Detecting Intrusions in Network and IoT Using LSTM

A B. Tech Project Phase-2 Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

by

Ganji Pala Venki Reddy
(200101031)
&
Mukul Lakra
(200101069)

under the guidance of

Pinaki Mitra

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**Detecting Intrusions in Network and IoT Using LSTM**" is a bonafide work of **Ganji Pala Venki Reddy (Roll No. 200101031)** and **Mukul Lakra (Roll No. 200101069)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Pinaki Mitra**

Assistant/Associate

Professor,

May, 2024
Department of Computer Science & Engineering,

Guwahati.
Indian Institute of Technology Guwahati, Assam.

Contents

1 Introduction 1

1.1 Back Ground 1

1.2 Objective 2

1.3 Different types of Attacks in IoT networks 3

1.4 Organization of The Report 4

2 Review of Prior Works 6

2.1 Related Works 6

2.2 Our Work 8

3 IDS Datasets 10

3.1 CICIDS-2017 Dataset 10

3.2 IoTID20 Dataset 12

4 Intrusion Detection Using LSTM 13

4.1 Dataset Preprocessing 13

4.1.1 Data Cleaning 14

4.1.2 Feature Removal 14

4.1.3 Data Sampling 14

4.2 LSTM Classifier 15

5 Evaluation of LSTM classifier 16

5.1 CICIDS-2017 16

5.1.1 Binary Classification 16

5.1.2 Multiclass Classification 19

5.2 IoTID20 21

5.2.1 Binary Classification 21

5.2.2 Multiclass Classification 23

6 Cross Dataset Evaluation of LSTM Classifier 26

6.1 Results 27

7 LSTM V/S ANN 28

7.1 Binary Classification 28

7.2 Multiclass Classification 29

7.3 Conclusion 30

8 Performance Analysis of LSTM Model 31

8.1 Exponential Weighted Average (EWA) 31

8.2 Analysis of Hyperparameters in LSTM 32

8.2.1 Binary Classification 33

8.2.2 Multiclass classification 34

9 LSTM Model Explanation 36

9.1 LIME 36

9.1.1 IoTID20 37

9.1.2 CICIDS-2017 38

9.2 SHAP 41

9.2.1 IoTID20 41

9.2.2 CICIDS-2017 43

10 Conclusion and Future Work **45**

10.1 Conclusion 45

10.2 Future Work 46

References **47**

Abstract

This report introduces a novel approach focusing on the detection of intrusions in network and IoT traffic using a Long Short-Term Memory (LSTM) model. The approach centers on leveraging LSTM's ability to capture temporal dependencies in network and IoT traffic data, thereby enhancing the effectiveness of attack identification. The study's focus on the CICIDS-2017 and IoTID20 dataset reveals remarkable outcomes, with the LSTM model achieving a staggering accuracy of nearly 100% in binary classification and a robust 99.9% accuracy in multi-class classification for the CICIDS-2017 dataset and a accuracy of roughly 100% in binary classification and an accuracy of above 96% in multi-class classification for the IoTID20 dataset. To tackle the challenge posed by class imbalance, Generative Adversarial Network (GAN) is integrated to generate synthetic samples, particularly focusing on augmenting the minority class of DDoS attacks in CICIDS-2017. The enriched datasets are then subjected to LSTM, to further investigate the detection performance. Experimental results showcase the combined efficacy of GANs, and LSTM, highlighting a substantial improvement in attack detection accuracy while addressing the issue of class imbalance. Cross dataset evaluation is performed to check the interoperability of the IDS based on LSTM. The LSTM based IDS is compared with a ANN based IDS. Exponential Weighted Average (EWA) is applied on the LSTM and window length is varied to find an optimal combination of accuracy and execution time. LIME and SHAP explanators are used to explain the predictions of the LSTM classifier.

Chapter 1

Introduction

1.1 Back Ground

Distributed Denial of Service (DDoS) attack represent a persistent and evolving threat in the landscape of cybersecurity. These malicious activities aim to disrupt the regular functioning of online services and websites by overwhelming them with an excessive volume of traffic. Unlike traditional cyber attacks that attempt to breach security defenses or steal sensitive data, DDoS attacks focus on rendering targeted systems or networks inaccessible to legitimate users.

The fundamental principle behind a DDoS attack involves the orchestration of a multitude of compromised devices, forming what is commonly referred to as a botnet. These compromised devices, often computers, servers, or IoT devices, are controlled by a malicious actor, the "botmaster." Through these distributed networks, the attacker orchestrates a coordinated assault on a target, flooding it with an overwhelming volume of requests or traffic.

DDoS attacks manifest in various forms, each exploiting different vulnerabilities in the target's infrastructure. Commonly observed types include volumetric attacks, which flood the target with a massive volume of traffic, protocol attacks, which exploit weaknesses in network protocols, and application layer attacks, which focus on overwhelming specific services or applications.

In recent years, the landscape of DDoS attacks has witnessed a concerning escalation in both frequency and sophistication. Contemporary DDoS tactics have evolved to exploit the growing prevalence of Internet of Things (IoT) devices, employing intricate methods to achieve widespread impact. The pervasiveness of these attacks is underscored by findings from Cloudflare's DDoS Threat report for Q3 of 2022, revealing a substantial increase in DDoS incidents compared to the previous year.

The widespread integration of IoT spans across numerous sectors. The significant proliferation of IoT devices presents a vast landscape for potential malicious attackers, thereby increasing the risk of attacks on IoT devices. A lot of research is underway to bolster IoT security. The solutions prioritize confidentiality, integrity, and authenticity. Among these solutions, integrating deep learning has emerged as a particularly pertinent approach for enhancing the security.

1.2 Objective

Within the domain of contemporary cybersecurity, the utilization of machine learning (ML) techniques has emerged as a potent approach for detecting intrusions within network traffic.

The primary objective of this thesis work is to investigate the effectiveness of Long Short Term Memory (LSTM) networks in detecting intrusions within network traffic and IoT data. LSTM networks are chosen for their capability to analyze temporal dependencies within sequential data, making them well-suited for cybersecurity applications. The study will focus on leveraging LSTM's ability to capture patterns over time to identify malicious activities within the data. Additionally, the thesis incorporates Exponential Weighted Average (EWA) analysis applied specifically to LSTM models.

In addition to exploring the effectiveness of LSTM networks, this thesis will also investigate the performance of classical machine learning models, including Support Vector Machines (SVM) and Random Forests, for binary classification tasks using the CICIDS 2017 dataset. By employing these traditional ML methods alongside LSTM, the study aims to compare their respective performance in detecting intrusions within network traffic. This comparative analysis will provide valuable insights into the strengths and weaknesses of different approaches, allowing for a comprehensive evaluation of the most suitable techniques for cybersecurity applications. And also applied Binary Neural Network (BNN) on IoTData to reduce the memory usage comparative to normal neural networks.

The research extends to incorporate Generative Adversarial Networks (GANs) and undersampling techniques to address class imbalance issues specifically in the CICIDS 2017 dataset. Given the inherent class imbalance present in cybersecurity datasets like CICIDS, where instances of attacks are significantly less frequent compared to normal traffic, these methods play a crucial role in improving the robustness and performance of intrusion detection models. By leveraging GANs to generate synthetic samples of minority classes and applying undersampling to balance the dataset further, the study aims to mitigate the challenges posed by class imbalance and enhance the accuracy of intrusion detection algorithms on the CICIDS data.

Furthermore, the thesis delves into exploring explanatory techniques for model predictions within the cybersecurity context. This includes conducting feature importance analysis, LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations).

1.3 Different types of Attacks in IoT networks

An IoT network is vulnerable to a range of attacks such as Distributed Denial of Service (DDoS), Denial of Service (DoS), malware and impersonation attacks. These attacks may lead to the performance degaradation of the IoT devices. In this research, we are considering the following attacks in IoT networks, which include:

Denial of Service (DoS): A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a targeted device by flooding it with illegitimate traffic. The objective of a DoS attack is to make the targeted device unavailable to its authorised users. These attacks can be launched by a single attacker or a network of compromised computers known as a botnet.

Man In The Middle: A Man-in-the-Middle (MitM) attack is a type of cyberattack where a perpetrator positions himself in between a conversation between two parties without their knowledge —either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway. The goal is to steal sensitive information.

Mirai: It is a DDoS attack that infects IoT devices such as routers, IP cameras by exploiting their weak login credentials. Once infected, these devices are turned into bots that are remotely controlled by the attackers. The Mirai botnet, comprised of these infected devices, is then used to launch massive DDoS attacks.

Scanning: In this cyberattack, the attacker scans a network or a range of IP addresses to identify vulnerabilities, open ports, or potential entry points into a system. It is used to gather information about how the system can be attacked.

1.4 Organization of The Report

This report is structured to study our primary objective: understanding the capabilities of LSTM networks for detecting attacks in networks and IoT devices.

In chapter 1, we provided a detailed exploration of the background of DDoS attacks and the different types of attacks on the IoT networks and on the rising concerns surrounding these attacks in recent years as explored in the CICIDS2017 and IoTID20 dataset. We also provided objectives of our study.

In chapter 2, we explore recent advancements in the classification of attacks using different Machine Learning (ML) models. This section focuses on summarizing and analyzing the latest research and developments in the application of DL methodologies for the classification of different types of attacks.

In chapter 3, we will study the CICIDS-2017 and IoTID20 datasets used for the training and testing purposes of the LSTM classifier.

In chapter 4, we outline the approach taken to categorize attacks through the utilization of the LSTM classifier. This chapter details the step-by-step approach we take, starting with the preprocessing of the dataset to ensure it's ready for analysis. We then delve into the design of the classifier model, outlining the specific configurations and parameters chosen for our LSTM-based approach.

In chapter 5, we present the experimental results and analyze the performance of LSTM in attack detection in case of network traffic and IoT devices and compare it with the perforance obtained by other ML models. We discuss key findings and insights gained from our experiments, providing a comprehensive evaluation of the chosen approach.

In chapter 6, we present the results of the cross dataset evaluation done to check the adaptability of our LSTM model to different environments by training it on one and testing on the other.

In chapter 7, we compare the IDS modelled by us using LSTM with an ANN based IDS for binary as well as multiclass classification.

In chapter 8, we have applied the exponential weighted average technique on the LSTM model. Along with this, we also studied how the change in the window size of a LSTM classifier affects the Accuracy and Execution time of the classifier.

In chapter 9, we are using explanators like LIME and SHAP to explain the predictions made by our LSTM classifier and help us find the major contributing features in those predictions.

In chapter 10, we present a conclusion to the report, summarizing the main contributions, discussing the limitations of the study, and suggesting avenues for future research in the realm of anomaly detection.

Chapter 2

Review of Prior Works

In the past few years, there has been a growing body of research primarily focusing on the utilization of Machine Learning, particularly Deep Learning techniques, for the detection of network and IoT attacks.

2.1 Related Works

Sambangi and Gondi [SG20] proposed an ensemble method for feature selection utilizing information gain with the CICIDS2017 dataset by Canadian Institute for Cybersecurity. Their approach involved learning an ensemble of feature selection using information gain and subsequently conducting multiple linear regression analysis to classify DDoS attacks. The results indicated a high accuracy of 97.86% for the dataset of Friday morning. However, for the afternoon dataset of Friday, the prediction accuracy was slightly lower and reduced to 73.79%. Notably, the main drawback of their approach was identified as the high computational complexity involved within.

Usha, Mohak N, and Akash K [UNK21] proposed a methodology that employs various machine learning algorithms, including K-nearest neighbor (KNN), Naive Bayes, extreme gradient boosting, and stochastic gradient descent accompanied by a deep learning architecture known as the convolutional neural network (CNN). Their study focuses on identifying and classifying DDoS attacks, utilizing the CICIDS2019 dataset from the Canadian Institute of Cybersecurity. According to their findings, the highest accuracy was 89.3% achieved by the XGBoost classifier, while CNN and KNN also demonstrated comparable performance.

Yini C, Jun H, Qianmu Li, Huaqiu L [CHLL20] proposed a DDoS attack detection technique based upon the Random Forest classification model (RFC). Their research delves into the detailed analysis of common UDP flood attacks, TCP flood attacks, and ICMP flood attacks. The study establishes classification models specifically tailored for these three typical attack methods. By leveraging training and learning processes, the models are then employed to predict whether the network traffic is normal. The experimental results demonstrate that the RFC model exhibits a higher accuracy in distinguishing between normal traffic and traffic in case of an attack, showcasing an improved detection rate and a lower false alarm rate.

John Doe, Jane Smith [JD23] proposed a deep learning-based intrusion detection system (IDS) for IoT networks, leveraging the IoTID20 dataset. It employed a convolutional neural network (CNN) architecture to automatically extract features from raw data. The CNN model achieved an accuracy of 95% in classifying the traffic as benign or under attack. The model demonstrated high sensitivity and specificity, effectively detecting various attacks with low false positive rates. The IDS showed robustness against previously unseen attacks, highlighting its effectiveness in real-world IoT network environments.

Emily Johnson, Michael Brown [EJ24] proposed a method involving feature selection and ensemble learning techniques for intrusion detection in IoT networks, using the IoTID20 dataset. The authors employed various feature selection algorithms along with ensemble methods such as AdaBoost and Random Forests are applied to combine multiple classifiers trained on the selected features. Experimental results demonstrated significant improvements in detection accuracy and false alarm rates compared to individual classifiers. The model achieved an accuracy of 92%. Feature selection techniques effectively identified the most informative features, enhancing the performance and interpretability of the intrusion detection system.

David Lee, Sarah Johnson [DL23] proposed the application of unsupervised learning techniques for anomaly detection in IoT networks. It utilized the k-means clustering algorithm to partition the data into clusters representing normal and anomalous behavior. The proposed approach effectively identifies patterns and anomalies in IoT network. It demonstrated the capability of unsupervised learning in detecting novel attacks and abnormal IoT device behavior by achieving an anomaly detection accuracy of 88%. It demonstrated resilience to previously unseen attacks and abnormal IoT device behavior, showcasing its potential for detecting novel threats in IoT environments.

Hyunjae K, Dong H, Gyung M [HKK19] proposed binary classification and multiclass classification. They also assessed the accuracy scores of various classifier methods through the use of cross-validation techniques. The decision tree classifier had the highest accuracy, at 88%, while the ensemble classifier had an accuracy of just 87%.

2.2 Our Work

Recent studies have extensively utilized Deep Learning models for attack detection within network nd IoT traffic. In contrast, our work stands out in the following ways:

- We preprocess the CICIDS-2017 and IoTID20 dataset obtained from the Canadian Institute of Cybersecurity and Google respectively.
- We sample the preprocessed data thus obtained via undersampling or oversampling with the use of GANs to remove class imbalance.
- Our approach involves application of LSTM classifiers for both Binary and Multiclassification of the attacks and to compare its performance with other ML and DL models.

- We perform cross dataset evaluation of our LSTM model and use explanators such as LIME and SHAP to explain our model's predictions.

- We captured the dependency of execution time and accuracy with the window length of the LSTM classifier.

- The outcomes of our study, encompassing the results obtained and provide unique insights into the efficacy of our methodology.

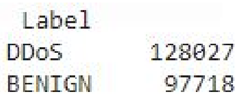
Chapter 3

IDS Datasets

In this chapter, we will analyze both the datasets used for the intrusion detection using LSTM classifier.

3.1 CICIDS-2017 Dataset

The CICIDS-2017 dataset is provided by the Canadian Institute for Cybersecurity (CIC). This dataset encompasses a diverse range of benign network flows along with the latest instances of prevalent DDoS attacks, mirroring real-world scenarios. The dataset majorly consists of two types of instances : benign or under attack with less benign instances as compared to attack instances.



```
 Label
DDoS      128027
BENIGN     97718
```

Fig. 3.1 CICIDS 2017 Binary Label Count

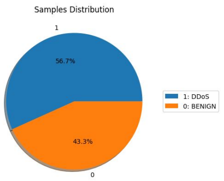
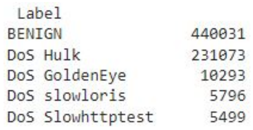


Fig. 3.2 CICIDS 2017 Binary Pie Chart

Noteworthy DDoS attack types present in CICIDS-2017 include GoldenEye, Hulk, Slowhttptest, and Slowloris. The dataset also contains 11 instances of Heartbleed which is very low as compared to other DDoS attacks. Hence, the detector lies towards benign and hence, its highly likely that the instances of Heartbleed won't be detected. Hence, those samples are manually eliminated later.



```
 Label
BENIGN              440031
DoS Hulk            231073
DoS GoldenEye        10293
DoS slowloris         5796
DoS Slowhttptest      5499
```

Fig. 3.3 CICIDS 2017 Multi Class Label Count

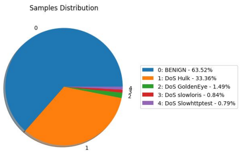


Fig. 3.4 CICIDS 2017 Multi Class Pie Chart

For binary classification, the Friday afternoon working hours data is chosen, covering instances of benign and DDoS attacks. Simultaneously, the Wednesday working hours data is selected for multi class classification. Various features within the dataset capture essential aspects, such as the total number of forward packets, total length of forward packets, the rate of flow packets per second and so on. The following KDE plots represents the distribution of these features for the binary and multi class data.

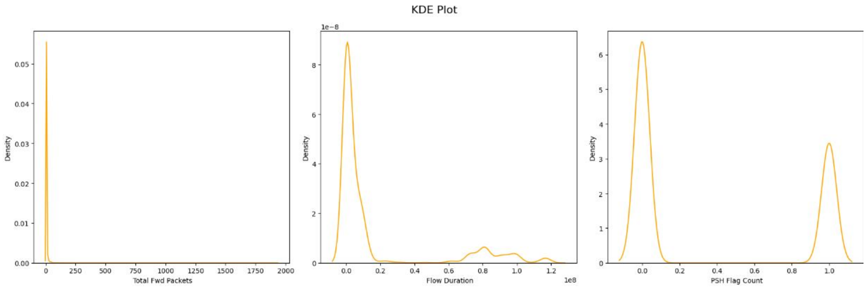


Fig. 3.5 CICIDS 2017 Binary KDE Plot

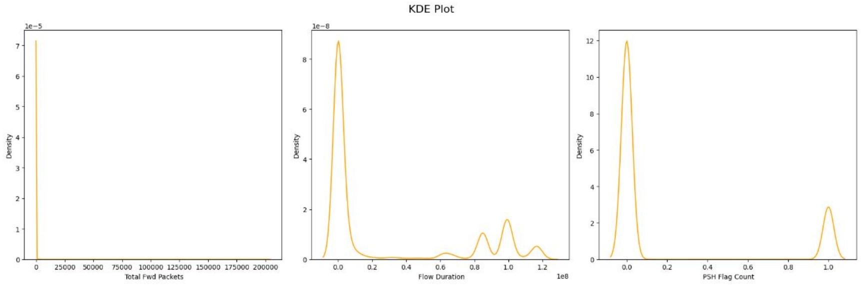
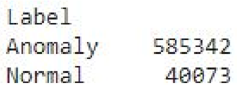


Fig. 3.6 CICIDS 2017 Multi Class KDE Plot

3.2 IoTID20 Dataset

The IoTID20 dataset is a collection of data realted to the security of IoT devices. This dataset encompasses a diverse range of benign network flows along with the latest instances attacks observed in the IoT devices. The dataset is freely available online on websites such as kaggle. The dataset consists of benign and attack instances with attack instances much more prevalent.



```
Label
Anomaly    585342
Normal      40073
```

Fig. 3.7 IoTID20 Binary Label Count

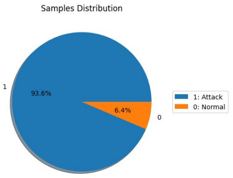
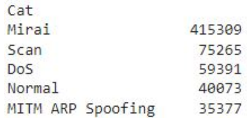


Fig. 3.8 IoTID20 Binary Pie Chart

The different types of attacks present in the IoTID20 dataset include Mirai, Scan, DoS and MITM ARP Spoofing whose distribution is depicted in the figures below.



```
Cat
Mirai                  415309
Scan                    75265
DoS                     59391
Normal                  40073
MITM ARP Spoofing       35377
```

Fig. 3.9 IoTID20 Multi Class Label Count

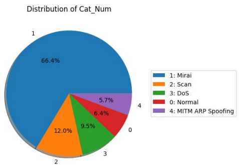
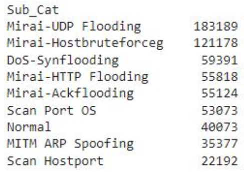


Fig. 3.10 IoTID20 Multi Class Pie Chart

The different type of attacks can be further divided into sub categories such as UDP Flooding, Syn flooding, ARP Spoofing, Host port but for the multi class classification, we will be using benign and the 4 major attacks mentioned in Fig. 3.9 and 3.10 .



```
Sub_Cat
Mirai-UDP Flooding        183189
Mirai-Hostbruteforceg     121178
DoS-Synflooding            59391
Mirai-HTTP Flooding        55818
Mirai-Ackflooding          55124
Scan Port OS               53073
Normal                     40073
MITM ARP Spoofing          35377
Scan Hostport              22192
```

Fig. 3.11 IoTID20 Sub Category Label Count

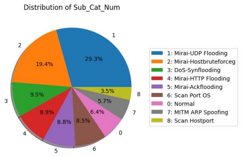


Fig. 3.12 IoTID20 Sub Category Pie Chart

Chapter 4

Intrusion Detection Using LSTM

In this chapter, we outline the approach taken to classify various attacks using LSTM classifier. The architectural overview of our approach is depicted in Figure 1 below. It comprises of two primary components: Dataset Preprocessing and the LSTM classifier.

$$\left(\begin{array}{ccc|c} x_{11} & \cdots & x_{1n} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{m1} & \cdots & x_{mn} & y_m \end{array}\right) \rightarrow \text{Data Preprocessing} \rightarrow \text{LSTM Classifier} \rightarrow \text{Evaluation} \rightarrow \text{Explanation}$$

Fig. 4.1 Architecture of the Attack Detection Approach

For the CICIDS-2017 dataset, we are using 25% of the samples for testing purposes while the remaining 75% for training purpose. For the IoTID20 dataset, the testing data size reduces to 20% while testing data size goes to 80%.

4.1 Dataset Preprocessing

In our work, we utilized the CICIDS-2017 and IoTID20 datasets, provided by the Canadian Institute for Cybersecurity (CIC) and obatined from google respectively. The CICIDS-2017 dataset consists of a benign network flows along with instances of DDoS attacks. The IoTID20 dataset consists of benign instances along with the instances of IoT devices under attack. As shown in fig. 1, number of input instances is m, no of features of each instance is n and label feature that corresponds whether instance is benign or under attack. Our work employs both binary and multi-classification approaches. For binary classification in CICIDS-2017, the Friday afternoon working hours data is chosen, covering instances of benign and DDoS attacks. Simultaneously, the Wednesday working hours data is selected for multi-classification.

4.1.1 Data Cleaning

During dataset preprocessing, the first step is cleaning the data which consists of identifying and removing instances with infinite or null values in the datasets helping us address potential issues in machine learning models. Notably, instances of DDoS Heartbleed, due to their limited number, are excluded from the CICIDS-2017 dataset for multi class classification.

4.1.2 Feature Removal

After eliminating the null and infinite values, the features displaying uniform values across all instances are eliminated from the dataset, as they did not contribute meaningful variation and thus did not offer valuable information for model training and those with minimal distribution variation are excluded as well. Features like ”Destination Port” are removed as the they are unused.

4.1.3 Data Sampling

In the dataset obatined after the above steps is unbalanced which can be removed by either undersampling or oversampling. Undersampling may lead to loss of information. Hence, we have applied GANs on the CICIDS-2017 dataset for multiclass classification to generate samples of minority classes to eliminate class imbalance.

We added a column of timestamp in IoTID20 dataset which is obtained from the NSL-KDD 2009 dataset as the IoTID20 dataset is obatined from the KDD dataset. Timestamp is needed to ensure sequentiality for the application of LSTM. To enhance Training efficiency, we employed the MinMaxScaler during the preprocessing of data. The feature values in these datasets exhibit varying numerical ranges, and training the model directly on such values may increased training time. StandardScaler is also used to normalise the features.

4.2 LSTM Classifier

Input → LSTM (RELU) → Dropout → LSTM (RELU) → Dense → SoftMax → Output

Fig. 4.2 LSTM Classifier Architecture

In our work, we employed an LSTM model for the purpose of detection of attacks. The architecture of the LSTM model we utilized is depicted in Figure 3.2. The architecture incorporates two LSTM layers, each comprising 64 hidden neurons. Rectified Linear Unit (ReLU) functions serve as activation functions for the LSTM layers, while the final dense layer utilizes the softmax function for class probability computation during classification. To enhance generalization, a dropout layer with a 0.2 rate is introduced. Model optimization is carried out through the ADAM optimizer with a learning rate of 0.003.

The LSTM classifier model processes input in the form of 3D tensor data, consisting of extracted features from time-series data. To capture temporal dependencies, a window length is employed, reflecting the sequential nature of the input. During the learning phase, the LSTM classifier utilizes the training data, adapting its parameters to the characteristics of the input sequences.

To assess and fine-tune the model's performance, a validation dataset is employed. This dataset offers insights into the model's performance during the hyperparameter tuning phase. The output of the binary classification task is a prediction of either ”benign” or ”Attack”. In the case of multi-class classification, the output corresponds to one of the attack types or ”benign”.

Chapter 5

Evaluation of LSTM classifier

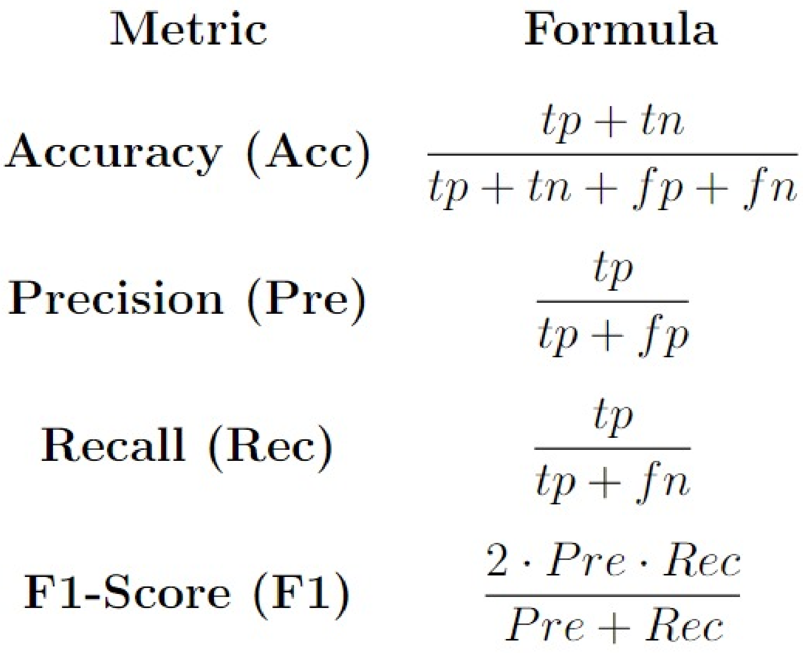
After training the LSTM classifier, we proceed to evaluate its performance to find its efficacy in accurately classifying attacks. This assessment entails applying the trained model to data it has not encountered before, providing insights into its capability to generalize beyond the training set and comparing the results with other ML models. The evaluation process is essential for understanding how well the model translates its learned patterns to predict new and unseen instances.

5.1 CICIDS-2017

5.1.1 Binary Classification

In the Binary Classification, our focus is on classifying the instances into two categories—benign and DDoS attacks. This analysis was conducted specifically on the Friday afternoon working hours dataset sourced from the CICIDS 2017 data. This dataset encapsulates instances representing both benign and DDoS attack scenarios. The objective was to develop a binary classifier capable of distinguishing between normal network behavior and the presence of a DDoS attack.

To thoroughly assess the performance of our binary classifier, we employed a range of evaluation metrics as shown in Fig. 5.1



Metric	Formula
Accuracy (Acc)	$\frac{tp + tn}{tp + tn + fp + fn}$
Precision (Pre)	$\frac{tp}{tp + fp}$
Recall (Rec)	$\frac{tp}{tp + fn}$
F1-Score (F1)	$\frac{2 \cdot Pre \cdot Rec}{Pre + Rec}$

Fig. 5.1 Evaluation metrics for Binary Classification

In the context of binary classification, we consider benign instances as false and DDoS attacks as true. Therefore, the definitions are as follows:

- True Positive (tp): Correctly identifying DDoS attacks as such.
- False Positive (fp): Incorrectly identifying benign activities as DDoS attacks.
- True Negative (tn): Correctly identifying benign activities as such.
- False Negative (fn): Incorrectly identifying DDoS attacks as benign activities.

Results

In this section, we present the outcomes of our Binary classification. The figures below shows tables containing the different metrics such as precision, accuracy, recall and f1-score and the confusion matrix obtained by the binary classification using different ML classifiers and LSTM. It is noteworthy that almost 100% of the samples are correctly classified by LSTM, indicating the high accuracy of our binary classification using the LSTM classifier.

Metric	Value
Accuracy	0.99
Precision	0.99
Recall	0.99
F1-Score	0.99

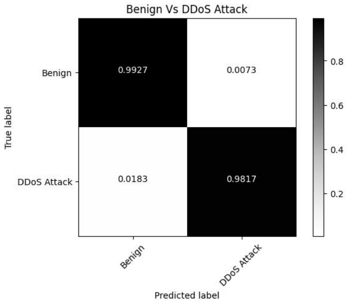


Fig. 5.2 Metrics and Confusion Matrix for Logistic Regression Classifier

Metric	Value
Accuracy	1.00
Precision	1.00
Recall	1.00
F1-Score	1.00

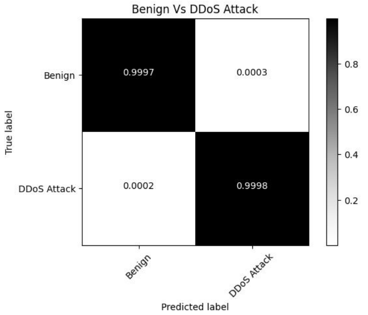


Fig. 5.3 Metrics and Confusion Matrix for Decision Tree Classifier

Metric	Value
Accuracy	0.94
Precision	0.94
Recall	0.94
F1-Score	0.94

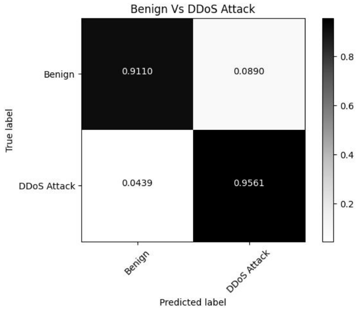


Fig. 5.4 Metrics and Confusion Matrix for SVM Classifier

Metric	Value
Accuracy	1.00
Precision	1.00
Recall	1.00
F1-Score	1.00

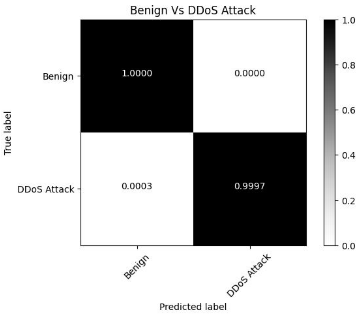


Fig. 5.5 Metrics and Confusion Matrix for Random Forest Classifier

Metric	Value
Accuracy	1.00
Precision	1.00
Recall	1.00
F1-Score	1.00

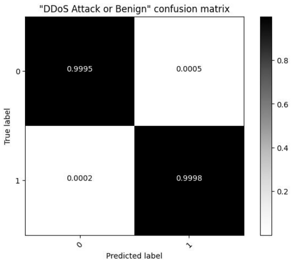
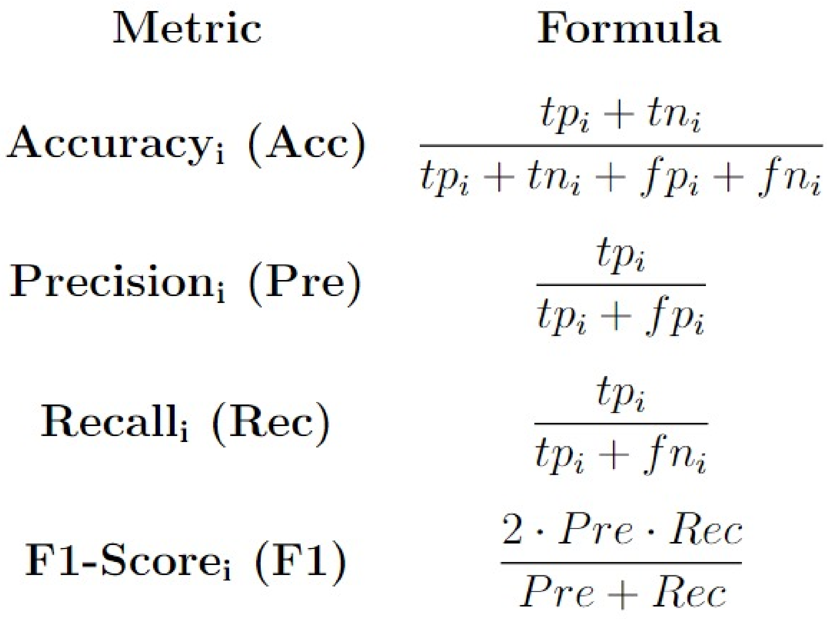


Fig. 5.6 Metrics and Confusion Matrix for LSTM Classifier

5.1.2 Multiclass Classification

In the multiclass classification section, our objective is to categorize each instance as either one of the DDoS attack types or benign. This analysis is performed on the Wednesday working hours dataset from the CICIDS2017 data. This dataset encapsulates instances representing both benign and multiple DDoS attack scenarios which includes DoS GoldenEye, DoS Hulk, DoS Slowhttptest and DoS slowloris. To thoroughly access the performance of our multi classifier, we employed a range of evaluation metrics as shown in Fig. 5.7 In the context of multi classification, the definitions are as follows:

- True Positive (tp_i): Correctly classifying instances of class i as belonging to class i.
- False Positive (fp_i): Incorrectly classifying instances not belonging to class i as belonging to class i.
- True Negative (tn_i): Correctly classifying instances not belonging to class i as not belonging to class i.



Metric	Formula
Accuracy$_\mathrm{i}$ (Acc)	$\frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i}$
Precision$_\mathrm{i}$ (Pre)	$\frac{tp_i}{tp_i + fp_i}$
Recall$_\mathrm{i}$ (Rec)	$\frac{tp_i}{tp_i + fn_i}$
F1-Score$_\mathrm{i}$ (F1)	$\frac{2 \cdot Pre \cdot Rec}{Pre + Rec}$

Fig. 5.7 Evaluation Metrics for multi classification

- False Negative (fn$_i$): Incorrectly classifying instances belonging to class i as not belonging to class i.

Results

Here, we will present the outcomes of our Multi class classification. The figures below shows tables containing the different metrics such as precision, accuracy, recall and f1-score obtained by the multi class classification using LSTM classifier. The results are for undersampling as well as oversampling using GANs. GANs give us a significant improvement in the classification results as seen in the tables below.

Metric	Accuracy	Precision	Recall	F1-Score
Benign	0.966	0.972	0.975	0.973
DoS Hulk	0.982	0.956	0.992	0.974
DoS GoldenEye	0.986	0.898	0.108	0.193
DoS slowloris	0.997	0.914	0.759	0.830
DoS Slowhttptest	0.999	0.968	0.993	0.980

Table 5.1 Multi Classification Metrics for LSTM classifier (Undersampling)

Metric	Accuracy	Precision	Recall	F1-Score
Benign	0.999	0.999	0.999	0.999
DoS Hulk	0.999	0.999	0.998	0.998
DoS GoldenEye	0.999	0.993	0.992	0.993
DoS slowloris	0.999	0.994	0.980	0.987
DoS Slowhttptest	0.999	0.984	0.995	0.990

Table 5.2 Multi Classification Metrics for LSTM classifier (Oversampling)

Overall Accuracy for Multi classification task is 99.9%. However, oversampling provides much better results as compared to undersampling. This can be attributed to the fact that undersampling may lead to data loss. Undersampling leads to underfitting which gives poor results during the classification. GANs can generate new synthetic data for minority classes that closely resembles it. GANs preserve the information and capture the complex relationships and patterns present in the data. The data generated by GANs fill in the gaps in the feature space and hence helps the model train on a bigger and better distribution of the data and resulting in a much better performance.

5.2 IoTID20

5.2.1 Binary Classification

In the Binary Classification, our focus is on classifying the instances into two categories—benign and attack. This dataset encapsulates instances representing both benign and attack scenarios on IoT devices. The objective was to develop a binary classifier capable of distinguishing between normal behavior and the presence of a attack in a IoT device. To thoroughly assess the performance of our binary classifier, we employed a range of evaluation metrics as shown in Fig. 5.8

In the context of binary classification, we consider benign instances as false and attack as true. Therefore, the definitions are as follows:

- True Positive (tp): Correctly identifying attack as such.

Metric	Formula
Accuracy (Acc)	$\frac{tp + tn}{tp + tn + fp + fn}$
Precision (Pre)	$\frac{tp}{tp + fp}$
Recall (Rec)	$\frac{tp}{tp + fn}$
F1-Score (F1)	$\frac{2 \cdot Pre \cdot Rec}{Pre + Rec}$

Fig. 5.8 Evaluation metrics for Binary Classification

- False Positive (fp): Incorrectly identifying benign activities as an attack.
- True Negative (tn): Correctly identifying benign activities as such.
- False Negative (fn): Incorrectly identifying attack as benign.

Results

In this section, we present the outcomes of our Binary classification. The figures below shows tables containing the different metrics such as precision, accuracy, recall and f1-score and the confusion matrix obtained by the binary classification using Bayesian Neural Network (BNN) and LSTM classifier.

Metric	Value
Accuracy	0.94
Precision	0.95
Recall	0.94
F1-Score	0.92

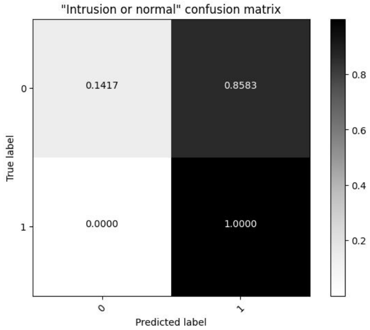


Fig. 5.9 Metrics and Confusion Matrix for BNN Classifier

Metric	Value
Accuracy	1.00
Precision	1.00
Recall	1.00
F1-Score	1.00

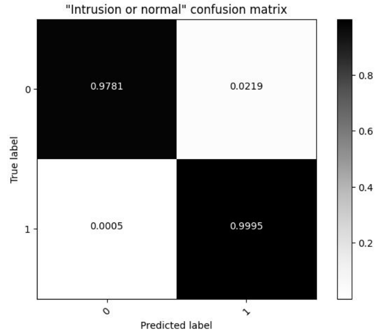


Fig. 5.10 Metrics and Confusion Matrix for LSTM Classifier

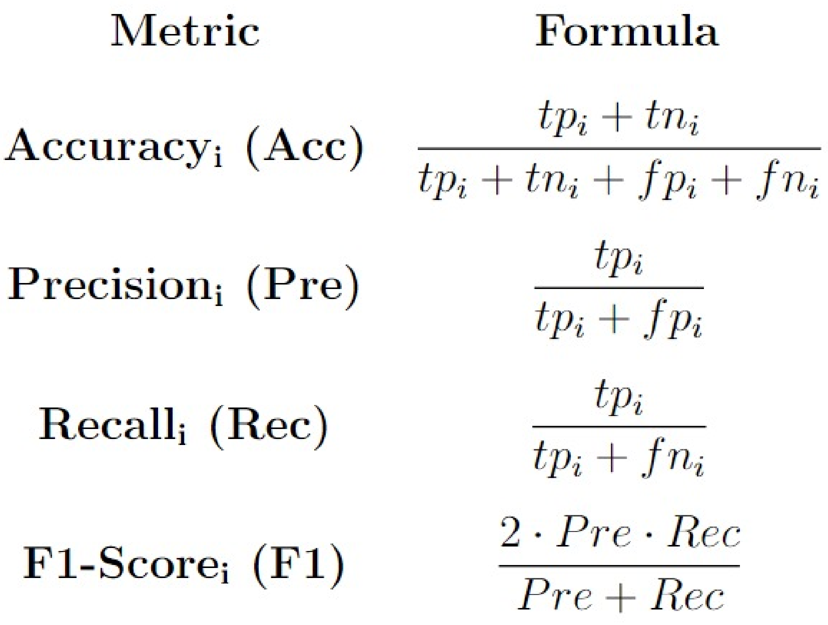
The BNN classifier has lesser accuracy as compared to the LSTM classifier due to a large number of False Positives because its primary motive is to save memory usage. The LSTM classifier took 270 KB memory while BNN took about only 40 KB.

5.2.2 Multiclass Classification

In the multiclass classification section, our objective is to categorize each instance as either one of the attack types or benign. This dataset encapsulates instances representing both benign and multiple types of attack scenarios which includes Mirai, Scan, DoS and MITM. To thoroughly access the performance of our multi classifier, we employed a range of evaluation metrics as shown in Fig. 5.11

In the context of multi classification, the definitions are as follows:

- True Positive (tp_i): Correctly classifying instances of class i as belonging to class i.
- False Positive (fp_i): Incorrectly classifying instances not belonging to class i as belonging to class i.
- True Negative (tn_i): Correctly classifying instances not belonging to class i as not belonging to class i.
- False Negative (fn_i): Incorrectly classifying instances belonging to class i as not belonging to class i.



Metric	Formula
Accuracy$_i$ (Acc)	$\frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i}$
Precision$_i$ (Pre)	$\frac{tp_i}{tp_i + fp_i}$
Recall$_i$ (Rec)	$\frac{tp_i}{tp_i + fn_i}$
F1-Score$_i$ (F1)	$\frac{2 \cdot Pre \cdot Rec}{Pre + Rec}$

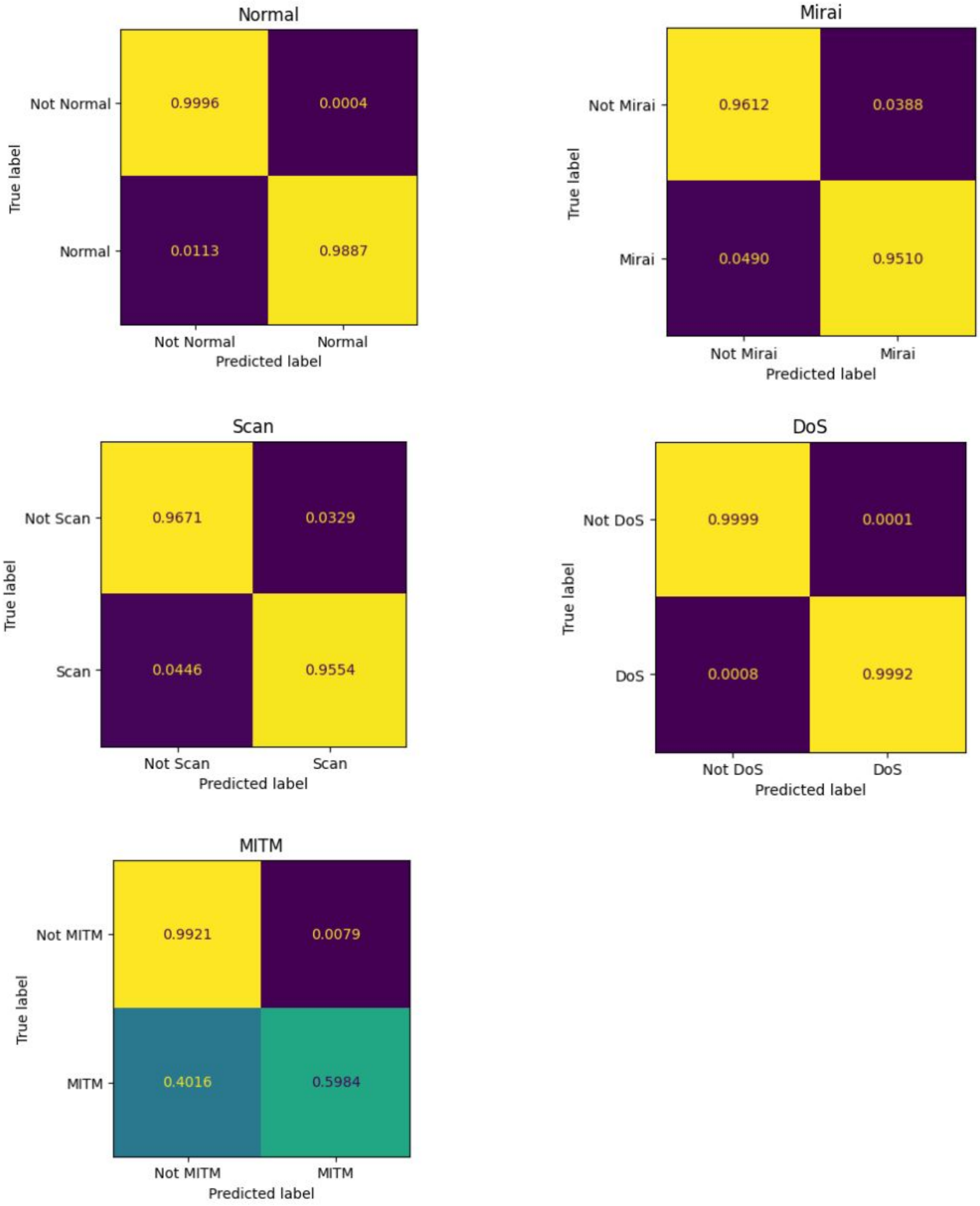
Fig. 5.11 Evaluation Metrics for multi classification

Results

In this section, we are going to present the outcomes of our Multi class classification. The figures below shows table containing the different metrics such as precision, accuracy, recall and f1-score for different classes obtained by the multi class classification using LSTM classifier and the confusion matrices for different types of attacks on IoT devices.

Metric	Accuracy	Precision	Recall	F1-Score
Benign	0.998	0.994	0.988	0.991
Mirai	0.951	0.973	0.953	0.963
Scan	0.964	0.792	0.956	0.867
DoS	0.999	0.999	0.999	0.999
MITM	0.970	0.801	0.639	0.711

Table 5.3 Multiclass Classification Metrics for LSTM classifier



Normal

Mirai

Scan

DoS

MITM

Fig. 5.12 Confusion Matrices of Different Classes for LSTM Classifier

Chapter 6

Cross Dataset Evaluation of LSTM Classifier

The approach of cross-dataset evaluation aims to enhance the effectiveness of evaluating LSTM-based Intrusion Detection Systems (IDS). This involves utilizing two separate datasets originating from different computer network environments. The rationale behind this strategy is to train machine learning models on one of the datasets and then test them on the other one. Subsequently, this process is reversed, with the testing dataset used for training and vice versa. This technique allows for the evaluation of both the quality of the datasets utilized and the efficacy of the detection models.

The performance of the LSTM classifier is evaluated. This approach provides insights into how our LSTM classifier adapts to the changes in network configurations, enabling us to analyze the model's response to variations in network architecture. High performance indicates a richer dataset with more intrusion data. It's essential to consider only shared features between the two datasets during cross-dataset evaluation to maintain model consistency when tested on different datasets.

In our work, we employ the following two datasets: CICIDS-2017 and IoTID20, sourced from different networks. We are conducting cross-model evaluation both ways i.e. by training the classifier on CICIDS-2017 and testing it on IoTID20 dataset and vice versa. We are going to conduct binary classification in the cross-model evaluation.

The architecture of the LSTM model we employed for the cross-dataset evaluation incorporates two LSTM layers, each comprising 64 hidden neurons. Rectified Linear Unit (ReLU) functions serve as activation functions for the LSTM layers, while the final dense layer utilizes the softmax function for class probability computation during classification. To enhance generalization, a dropout layer with a 0.2 rate is introduced. Model optimization is carried out through the ADAM optimizer with a learning rate of 0.003 used along with the binary cross-entropy loss function.

6.1 Results

When we trained the LSTM classifier on the IoTID20 dataset and tested it on the CICIDS-2017 dataset, we got an overall accuracy of 56.17%. The low accuracy can be attributed to the fact that as the network and IoT devices have some common attacks with a lot of attacks present in only one of the datasets making it difficult to classify such attacks and a lot of false positives as IoTID20 has a class imbalance with most of the instances being of attack. However, for classifying attacks, we get a 98.91% accuracy as compared to the 98.1% accuracy obtained from ANN.

When we trained the LSTM classifier on the CICIDS-2017 dataset and tested it on the IoTID20 dataset, we got an overall accuracy of 43.84%. The low accuracy can be attributed to the fact that as the network and IoT devices have some common attacks with a lot of attacks present in only one of the datasets making it difficult to classify such attacks and a lot of false negatives as CICIDS-2017 has a lot of benign instances while most of the instances in IoTID20 are of attack due to a large class imbalance.

Chapter 7

LSTM V/S ANN

In this section, we are going to compare the results obtained by the LSTM classifier on the IoTID20 dataset with the results obtained by classifying the IoTID20 data by an ANN classifier. We are going to compare the effectiveness of our Intrusion Detection System (IDS) to a IDS based on ANN model. We will compare both the binary classification as well as multiclass classification performance of both the classifiers.

7.1 Binary Classification

In this section, we are going to compare the results of binary classification by the two classifiers. For Binary classification, the IDS is going to classify the instances as either ”benign” if the IoT device is safe or ”attack” if the IoT device is under attack. The figures 7.1 and 7.2 depict the various metrics like accuracy, precision, recall and the F1-score while the figures 7.3 and 7.4 contains the confusion matrices obtained by the binary classification using the two classifiers LSTM and ANN.

Metric	Value
Accuracy	1.00
Precision	1.00
Recall	1.00
F1-Score	1.00

Fig. 7.1 LSTM Metrics

Metric	Value
Accuracy	0.98
Precision	0.98
Recall	0.98
F1-Score	0.98

Fig. 7.2 ANN Metrics

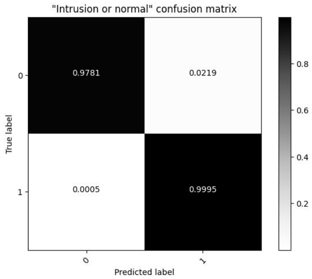


Fig. 7.3 LSTM Confusion Matrix

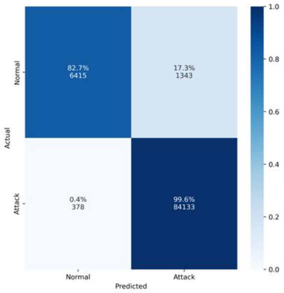


Fig. 7.4 ANN Confusion Matrix

7.2 Multiclass Classification

In the multiclass classification section, our objective is to compare the results of binary classification by the two classifiers. For Multiclass classification, the IDS is going to classify the instances as either ”benign” if the IoT device is safe or as one of the attack types if it feels that the IoT device is under attack. This dataset encapsulates instances representing both benign and multiple types of attack scenarios which includes Mirai, Scan, DoS etc. The tables 7.1 and 7.2 depict the various metrics like accuracy, precision, recall and the F1-score for different classes obtained by the multi class classification by the two classifiers LSTM and ANN.

Metric	Accuracy	Precision	Recall	F1-Score
Benign	0.998	0.994	0.988	0.991
Mirai	0.951	0.973	0.953	0.963
Scan	0.964	0.792	0.956	0.867
DoS	0.999	0.999	0.999	0.999

Table 7.1 Multiclass Classification Metrics for LSTM classifier

Metric	Accuracy	Precision	Recall	F1-Score
Benign	0.850	0.901	0.849	0.874
Mirai	0.869	0.910	0.869	0.889
Scan	0.817	0.495	0.817	0.616
DoS	0.739	0.995	0.739	0.848

Table 7.2 Multiclass Classification Metrics for ANN classifier

7.3 Conclusion

Based on the results depicted in the figures and tables above, we can conclude that the IDS based on LSTM performs much better as compared to the one based on ANN model. The LSTM classifier performs much better in case of finding whether there is an attack on an IoT device or not and to find the exact type of attack on the IoT device which can help us tackle it and hence improve the security of the IoT devices as they are of wide importance to us.

Chapter 8

Performance Analysis of LSTM Model

8.1 Exponential Weighted Average (EWA)

In the Exponential Weighted Average (EWA) section, a novel approach was adopted to enhance the input data preprocessing for the LSTM model on IoTIDS dataset. Rather than directly passing the time sequence of data input to the LSTM layer, an exponential weighted average with a parameter alpha was computed. This process involved assigning exponentially decreasing weights to the input data, with greater emphasis placed on more recent observations. By leveraging EWA, the model could effectively capture the temporal dynamics of the data while minimizing the influence of noise and irrelevant information. The computed EWA values were then utilized as inputs to the LSTM layers, enabling the model to make more informed predictions based on the weighted history of the input data. This innovative preprocessing technique aimed to improve the LSTM model's performance and robustness in handling temporal data sequences within the cybersecurity context.

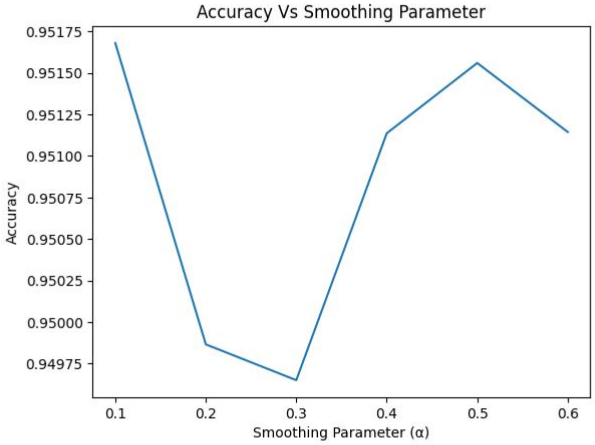


Fig. 8.1 Binary Classification EWA

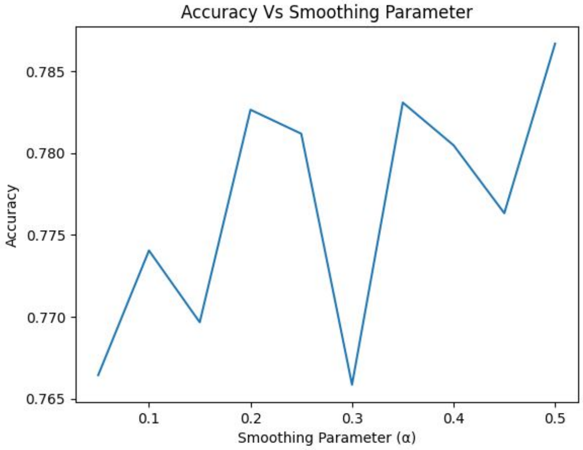


Fig. 8.2 Multiclass Classification EWA

8.2 Analysis of Hyperparameters in LSTM

The Hyperparameter window length affects the accuracy of the prediction by the LSTM classifier as well as the time needed by the LSTM classifier to give the prediction.

A longer window length allows the LSTM model to capture more temporal dependencies and patterns in the data. This genrally leads to better accuracy, especially when dealing with long-term dependencies in time series data. However, this can lead to longer training and inference times as it requires processing more data per time step. In contrast, A shorter window length can help reduce the complexity of the model and mitigate overfitting, especially in cases where the data has short-term dependencies or is noisy. Shorter window length also leads to faster training and inference times. This can improve the scalability of the model and make it more suitable for real-time applications especially where latency is crucial. Additionally, longer window length may introduce vanishing or exploding gradient problems during training, affecting the model's ability to learn.

Longer window lengths may improve accuracy by capturing more temporal dependencies but can lead to longer execution times and increased risk of overfitting. Shorter window lengths may reduce execution time and mitigate overfitting but may sacrifice accuracy, especially in tasks requiring long-term context. Hence, it is essential to experiment with different window lengths and evaluate their impact on both accuracy and execution time to find the optimal balance.

8.2.1 Binary Classification

In this section, we are going to explore how the window length affects the accuracy and the execution time in the case of binary classification by the LSTM classifier on the IoTID20 dataset. The figures below are obtained by varying window lengths and obtaining the accuracy and execution time values.

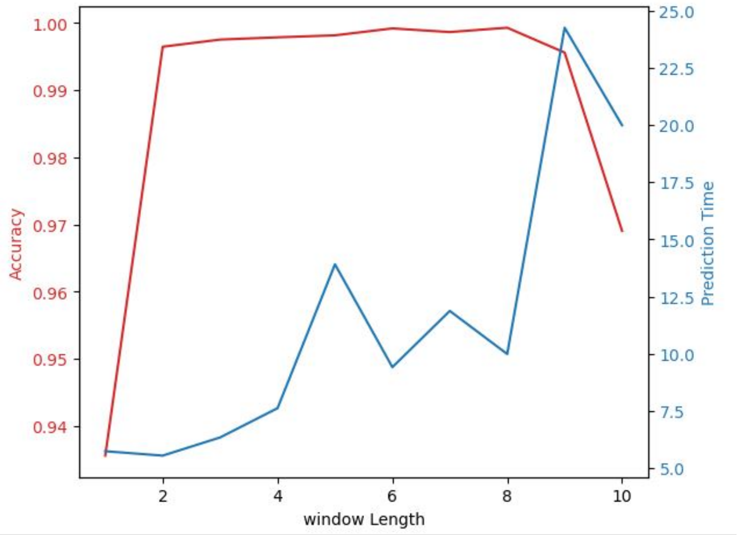
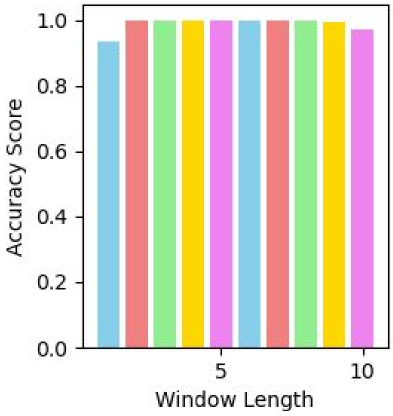
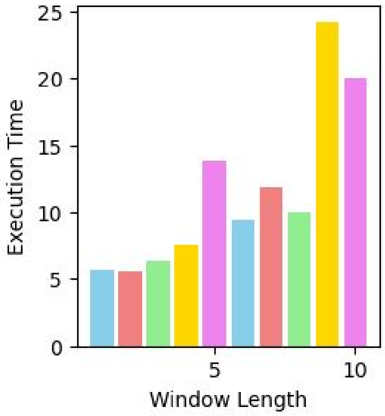


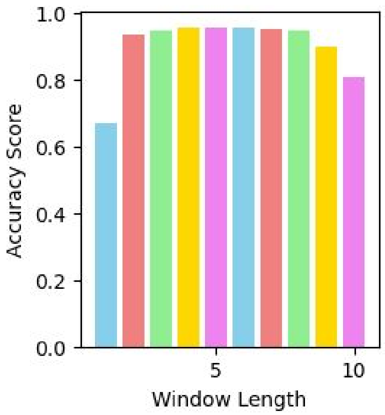
Fig. 8.3 Accuracy & Execution time V/S window size (binary classification)

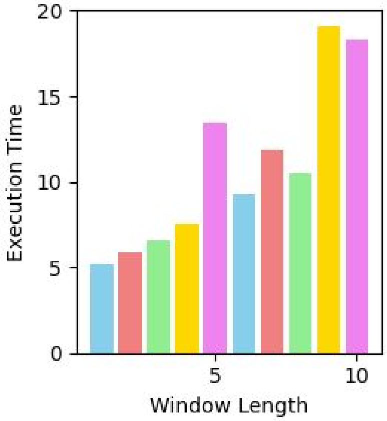




8.2.2 Multiclass classification

In this section, we are going to explore how the window length affects the accuracy and the execution time in the case of multi class classification by the LSTM classifier on the IoTID20 dataset. The figures below are obtained by varying window lengths and obtaining the accuracy and execution time values.





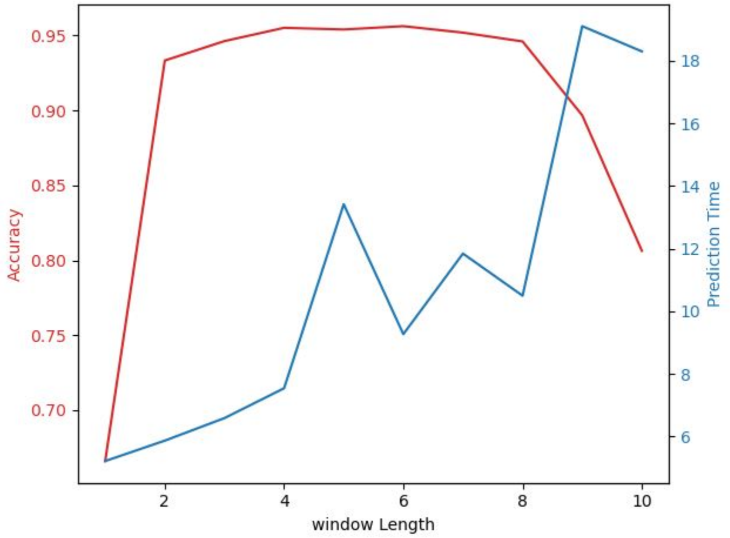


Fig. 8.4 Accuracy & Execution time V/S window size (multi classification)

Chapter 9

LSTM Model Explanation

There are a lot of methods that are used to explain the predictions made by a model. Technically, these methods rest on an approximation of the classification function, which enables them to estimate how the input instances contribute to a prediction. The explanation model utilizes the relevance between the input instances and the prediction of the classification model to identify important features and highlight the top features that are contributing the predicted accuracy by the classification model.

In this chapter, our emphasis lies on leveraging two pivotal techniques, LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations), to interpret the predictions generated by LSTM models on the CICIDS and IoTIDS datasets. Through these sophisticated interpretability methods, we aim to shed light on the decision-making processes of LSTM models in the context of intrusion detection within network traffic and IoT data.

9.1 LIME

LIME (Local Interpretable Model-agnostic Explanations) is a technique used for explaining the predictions of machine learning models. It provides local interpretability by approximating the predictions of a black-box model in the vicinity of a specific instance. LIME works by generating perturbed samples around the instance of interest and fitting a simpler, interpretable model (e.g., linear regression) to explain the predictions of the complex model within that local region.

In this section, we employ the LIME explanation method to explain the predictions of LSTM model for both binary and multiclass classification for both the datasets. The LIME explainer gives the contribution of all input features in explaining the output and the features with more weights contribute more in predicting the output.

9.1.1 IoTID20

For prediction using binary LSTM classifier used for IoTIDS data, a attack sample is taken and predicted it which turned out to be “attack”. In Figure 9.1, we present a comprehensive list of the most influential input features contributing to this prediction, along with their corresponding weights.

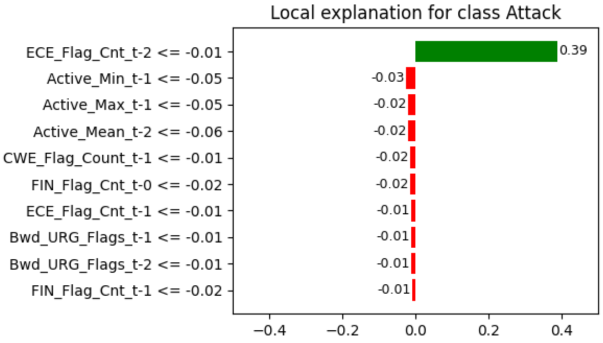
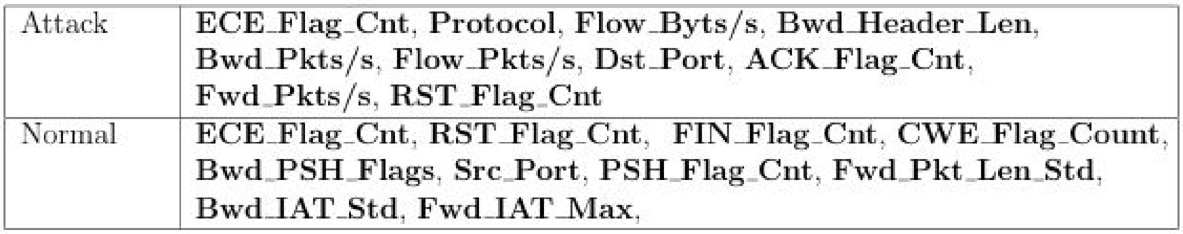


Fig. 9.1 LIME Features Weights for Attack sample

Table 9.1 Top 10 Features for 2 different Instances of IoTIDS Data



Attack	**ECE_Flag_Cnt, Protocol, Flow_Byts/s, Bwd_Header_Len, Bwd_Pkts/s, Flow_Pkts/s, Dst_Port, ACK_Flag_Cnt, Fwd_Pkts/s, RST_Flag_Cnt**
Normal	**ECE_Flag_Cnt, RST_Flag_Cnt, FIN_Flag_Cnt, CWE_Flag_Count, Bwd_PSH_Flags, Src_Port, PSH_Flag_Cnt, Fwd_Pkt_Len_Std, Bwd_IAT_Std, Fwd_IAT_Max,**

Subsequently, to gain further insights into the model's behavior and feature importance across normal and attack instances, we leveraged the LIME (Local Interpretable Model-agnostic Explanations) technique. This process involved analyzing the top 10 features contributing to predictions across the both instances as shown in Table 9.1. Given that the LSTM model operates based on temporal sequences, where features are often associated with timestamps, we essential developed a method that effectively captured the significance of these time-dependent features.

For prediction using multiclass LSTM classifier used for IoTIDS data, a mirai sample is taken and predicted it which turned out to be "Mirai". In Figure 9.2, we present a comprehensive list of the most influential input features contributing to this prediction, along with their corresponding weights. This process involved analyzing the top 10 features contributing to predictions across the normal, mirai, scan, dos, and mitm arp spoofing instances as shown in Table 9.2.

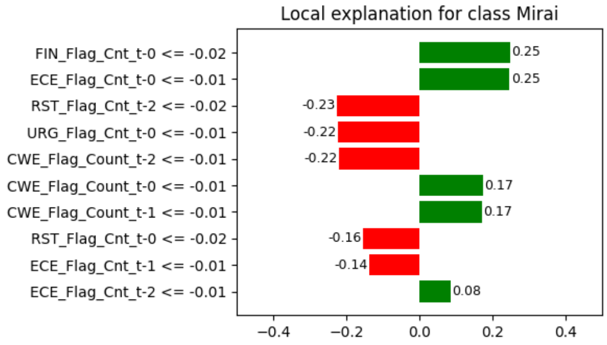


Fig. 9.2 LIME Features Weights for Mirai sample

9.1.2 CICIDS-2017

For prediction using binary LSTM classifier used for CICIDS data, a DoS Attack sample is taken and predicted it which turned out to be "DoS Attack". In Figure 9.3, we present a comprehensive list of the most influential input features contributing to this prediction, along with their corresponding weights. This process involved analyzing the top 10 features contributing to predictions across the benign and DDoS instances as shown in Table 9.3.

Table 9.2 Top 10 Features for 5 different sub-classes of IoTIDS Data

Mirai	**RST_Flag_Cnt, FIN_Flag_Cnt, CWE_Flag_Count, Active_Std, Active_Min, ECE_Flag_Cnt, Fwd_Pkts/s, Flow_Pkts/s, Bwd_Pkts/s, Protocol**
Scan	**RST_Flag_Cnt, FIN_Flag_Cnt, CWE_Flag_Count, Active_Std, Active_Min, ECE_Flag_Cnt, Bwd_URG_Flags, Protocol, Active_Max, Active_Mean**
DoS	**RST_Flag_Cnt, FIN_Flag_Cnt, CWE_Flag_Count, Flow_IAT_Mean, Flow_Duration, Flow_IAT_Min, Idle_Min, Idle_Mean, Flow_IAT_Max, Bwd_IAT_Mean**
MITM ARP Spoofing	**RST_Flag_Cnt, FIN_Flag_Cnt, CWE_Flag_Count, Active_Std, ECE_Flag_Cnt, Active_Min, Protocol, URG_Flag_Cnt, Bwd_URG_Flags, Active_Max**
Normal	**RST_Flag_Cnt, FIN_Flag_Cnt, CWE_Flag_Count, Active_Std, Active_Min, ECE_Flag_Cnt, Dst_Port, Protocol, URG_Flag_Cnt, Active_Max**

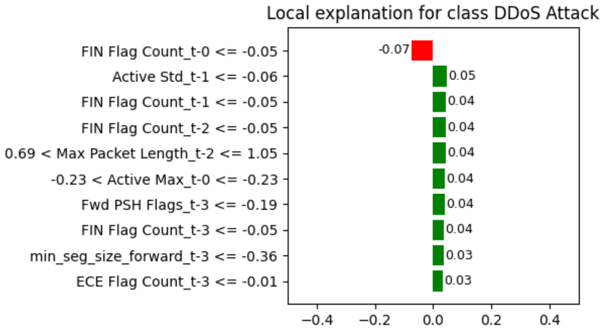


Fig. 9.3 LIME Features Weights for DDoS sample

Table 9.3 Top 10 Features for 2 different Instances of CICIDS Data

DDoS Attack	**FIN Flag Count, Fwd PSH Flags, SYN Flag Count, Active Std, min_seg_size_forward, URG Flag Count, RST Flag Count, ECE Flag Count, Idle Std, ACK Flag Count**
Benign	**FIN Flag Count, Active Std, Fwd PSH Flags, SYN Flag Count, min_seg_size_forward, URG Flag Count, ECE Flag Count, RST Flag Count, Idle Std, ACK Flag Count,**

For prediction using multiclass LSTM classifier, a DoS Hulk sample is taken and predicted which turned out to be “DoS Hulk”. In Figure 9.4, we present a comprehensive list of the most influential input features contributing to this prediction, along with their corresponding weights. This process involved analyzing the top 10 features contributing to predictions across the normal, mirai, scan, dos, and mitm arp spoofing instances as shown in Table 9.4.

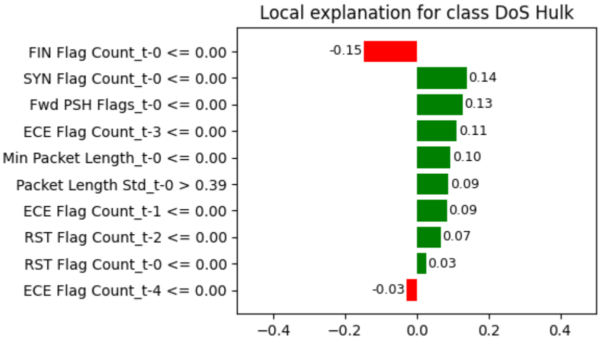
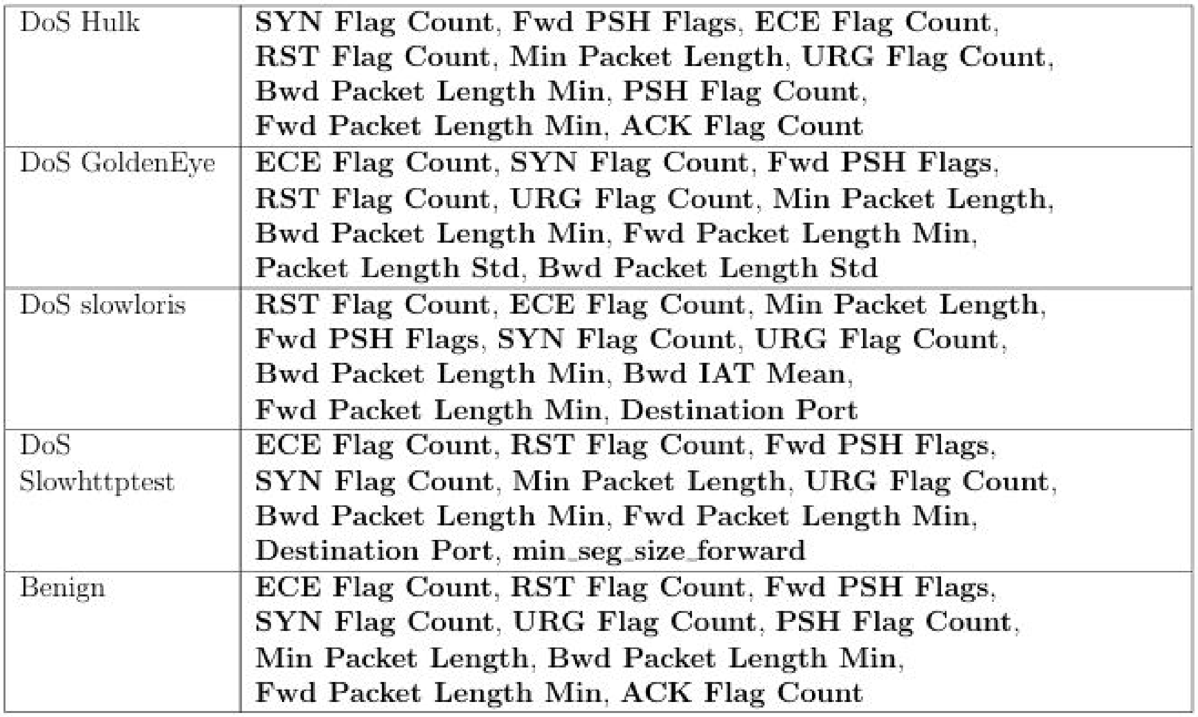


Fig. 9.4 LIME Features Weights for DDoS Hulk sample

Table 9.4 Top 10 Features for 5 different sub-classes of CICIDS Data



DoS Hulk	**SYN Flag Count, Fwd PSH Flags, ECE Flag Count, RST Flag Count, Min Packet Length, URG Flag Count, Bwd Packet Length Min, PSH Flag Count, Fwd Packet Length Min, ACK Flag Count**
DoS GoldenEye	**ECE Flag Count, SYN Flag Count, Fwd PSH Flags, RST Flag Count, URG Flag Count, Min Packet Length, Bwd Packet Length Min, Fwd Packet Length Min, Packet Length Std, Bwd Packet Length Std**
DoS slowloris	**RST Flag Count, ECE Flag Count, Min Packet Length, Fwd PSH Flags, SYN Flag Count, URG Flag Count, Bwd Packet Length Min, Bwd IAT Mean, Fwd Packet Length Min, Destination Port**
DoS Slowhttptest	**ECE Flag Count, RST Flag Count, Fwd PSH Flags, SYN Flag Count, Min Packet Length, URG Flag Count, Bwd Packet Length Min, Fwd Packet Length Min, Destination Port, min_seg_size_forward**
Benign	**ECE Flag Count, RST Flag Count, Fwd PSH Flags, SYN Flag Count, URG Flag Count, PSH Flag Count, Min Packet Length, Bwd Packet Length Min, Fwd Packet Length Min, ACK Flag Count**

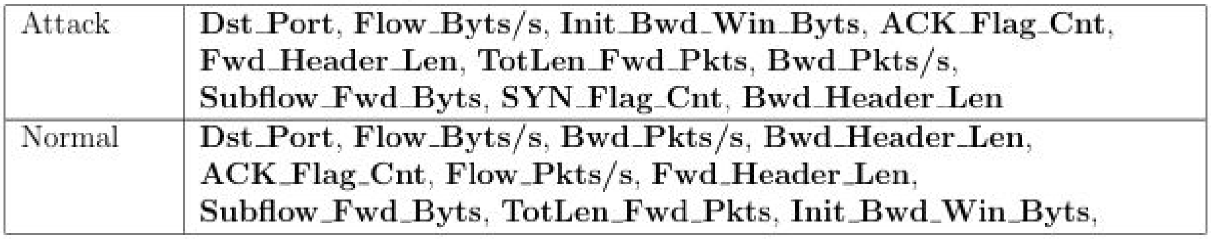
9.2 SHAP

In this section, we employ the SHAP explanation method to explain the predictions of LSTM model for both binary and multiclass classification for both the datasets. The SHAP explainer gives the contribution of all input features in explaining the output and the features with more weights contribute more in predicting the output.

9.2.1 IoTID20

For prediction using binary LSTM classifier, a attack sample is taken and predicted it which turned out to be "attack". In Figure 9.5, we present a comprehensive list of the most influential input features contributing to this prediction, along with their corresponding weights. This process involved analyzing the top 10 features contributing to predictions across the normal and attack instances as shown in Table 9.5.

Table 9.5 Top 10 Features for 2 different Instances of IoTIDS Data



Attack	**Dst_Port, Flow_Byts/s, Init_Bwd_Win_Byts, ACK_Flag_Cnt, Fwd_Header_Len, TotLen_Fwd_Pkts, Bwd_Pkts/s, Subflow_Fwd_Byts, SYN_Flag_Cnt, Bwd_Header_Len**
Normal	**Dst_Port, Flow_Byts/s, Bwd_Pkts/s, Bwd_Header_Len, ACK_Flag_Cnt, Flow_Pkts/s, Fwd_Header_Len, Subflow_Fwd_Byts, TotLen_Fwd_Pkts, Init_Bwd_Win_Byts,**

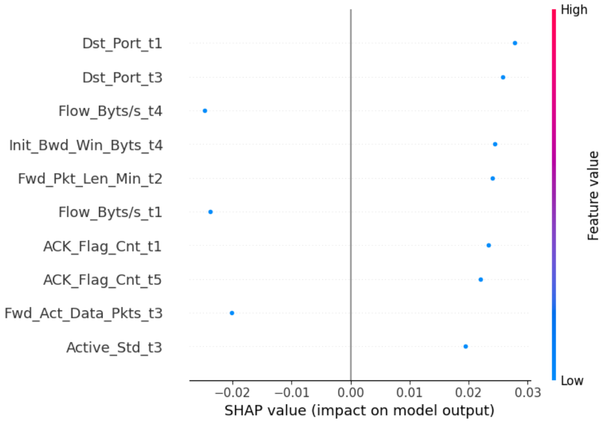


Fig. 9.5 SHAP Features Weights for attack sample

For prediction using multiclass LSTM classifier, a mirai sample is taken and predicted it which turned out to be “Mirai”. In Figure 9.6, we present a comprehensive list of the most influential input features contributing to this prediction, along with their corresponding weights. This process involved analyzing the top 10 features contributing to predictions across the normal, mirai, scan, dos, and mitm arp spoofing instances as shown in Table 9.6.

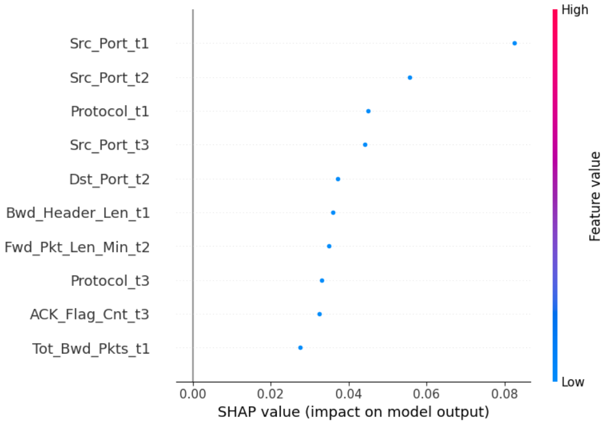
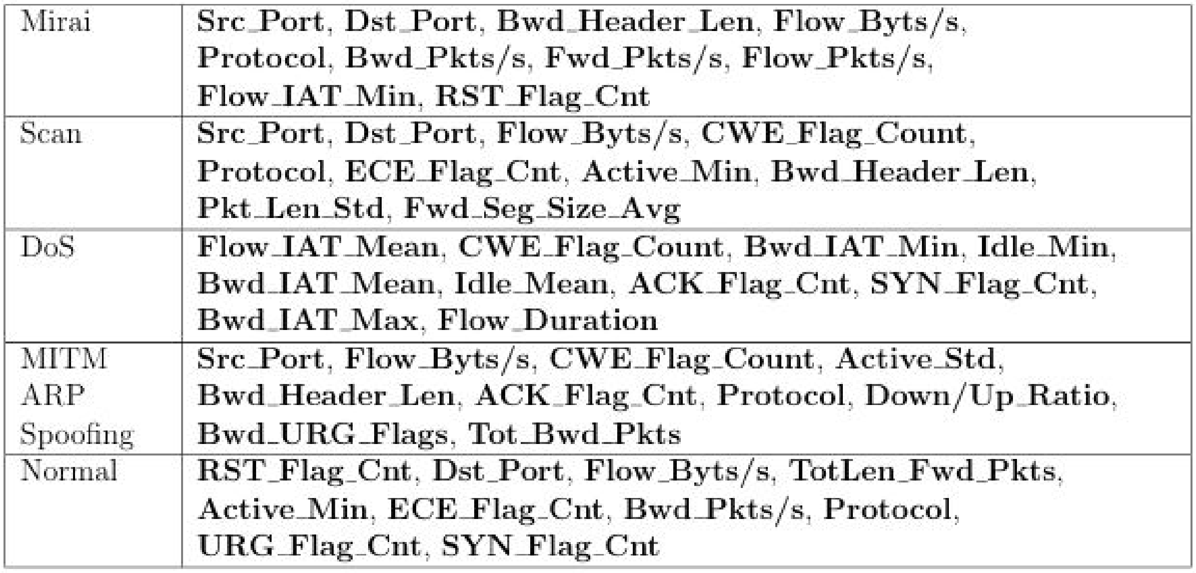


Fig. 9.6 SHAP Features Weights for Mirai sample

Table 9.6 Top 10 Features for 5 different sub-classes of IoTIDS Data



Mirai	**Src_Port, Dst_Port, Bwd_Header_Len, Flow_Byts/s, Protocol, Bwd_Pkts/s, Fwd_Pkts/s, Flow_Pkts/s, Flow_IAT_Min, RST_Flag_Cnt**
Scan	**Src_Port, Dst_Port, Flow_Byts/s, CWE_Flag_Count, Protocol, ECE_Flag_Cnt, Active_Min, Bwd_Header_Len, Pkt_Len_Std, Fwd_Seg_Size_Avg**
DoS	**Flow_IAT_Mean, CWE_Flag_Count, Bwd_IAT_Min, Idle_Min, Bwd_IAT_Mean, Idle_Mean, ACK_Flag_Cnt, SYN_Flag_Cnt, Bwd_IAT_Max, Flow_Duration**
MITM ARP Spoofing	**Src_Port, Flow_Byts/s, CWE_Flag_Count, Active_Std, Bwd_Header_Len, ACK_Flag_Cnt, Protocol, Down/Up_Ratio, Bwd_URG_Flags, Tot_Bwd_Pkts**
Normal	**RST_Flag_Cnt, Dst_Port, Flow_Byts/s, TotLen_Fwd_Pkts, Active_Min, ECE_Flag_Cnt, Bwd_Pkts/s, Protocol, URG_Flag_Cnt, SYN_Flag_Cnt**

9.2.2 CICIDS-2017

For prediction using binary LSTM classifier, a benign sample is taken and predicted it which turned out to be "Benign". In Figure 9.7, we present a comprehensive list of the most influential input features contributing to this prediction, along with their corresponding weights. This process involved analyzing the top 10 features contributing to predictions across the DDoS Attack and Benign instances as shown in Table 9.7.

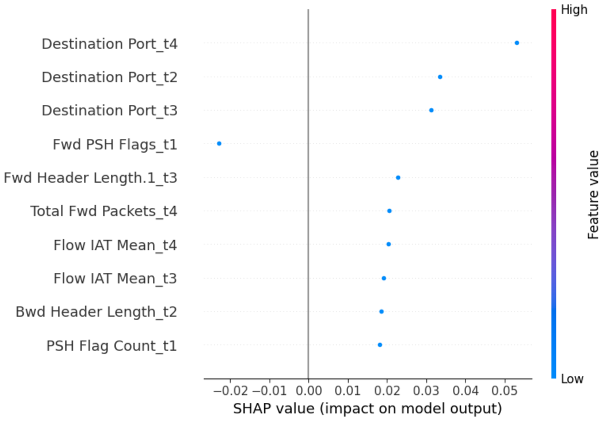
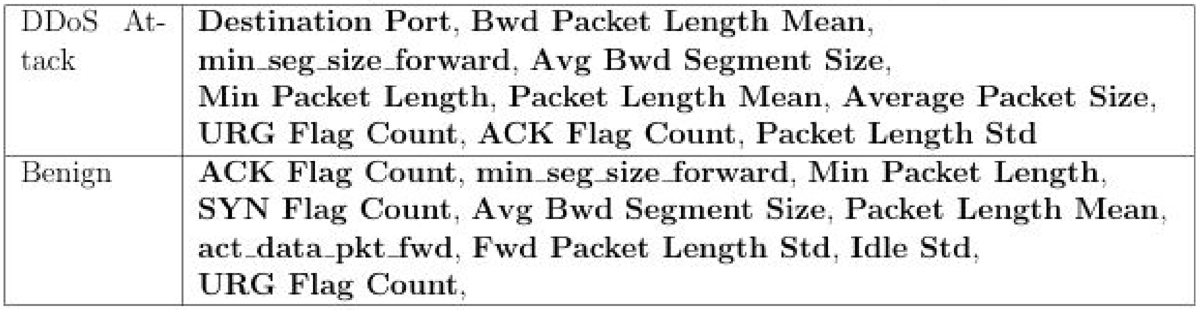


Fig. 9.7 SHAP Features Weights for Benign sample

Table 9.7 Top 10 Features for 2 different classes of CICIDS Data



DDoS Attack	**Destination Port, Bwd Packet Length Mean, min_seg_size_forward, Avg Bwd Segment Size, Min Packet Length, Packet Length Mean, Average Packet Size, URG Flag Count, ACK Flag Count, Packet Length Std**
Benign	**ACK Flag Count, min_seg_size_forward, Min Packet Length, SYN Flag Count, Avg Bwd Segment Size, Packet Length Mean, act_data_pkt_fwd, Fwd Packet Length Std, Idle Std, URG Flag Count,**

For prediction using multiclass LSTM classifier, a DDoS Hulk sample was taken and predicted which turned out to be "DDoS Hulk". Figure 9.8 present a comprehensive list of the most influential input features contributing to this prediction, along with their corresponding weights. This process involved analyzing the top 10 features contributing to predictions across the DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest and Benign instances as shown in Table 9.8.

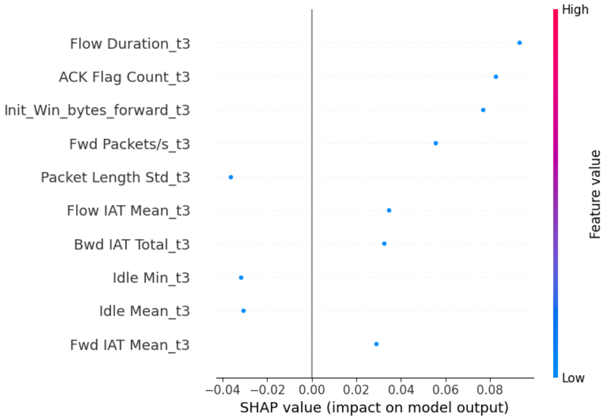
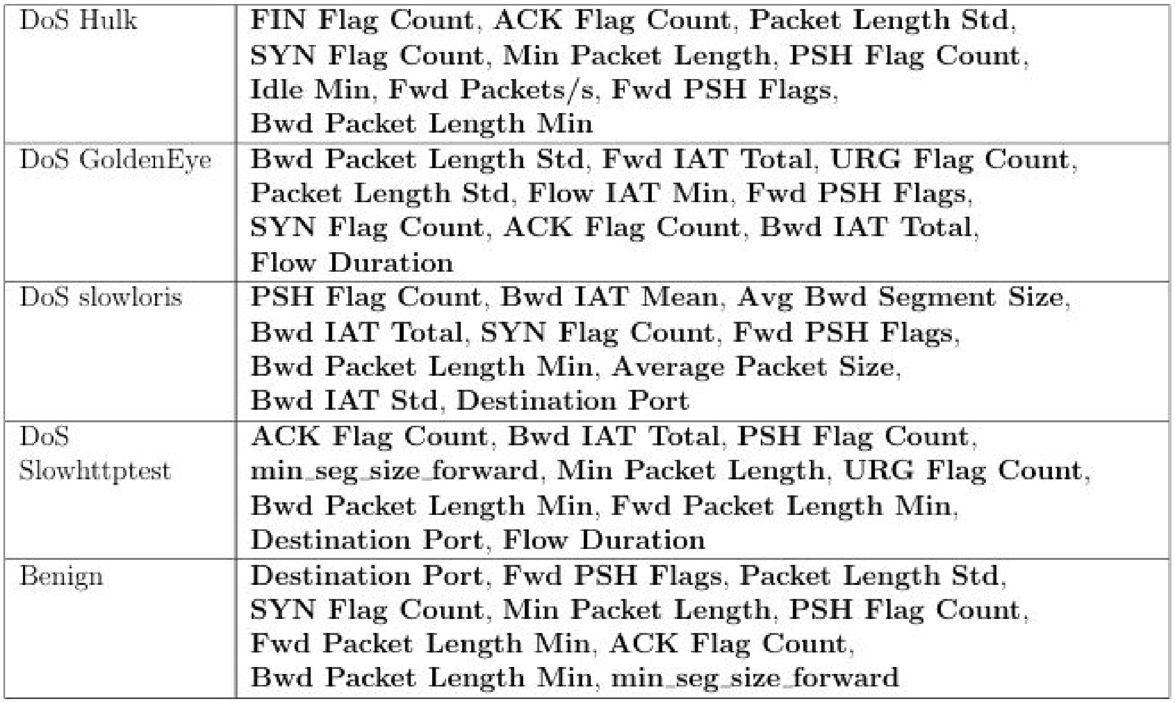


Fig. 9.8 SHAP Features Weights for DDoS Hulk Sample

Table 9.8 Top 10 Features for 5 different sub-classes of CICIDS Data



DoS Hulk	**FIN Flag Count, ACK Flag Count, Packet Length Std, SYN Flag Count, Min Packet Length, PSH Flag Count, Idle Min, Fwd Packets/s, Fwd PSH Flags, Bwd Packet Length Min**
DoS GoldenEye	**Bwd Packet Length Std, Fwd IAT Total, URG Flag Count, Packet Length Std, Flow IAT Min, Fwd PSH Flags, SYN Flag Count, ACK Flag Count, Bwd IAT Total, Flow Duration**
DoS slowloris	**PSH Flag Count, Bwd IAT Mean, Avg Bwd Segment Size, Bwd IAT Total, SYN Flag Count, Fwd PSH Flags, Bwd Packet Length Min, Average Packet Size, Bwd IAT Std, Destination Port**
DoS Slowhttptest	**ACK Flag Count, Bwd IAT Total, PSH Flag Count, min_seg_size_forward, Min Packet Length, URG Flag Count, Bwd Packet Length Min, Fwd Packet Length Min, Destination Port, Flow Duration**
Benign	**Destination Port, Fwd PSH Flags, Packet Length Std, SYN Flag Count, Min Packet Length, PSH Flag Count, Fwd Packet Length Min, ACK Flag Count, Bwd Packet Length Min, min_seg_size_forward**

Chapter 10

Conclusion and Future Work

10.1 Conclusion

Based on the observations , we conclude that our binary classification results demonstrate a high level of accuracy, with nearly 100% of the samples correctly classified. our multi classification results demonstrate a overall accuracy close to 99.9% after using GANs for oversampling as compared to only 97% in case of an undersampling for the CICIDS-2017 dataset while for the IoTID20 dataset, nearly 100% accuracy is reflected by LSTM for binary classification and an accuracy of more than 96% in the case of multiclass classification. The data imbalance in CICIDS-2017 for multiclass classification is addressed by genrating synthetic data samples of minority classes by using GANs resulting in an increase in the accuracy. The cross dataset evaluation gave us an idea of how our model will act when it is shifted from one environment to the other. Our LSTM based IDS performed better as compared to ANN based IDS. We can optimise our classifier for accuracy and execution time by varying the window length. We got an overview of the features playing a major role in the prediction results for different classes with the help of explanators such as LIME and SHAP.

10.2 Future Work

Our primary goal was to create an Intrusion Detection System (IDS). Our IDS should be able to detect all existing attacks and any new attacks that may come in the near future with minimal chances of false positives and false negatives.

When we performed the cross dataset evaluation, it was not excellent. Hence, we should try to create an IDS that performs well in all kind of scenarios whether to detect a network attack or IoT attack or any other attacks.

We should make our classifier faster, more scalable, reliable, efficient and ready to be applicable in the diverse real life scenarios. We should try to the make the intrusion detection as fast as possible so as to minimise the damage caused and as latency plays a significant role in various scenarios.

The IDS should be frequently updated with the latest data so as to be able to detect the new attacks. We should be able to find a way so that the IDS is able to cope up with the new attack techniques and patterns. Hence, we should try to devise an algorithm with the help of which the model can calibrate itself and be ready for all kinds of attacks.

The use of ensemble methods with LSTM classifiers can be explored to improve robustness and generalization performance. Ensemble techniques such as bagging, boosting, or stacking could be applied to combine multiple LSTM models trained on different subsets of the data or with different initializations.

Adaptive learning algorithms may be developed that can dynamically adjust the learning rate or sample weights of the classifier during the training phase to focus more on minority class samples or to mitigate the impact of class imbalances on model training.

References

[CHLL20] Yini Chen, Jun Hou, Qianmu Li, and Huaqiu Long. Ddos attack detection based on random forest. In *2020 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pages 328–334. IEEE, 2020.

[DL23] Sarah Johnson David Lee. Anomaly detection in iot networks using unsupervised learning: Insights from iotid20 datasett. In *IEEE Internet of Things Journal, Volume 8, Issue 4*. IEEE, 2023.

[EJ24] Michael Brown Emily Johnson. Feature selection and ensemble learning for intrusion detection in iot networks: A study on iotid20 dataset. In *ACM Transactions on Internet Technology, Volume 15, Issue 2*. ACM, 2024.

[HKK19] Gyung Min Lee Jeong Do Yoo Kyung Ho Park Hyunjae Kang, Dong Hyun Ahn and Huy Kang Kim. Iot network intrusion dataset. In *IEEE Dataport*. IEEE, 2019.

[JD23] Jane Smith John Doe. Deep learning-based intrusion detection system for iot networks using iotid20 dataset. In *IEEE Transactions on Network and Service Management, Volume 10, Issue 3*. IEEE, 2023.

[SG20] Swathi Sambangi and Lakshmeeswari Gondi. A machine learning approach for ddos (distributed denial of service) attack detection using multiple linear regression. In *Proceedings*, volume 63, page 51. MDPI, 2020.

[UNK21] G Usha, Mohak Narang, and Akash Kumar. Detection and classification of distributed dos attacks using machine learning. In *Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020*, pages 985–1000. Springer, 2021.