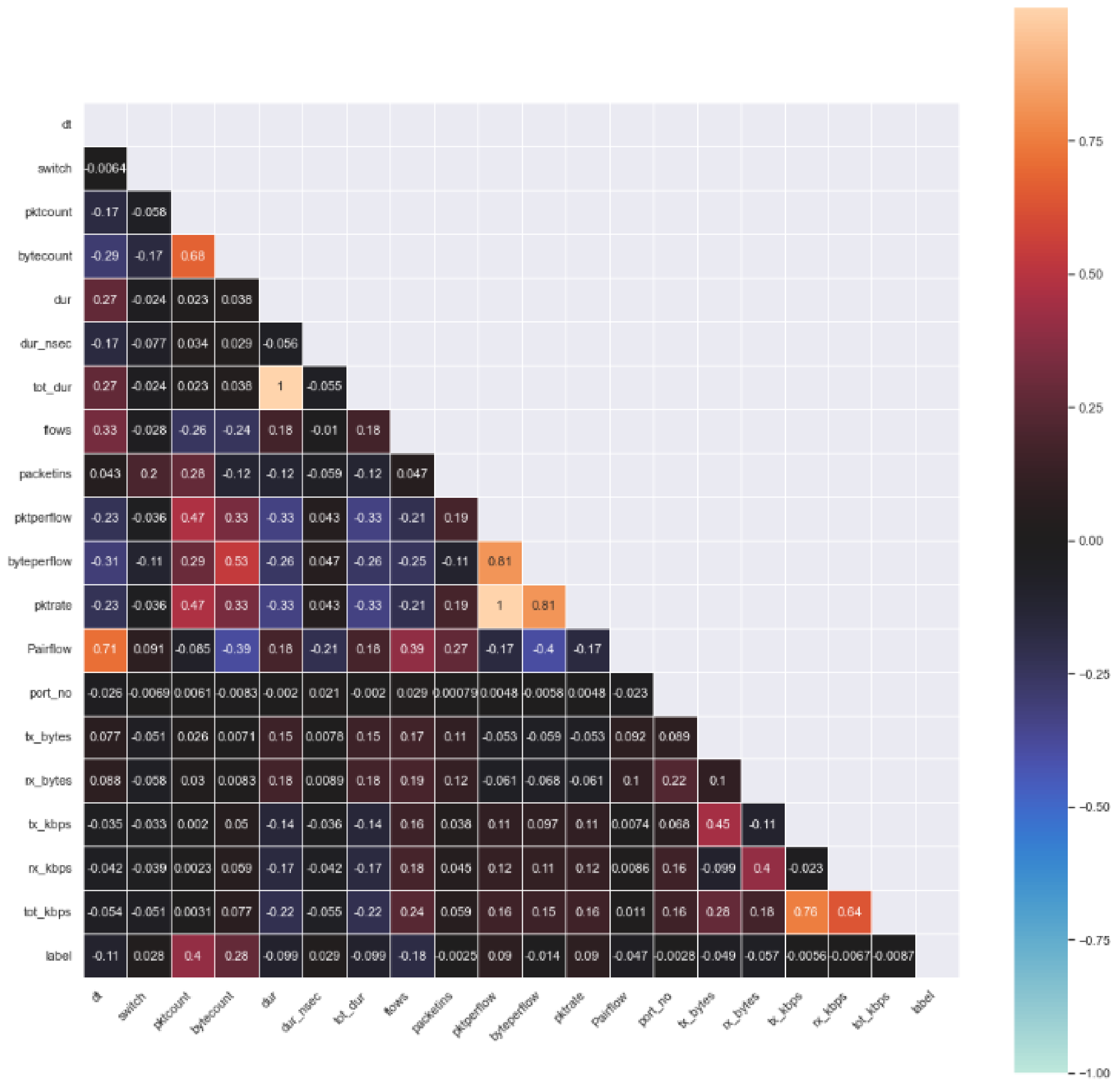
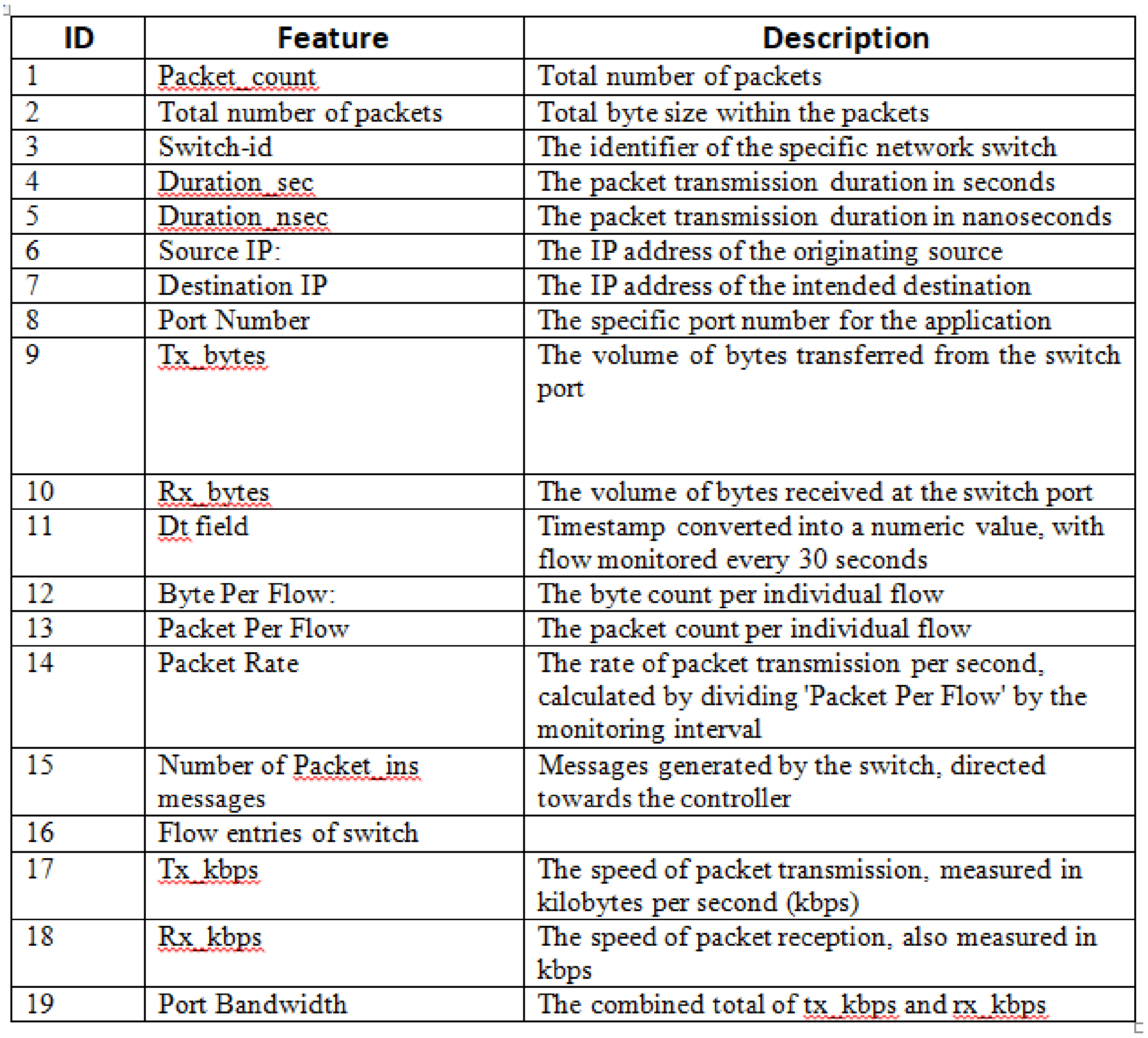


Figure 4.1: Heat map of correlation of features

This comprehensive dataset comprises 23 distinct features. Some of these are directly extracted from the network switches, while others are derived through calculated measures. The extracted features are:



ID	Feature	Description
1	Packet_count	Total number of packets
2	Total number of packets	Total byte size within the packets
3	Switch-id	The identifier of the specific network switch
4	Duration_sec	The packet transmission duration in seconds
5	Duration_nsec	The packet transmission duration in nanoseconds
6	Source IP:	The IP address of the originating source
7	Destination IP	The IP address of the intended destination
8	Port Number	The specific port number for the application
9	Tx_bytes	The volume of bytes transferred from the switch port
10	Rx_bytes	The volume of bytes received at the switch port
11	Dt field	Timestamp converted into a numeric value, with flow monitored every 30 seconds
12	Byte Per Flow:	The byte count per individual flow
13	Packet Per Flow	The packet count per individual flow
14	Packet Rate	The rate of packet transmission per second, calculated by dividing 'Packet Per Flow' by the monitoring interval
15	Number of Packet_ins messages	Messages generated by the switch, directed towards the controller
16	Flow entries of switch	
17	Tx_kbps	The speed of packet transmission, measured in kilobytes per second (kbps)
18	Rx_kbps	The speed of packet reception, also measured in kbps
19	Port Bandwidth	The combined total of tx_kbps and rx_kbps

Table 1: Dataset features and description

The final feature of the dataset, termed the 'class label,' serves as the output and classifies the nature of the network traffic as either benign or malicious. Benign traffic is labelled '0,' while malicious traffic is marked as '1.'

During the network simulation, approximately 250 minutes were spent collecting data, resulting in 104,345 instances of recorded data. To enrich the dataset, the simulation was run intermittently to gather additional instances of data. This extensive dataset provides a rich foundation for testing the effectiveness of different machine learning and deep learning models in identifying and classifying DDoS attacks within an SDN environment.

1. Accuracy

In the field of machine learning, accuracy refers to the measure of how well a predictive model can correctly classify or predict instances within a given dataset. It is an essential evaluation metric used to assess the performance and reliability of machine learning algorithms. Accuracy is typically expressed as a percentage and is calculated by dividing the number of correctly predicted instances by the total number of instances in the dataset. The resulting value represents the proportion of correct predictions made by the model.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{Total\ Predictions}$$

2. Precision, Recall, and F1-Score

Precision: can be described as the ratio of correctly classified attacks (TP) with total flows classified as the attack (TP+FP).

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

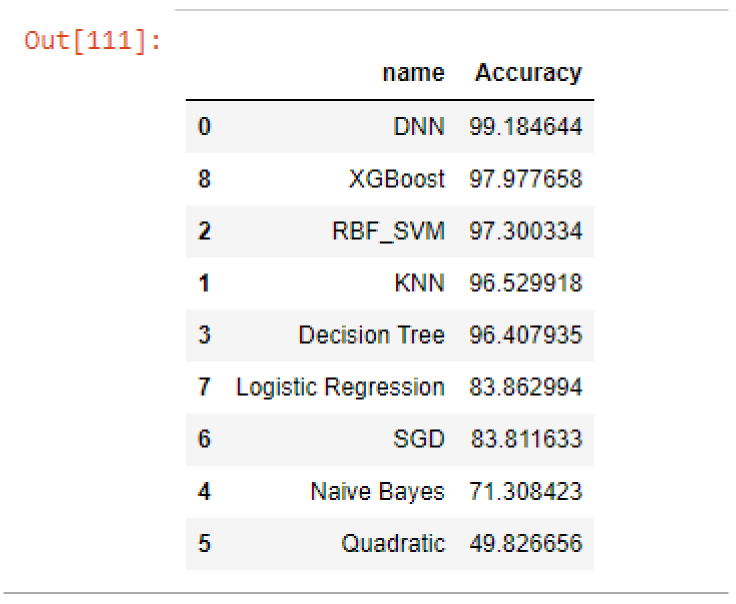
3. F1-Score: is the harmonic mean(HM) calculated by considering precision and recall.

4. $F1 - Score = \frac{2*(Precision*Recall)}{Precision+Recall}$

Please remember that depending upon the problem statement, some metrics might be more important to tune than others. For instance, in a DDoS detection scenario, having high recall might be more important than precision as we would want to capture as many actual attacks as possible, even at the cost of raising some false alarms.

CHAPTER 5: RESULTS

In this study, multiple machine learning and deep learning models were employed to classify network traffic, distinguishing between benign and malicious patterns. The results are summarized as follows:



```
Out[111]:
```

	name	Accuracy
0	DNN	99.184644
8	XGBoost	97.977658
2	RBF_SVM	97.300334
1	KNN	96.529918
3	Decision Tree	96.407935
7	Logistic Regression	83.862994
6	SGD	83.811633
4	Naive Bayes	71.308423
5	Quadratic	49.826656

Figure 5.1: Summarised table of classifiers used with its accuracies

The figure presented consists of various Machine Learning models and their corresponding accuracy scores when applied to a particular dataset for DDoS attack detection. A low score indicates that the model didn't perform well in correctly predicting the target variable, while a high score suggests an excellent predictive performance. Here, we will analyze the accuracy of each model in ascending order, thus providing a comparative analysis.

Starting from the lowest accuracy score, the Quadratic model has an accuracy of 49.826656%. This low performance might be due to its inability to fit the complexities in the dataset accurately. Quadratic models tend to perform poorly when data relationships are more complex or non-linear, which seems to be the case with this dataset.

Next in line is the Naive Bayes classifier with an accuracy of 71.308423%. While better than the Quadratic model, it is still notably lower than other models. Naive Bayes classifiers, although efficient, assume that the features in the dataset are independent of each other, which might not always be the case, thereby affecting its performance.

Moving ahead, the Stochastic Gradient Descent (SGD) and Logistic Regression models both show moderate performances with similar accuracy scores of 83.811633% and 83.862994% respectively. Both these models are general-purpose classifiers suitable for large datasets, but they might struggle with complex, high-dimensional data like the one at hand, which likely resulted in their middling performances.

Following these, we have a significant jump in accuracy with the Decision Tree model at 96.407935%. This model offers higher interpretability and can handle non-linear relationships quite well. Its performance, however, can be compromised due to overfitting if not properly pruned.

Just above the Decision Tree model, the K-Nearest Neighbors (KNN) model demonstrates an accuracy of 96.529918%. This simple yet effective model can achieve good performance, but its accuracy highly depends on the number of neighbors chosen ('K') and the metric used for calculating 'distance'. It also struggles with high-dimensional data, which might explain why it doesn't reach the top of our list.

The Radial Basis Function Support Vector Machine (RBF_SVM) model follows closely, with an accuracy of 97.300334%. The model is powerful in dealing with high-dimensional data and has the flexibility to cater to both linear and non-linear classification problems, thus showing a higher performance.

The XGBoost model has the second-highest accuracy of 97.977658%. As an optimized gradient boosting algorithm, it's renowned for its superior performance, robustness, and speed. Its ability to prevent overfitting and handle high-dimensional, complex data likely contributes to its high accuracy.

Lastly, the Deep Neural Network (DNN) stands as the top-performing model with an accuracy of 99.184644%. With its ability to learn intricate patterns from large volumes of data, the DNN model clearly stands out in predicting DDoS attacks from the dataset, showcasing its superior predictive performance.

In conclusion, the performances of the models reveal that the more complex models (DNN, XGBoost) tend to outperform simpler ones when dealing with a dataset of this nature. However, simpler models could still be useful when interpretability and computational efficiency are critical considerations.

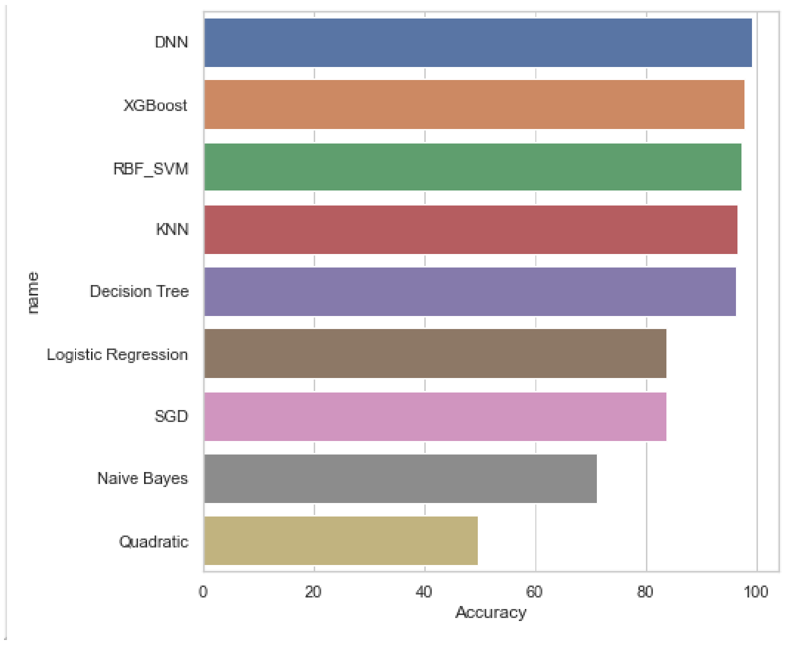


Figure 5.2: A bar plot of accuracies of different classifiers

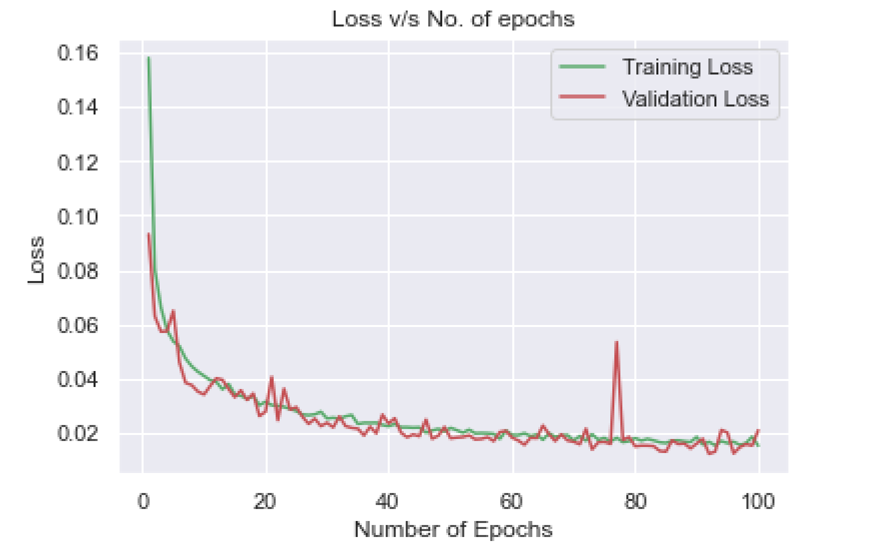


Figure 5.3: Loss vs No of Epochs

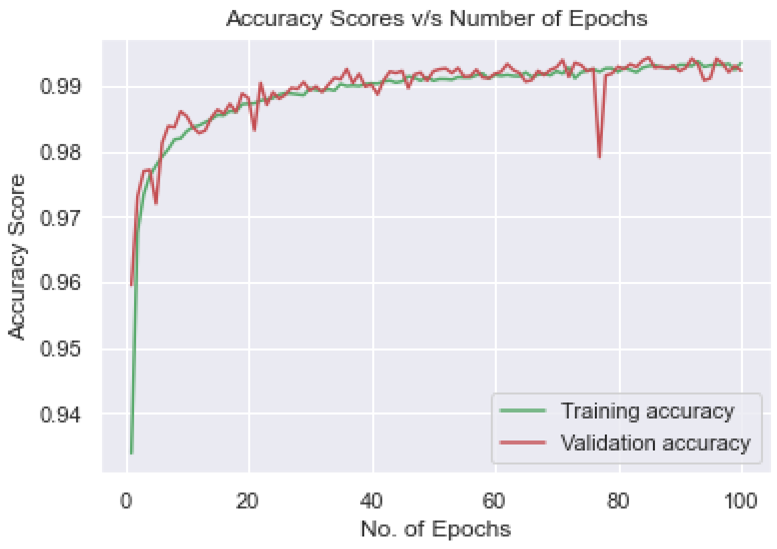
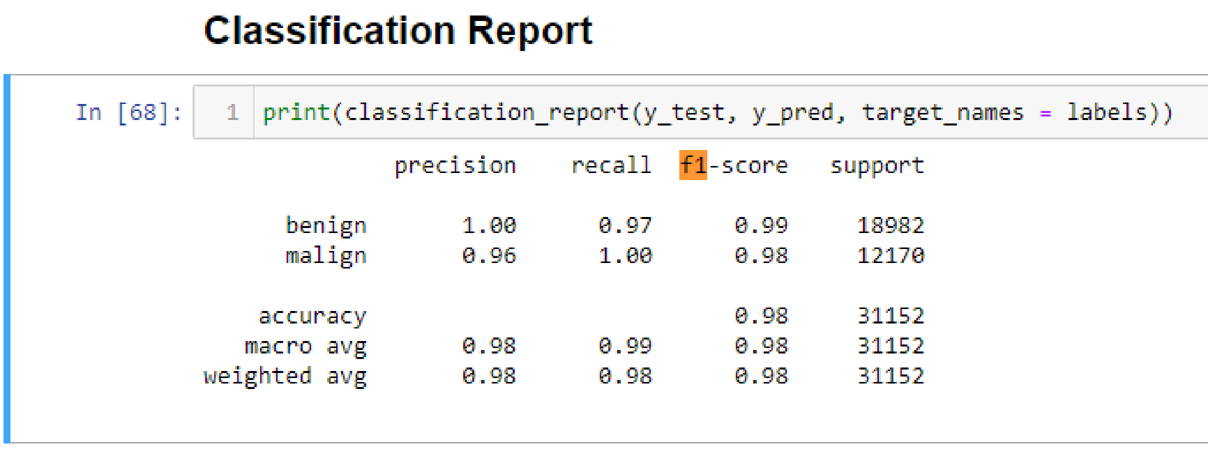


Figure 5.4: Accuracy Score vs No of Epochs

The above figure gives us a graphical representation of loss vs No of Epochs and Accuracy Score vs No of Epochs.

The below figure shows the classification report containing f1-score, precision, recall and support.



Classification Report

```
In [68]:  print(classification_report(y_test, y_pred, target_names = labels))
```

```
              precision    recall  f1-score   support

      benign       1.00      0.97      0.99     18982
      malign       0.96      1.00      0.98     12170

    accuracy                           0.98     31152
   macro avg       0.98      0.99      0.98     31152
weighted avg       0.98      0.98      0.98     31152
```

Figure 5.5: Classification report summary

CHAPTER 6: CONCLUSION

The overarching aim of this study was to evaluate the performance of different machine learning and DN models in classifying network traffic, and specifically in identifying Distributed Denial of Service (DDoS) attacks. As network security continues to be a pressing concern, finding effective methods for detecting malicious activity remains a critical task.

In this study, a total of eight models were employed and their performance assessed in terms of accuracy. The models tested ranged from simpler ones like Naive Bayes and Logistic Regression to more complex and advanced methods like XGBoost and Deep Neural Networks (DNN). The results demonstrated varying degrees of effectiveness, with DNN standing out as the most accurate model, achieving an accuracy of 99.187851%. Other models, such as XGBoost and RBF SVM, also showed promising results with high accuracy scores, outperforming traditional and less complex models.

The significance of sophisticated machine learning and deep learning models in the field of network security is highlighted by these findings. The superior performance of DNN, XGBoost, and RBF SVM demonstrates their ability to handle high-dimensional, complex data and detect intricate patterns within network traffic.

However, while the results are promising everything has its own challenges. The interpretability of such models, especially DNN, is often limited, making it harder to understand the reasons behind their predictions. Additionally, the computational resources required to run these models can be extensive, which may limit their use in some settings.

The study's findings demonstrate the effectiveness of several deep learning and machine learning models in identifying DDoS attacks. It encourages the continued exploration of these models and their improvement, pushing forward our capabilities in network intrusion detection. Yet, it also prompts a consideration of the trade-offs between model complexity, interpretability, and resource requirements, aspects that will continue to be central in the further development and application of these models for network security.

REFERENCES

[1] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2010). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications (CISDA) (pp. 53–58).

[2] Sommer, R., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In Proceedings of the IEEE Symposium on Security and Privacy (pp. 305–316).

[3] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network Anomaly Detection: Methods, Systems and Tools. IEEE Communications Surveys & Tutorials, 16(1), 303–336

[4] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. Network and Distributed System Security Symposium (NDSS).

[5] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. 4th International Conference on Information Systems Security and Privacy (ICISSP), 108–116

[6] Zeb, K., Asif, M., Mahmood, A. N., Ahmad, H. F., & Almogren, A. (2020). SDN-based DDoS Attack Detection using CNN. Computer Networks, 172, 107147.

[7] L. Dhanabal and Dr. S.P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," 2015 International Conference on Advanced Computing and Communication Systems, Coimbatore, 2015, pp. 1-7, doi: 10.1109/ICACCS.2015.7324092

[8] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B. and Swami, A., 2017. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (pp. 506-519).

[9] S. Haider, B. A. Muhammad and S. Shabir, "Random Forest Model for Network Intrusion Detection System," 2019 1st International Conference on Advances in Science,

Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-5, doi: 10.1109/ICASERT.2019.8934567

[10] Li, W., Meng, W., Kwok, L.F., Zhong, W. and Lai, C., 2018, April. An efficient KNN algorithm implemented on FPGA based heterogeneous computing system for large-scale dataset. In 2018 28th International Conference on Field Programmable Logic and Applications (FPL) (pp. 368-3687). IEEE.

[11] S. Gupta, "ANALYZING THE MACHINE LEARNING ALGORITHMS- NAÏVE BAYES , RANDOM TREE , AND SUPPORT VECTOR MACHINES SVM USING THE KDD99 DATA SET TO PREDICT AND CLASSIFY THE," no. 2, pp. 452– 459, 2016

[12] P. Verma, S. Anwar, S. Khan, and S. B. Mane, "Network Intrusion Detection Using Clustering and Gradient Boosting," 2018 9th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2018, pp. 1–7, 2018, doi: 10.1109/ICCCNT.2018.8494186.

[13] A. S. Amira, S. E. O. Hanafi, and A. E. Hassanien, "Comparison of classification techniques applied for network intrusion detection and classification," J. Appl. Log., vol. 24, pp. 109–118, 2017, doi: 10.1016/j.jal.2016.11.018.

[14] D. P. Gaikwad and R. C. Thool, "Intrusion detection system using Bagging with Partial Decision Tree base classifier," Procedia Comput. Sci., vol. 49, no. 1, pp. 92–98, 2015, doi: 10.1016/j.procs.2015.04.231.

[15] W. C. Lin, S. W. Ke, and C. F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," Knowledge-Based Syst., vol. 78, no. 1, pp. 13–21, 2015, doi: 10.1016/j.knosys.2015.01.009.

[16] A. R. Syarif and W. Gata, "Intrusion detection system using hybrid binary PSO and K-nearest neighborhood algorithm," Proc. 11th Int. Conf. Inf. Commun. Technol. Syst. ICTS 2017, vol. 2018-Janua, pp. 181–186, 2018, doi: 10.1109/ICTS.2017.8265667

[17] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," Comput. Networks, vol. 136, pp. 37–50, 2018, doi: 10.1016/j.comnet.2018.02.028.

[18] N. Farnaaz and M. A. Jabbar, “Random Forest Modeling for Network Intrusion Detection System,” Procedia Comput. Sci., vol. 89, pp. 213–217, 2016, doi: 10.1016/j.procs.2016.06.047.

[19] K. Atefi, S. Yahya, A. Rezaei, and S. H. B. M. Hashim, “Anomaly detection based on profile signature in network using machine learning technique,” Proc. - 2016 IEEE Reg. 10 Symp. TENSYMP 2016, pp. 71–76, 2016, doi: 10.1109/TENCONSpring.2016.7519380

[20] B. Brao and K. Swathi, “Fast kNN Classifiers for Network Intrusion Detection System,”April,2017, doi: 10.17485/ijst/2017/v10i14/93690

PAPER NAME

Sugandh Thesis Complete file.pdf

WORD COUNT	CHARACTER COUNT
6796 Words	**38813 Characters**
PAGE COUNT	FILE SIZE
35 Pages	**1.3MB**
SUBMISSION DATE	REPORT DATE
May 30, 2023 6:22 PM GMT+5:30	**May 30, 2023 6:23 PM GMT+5:30**

● 19% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 13% Internet database
- 6% Publications database
- Crossref database
- Crossref Posted Content database
- 16% Submitted Works database

● Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 8 words)