National College of Ireland

IDS for IoT to detect DDoS attacks using BiLSTM and cGAN

MSc Research Project
Cybersecurity

Hrishikesh Krishnan
Student ID: 22192727

School of Computing
National College of Ireland

Supervisor: Dr. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	HRISHIKESH KRISHNAN
Student ID:	22192727
Programme:	MSC CYBERSECURITY **Year:** 2024
Module:	PRACTICUM
Supervisor:	DR. IMRAN KHAN
Submission Due Date:	16/09/2024
Project Title:	IDS FOR IOT TO DETECT DDOS ATTACKS USING BiLSTM AND CGAN
Word Count: 5877	**Page Count:** 18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: HRISHIKESH KRISHNAN

Date: 16/09/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

IDS for IoT to detect DDoS attacks using BiLSTM and cGAN

Hrishikesh Krishnan
22192727

Abstract

The IoT devices have transformed the industries and the lives of everyone significantly and the IoT technology has been widely accepted by many homes to make their life easier. This huge rise in the use of IoT devices for everyday tasks and the architecture of these devices made them an attractable target for the attackers. The small architecture and limited computational capacity make them fall for DDoS attacks easily. This invites the need for an intrusion detection system to detect DDoS attacks. This work aims to develop a Bidirectional LSTM model integrated with Conditional GAN for data augmentation to detect DDoS attacks in IoT devices. The model is then evaluated, and it showed excellent results including accuracy 88.6%, precision 88.65%, recall 88.6% and f1 score 88.4%.

Keywords: IDS, IoT, DDoS, BiLSTM, cGAN

1 Introduction

The influence of IoT devices in the transformation of industries and everyday life are very high. This includes enabling smarter cities, improving consumer experiences and these has been improving significantly by the involvement of IoT devices. According to recent reports, the number of IoT devices is expected to reach 75 billion by 2025 (Statista, 2016). This huge outburst of IoT devices has made IoT networks an attractive target for cyber-attacks. The most common attack on IoT networks is Distributed Denial of Service (DDoS) attack. In October 2016, the Mirai botnet launched a massive DDoS attack by exploiting IoT devices. This attack temporarily shut down major websites like Twitter, Reddit and Netflix (Malwarebytes, 2018).

IoT devices typically has limited computational resources, and this restricted processing capability, memory and storage capacity makes it difficult to include strong security features. Advanced encryption and real time threat detection require much more computational resources than that the IoT device has. This is the major limitation of IoT devices, and this is the major reason that IoT devices becomes the victim of cyber-attacks. The lack of standardization in the IoT devices, where different manufactures implement different security standards, and this makes it difficult to follow consistent security practices for all devices and this inconsistency leaves many devices vulnerable. Often in most cases the IoT devices does not receive much support in terms of firmware updates, and these could create a huge gap, and it can be exploited easily. The Mirai botnet attack exploited this firmware vulnerability. As these IoT devices create a network, any of these devices become vulnerable and if exploited, it compromises the entire network, and the consequences can be huge.

Distributed Denial of Service (DDoS) attacks aim to disrupt the normal functioning of the target. The target can be a server, network or a service and it overwhelms the target with traffic (Cloudflare, no date). This attack can be executed using a collection of combined compromised devices that create the traffic. IoT devices that typically does not have much computational resources fall victims to these attacks often. DDoS attacks are among the most common attacks on any infrastructure because of its relatively simple execution.

Intrusion Detection System (IDS) is implemented as the sentinel for applications that face the internet where all the applications that face the internet must be secured against the attacks that can happen in any form. IDS monitors the internet traffic, and it alerts in case of any anomalies. IDS can have three types, and they are based on the type of techniques it uses to detect the attacks. Anomaly-based, Signature-based and Hybrid are the three types of IDS. Machine Learning and Deep Learning techniques have taken the IDS much forward in case of detection and false detection. Deep Learning can improve the accuracy of the detection, and it also reduces the false alarm rate if utilised accurately.

This research focuses on implementation of Bidirectional Long Short-Term Memory (BiLSTM) based Intrusion Detection System for DDoS detection in IoT devices and utilised Conditional Generative Adversarial Network (cGAN) for data augmentation, which helps to improve the detection by providing better training to the model. The dataset utilised for this research is 'CICIoT2023' and it is entirely created for the purpose of intrusion detection. This dataset contains various attack classes but mostly focused on DoS and DDoS.

Long Short-Term Memory is an improved version of Recurrent Neural Network designed by Hochreiter & Schmidhuber. The usual RNN faced the issue of holding memory for a long time. LSTM introduced memory cells that can hold memory for a longer period for sequential data. This memory cell is controlled by three gates called Input Gate, Forget Gate, Output Gate. Input gate decides what information is added to the memory cell. Forget gate devices what information is removed from memory cell. Output gate decides what information is output from memory cell (GeeksforGeeks, 2019).

Bidirectional Long Short-Term Memory (BiLSTM) is a variation of LSTM network where the input flows in both directions and this can utilise information from both sides. BiLSTM adds an LSTM layer that reverses the direction of information flow (Baeldung, 2022).
This dual direction processing captures patterns that may have been missed when the data is processed in just one direction. The Bidirectional movement provides better context from both ends of the data point. The final output is the combined output of the two LSTM networks. BiLSTM handles variable sequence lengths properly as it makes it a good choice for handling network traffic that has irregular lengths. Here, BiLSTM is used to detect anomalies in the network traffic that indicate the possibility of DDoS attack. It will detect subtle traces in sequence data that might mean malicious activity.

Research Question: How does IDS based on BiLSTM with cGAN for data augmentation perform against DDoS attacks on IoT devices.

Objectives:

- To develop an IDS model using Bidirectional LSTM and cGAN to detect DDoS attacks in IoT devices.
- To evaluate the performance of the developed model using various metrices.

This report is based on the research done on Intrusion Detection System for IoT to detect DDoS attacks based on BiLSTM and cGAN is split into 5 sections, 'Related Work' section walks through the related works that inspired to perform this research. 'Research Methodology' explains the benefits and reason to select those technology. 'Design Specification' explains in detail about the model's specifications. 'Implementation' section walks through the steps that are performed for this research. 'Evaluation' section shows the model's performance.

2 Related Work

Benaddi, H., Jouhari, M., Ibrahimi, K., Benslimane, A. and Amhoud, E.M. (2022) propose a novel approach for detecting anomalies in IoT networks using conditional generative adversarial networks (cGAN) and hybrid deep learning model combining convolutional neural networks (CNN) and long short-term memory (LSTM) networks. They used Bot-IoT dataset, and they first trained the CNNLSTM based IDS, and they tried to identify the attack classes that has poor detection performance. They then used cGAN to generate synthetic data for those particular attack classes and it is used to retrain the IDS. For certain attack types, their results showed improvements, but the overall accuracy went down. The accuracy of Reconnaissance attacks increased from 0.36 to 0.5 and Theft attacks rose from 0 to 0.4. They effectively use of cGAN to generate synthetic data to address bata imbalance. But to generalise the model, further refinement of cGAN training process might be required.

According to Dunmore, A., Jang-Jaccard, J., Sabrina, F. and Kwak, J. (2023), various GAN models have been applied to improve the IDS itself. Deep Convolutional Generative Adversarial Network which builds on the architecture of traditional Convolutional Neural Network has shown promising results. The PassGAN model that was designed to generate password guesses by learning from real passwords has shown 51% to 71% success rate in matching passwords from HashCat dataset. Conditional GAN provided controls on output of a GAN model to create focused data samples. This is considered a benefit as the GAN can be directed in the process of sample creation. This control is that makes it better than most of other GAN models. This effective survey shows the different types of GAN models. They could have provided a performance comparison of GAN models performing various tasks, that could give a clear idea on which model to select.

Kumar, V. and Sinha, D. (2023) propose a model using XGBoost and Wasserstein Conditional Generative Adversarial Networks with gradient penalty. Their model aims to generate synthetic data for underrepresented attack classes and thus improving detection performance across various datasets including NSL-KDD, UNSW-NB15 and BoT-IoT. The study concludes that WCGAN performs better than traditional GANs in generating better quality samples that

maintain the statistical properties of real attack data. Machine learning models like XGBoost, Random Forest, Decision Tree and SVM were tested and XGBoost showed better performance in multi-class classification tasks. The authors highlight that their model significantly improves detection accuracy compared to existing models, offering robust solutions for imbalanced datasets in cybersecurity contexts.

Wu, Y., Nie, L., Wang, S., Ning, Z. and Li, S. (2023) presented a sophisticated model for intrusion detection using edge computing. They focused on the limitations of existing cloud centric computing like latency and ease of use in addressing the security challenges posed by IoT's open and distributed nature. They proposed a method combining big data mining with a fuzzy rough set, a convolutional neural network (CNN), and a generative adversarial network (GAN). This method starts with a fuzzy rough set-based algorithm for feature selection from large IoT dataset and then by applying CNN for efficient feature extraction and intrusion detection. CNN and GAN are combined to improve the detection performance. This model achieved 4% higher accuracy in simulations.

Boukhalfa, A., Abdellaoui, A., Hmina, N. and Chaoui, H. (2020) presented an approach to Network Intrusion Detection Systems (NIDS) using Long Short-Term Memory (LSTM). The proposed NIDS took advantage of the LSTM's memory retention capability to detect both existing and novel attacks by recognising patterns over extended periods. They used NSL-KDD dataset, and they achieved accuracy rates of 99.98% and 99.93% for binary and multi-class classifications respectively. The proposed LSTM model showed better performance than machine learning classifiers like SVM, KNN and Decision Tree in terms of precision, recall and false positive rates. The LSTM model attained a false positive rate of 0.068% for binary classification and 0.023% for multi-class classification. These obtained results promise the performance of LSTM in NIDS.

Liang, X., Xing, H. and Hou, T. (2023) introduced a network intrusion detection method using Convolutional Neural Network – Bidirectional Long Short-Term Memory (CNN-BiLSTM) model integrated with Conditional Generative Adversarial Network (cGAN) to address the network traffic data imbalance. Their methodology involved preprocessing raw traffic into Gaussian domains, then using CGAN to generate synthetic data to ensure balanced dataset and use CNN and BiLSTM to extract crucial spatial and temporal features respectively. They used NSL-KDD dataset, and the model showed improvements in accuracy, precision, and F1score. The use of CGAN to balance the dataset could improve the training by mimicking the real data. The combination of CNN for spatial feature extraction and BiLSTM for temporal dynamics ensures higher performance in capturing network traffic. But the deployment of this model raises concerns in terms of computational recourses and real time processing.

3 Research Methodology

The research methodology is divided into 5 phases. Data collection, data preprocessing, BiLSTM model, cGAN, and evaluation. Figure 1 shows the complete steps involved in this research work.

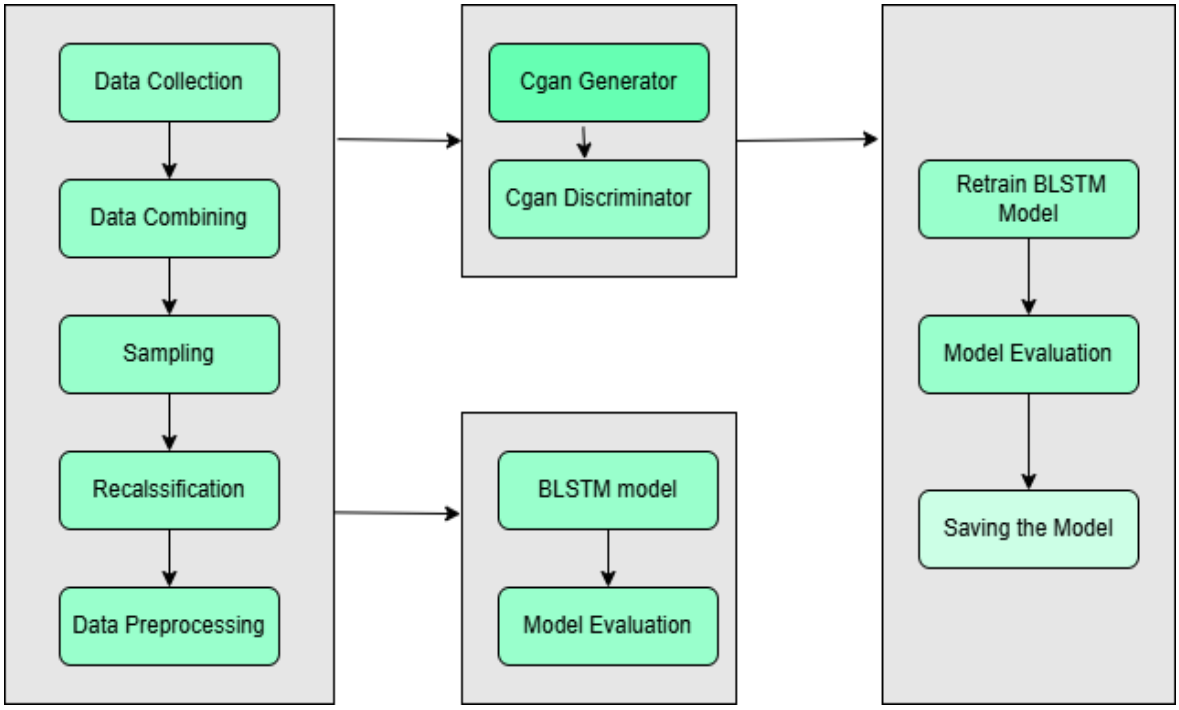

Figure 1: Research Methodology Flowchart

3.1 Data Collection

CICIoT2023 dataset (Canadian Institute for Cybersecurity, 2023) was selected for building and evaluating the BiLSTM model. This dataset consists of 33 attack scenarios which is divided into 7 classes against IoT devices. Attacks are classified as DDoS, DoS, Recon, Web-based, Brute Force, Spoofing and Mirai. Almost 164 datasets were present in CICIoT2023, this was combined and saved as single dataset named 'combined_1dataset.csv'. The combined dataset was too huge to process; therefore, dataset was sampled to 5000 samples. The dataset was refined further to improve the performance of the model, since it contained various network traffic features. The attacks in the dataset were reclassified as Benign, DDoS and attack, where 11 DDoS related attack combined under DDoS and all other attack types were grouped under 'attack label. Reclassified datasets consist of 1764 DDoS attacks, 3087 other attacks and 147 Benign traffic. Dataset includes various features which helps in detecting DDoS in IoT devices, 'flow_duration' represent total duration of the flow, where extremely short or long duration traffic may be an indication of DDoS traffic. DDoS attacks may exploit certain protocols and deviations in header sizes which can be find using the 'Header_length 'and 'Protocol Type' which indicate size of header and type of protocol used. 'Rate', Srate' -source rate and 'Drate' -destination rate, measure the speed of the traffic. TCP Flags like 'fin_flag_number', 'syn_flag_number', 'rst_flag_number' dominate different traffic types, high number of SYN flags indicate SYN flood. 'std'-standard deviation of packet size may be an indicator of DDoS. To measure the time between the packets' IAT' Inter-Arrival Time is used, rapid increase of

traffic results low IATs. Statistical features like ‘magnitude’, ‘radius’, ‘covariance’, ‘variance’, derived from the data provide details regarding traffic patterns. We can identify DDoS attacks through sudden increase in traffic over certain protocol, therefore ‘HTTP’,’HTTPS’,’DNS’ protocol are mentioned in the dataset. Through ‘Tot size’ deviation from normal packet size can be measured.

3.2 Data Preprocessing

Data preprocessing is the most important step before passing the data to the model for training. Preprocessing steps in this work include stripping spaces off the column names which ensures that no hidden spaces are present, and it helps to prevent the errors caused by hidden spaces. Infinite values in the dataframe causes issues and it must be handled properly. All the infinite values are replaced with ‘NaN’ in the dataframe. After replacing all infinite values with ‘NaN’, all the rows with ‘NaN’ values are removed to ensure training is done on complete data. Label Encoder is used to convert all the categorical labels into numeric format. Standard scaler standardizes the features by removing the mean and scaling to unit variance. Finally, data is reshaped to match the LSTM network’s expected input shape.

3.3 BiLSTM Model

The Bidirectional LSTM uses two layers of LSTM, where the data flow direction is opposite in those layers. This working provide better contextual understanding as the data is read from both directions. DDoS attacks can be identified by certain patterns that develop before and after events in the data sequence. A sudden increase in ‘Rate’, ‘Srate’ or ‘syn_flag_number’ or ‘rst_flag_number’ might indicate an attack. BiLSTM process the data forward and backward, so that it understands the before and after scenario for better decision making. Features like ‘flow_duration’, ‘IAT’ and ‘Std’ of packet sizes show temporal dependencies that can identify abnormal behaviour and BiLSTM can better understand these dependencies by integrating the information from past and future inputs. BiLSTM can integrate static and dynamic features across time, which can identify attack patterns. The traditional RNN was replaced with BiLSTM for this research because RNN struggled with handling long term dependencies. Traditional RNN’s preference goes to the recent inputs in the sequence but BiLSTM processes the early signs and the recent inputs.

In this work, Bidirectional LSTM network is used for pattern recognition, and it is crucial in detecting anomalies and changes in the traffic behaviour. This network can process the sequences with insights from both past and future context. The dual directional analysis is crucial for identifying complex patterns in network data. To detect DDoS attacks in a network traffic, where the patterns may develop over time and the importance of sequence data is crucial, BiLSTM can identify even small changes in the traffic behaviour that might be a security threat. Figure 2 shows the sequence flow in the two LSTM layers.

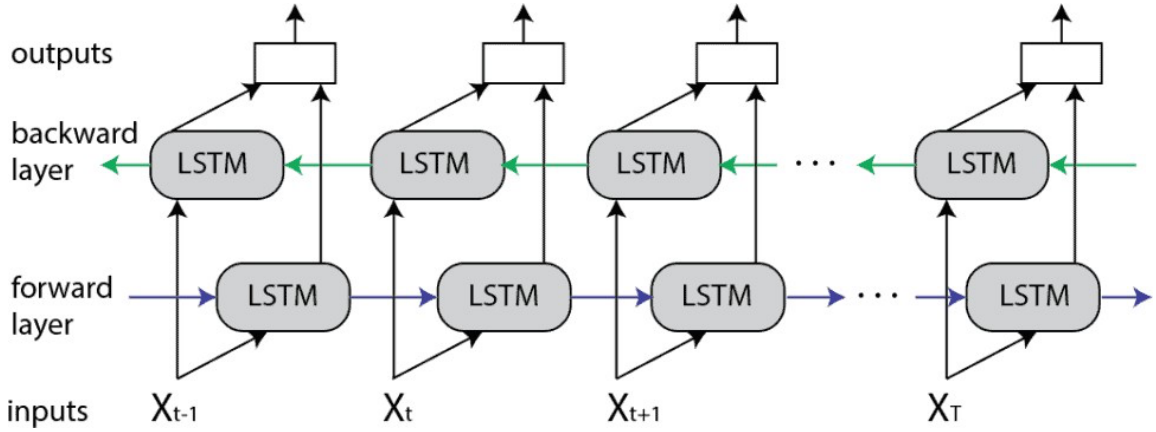

Figure 2: Bidirectional LSTM

3.4 cGAN Model

Generative Adversarial Network is a deep learning framework that is used to generate random, plausible examples based on our needs (GeeksforGeeks, 2023). There are two components for GAN, they are Generator and Discriminator. The Generator network's role is to produce or generate new data that is as real as the genuine data and makes it difficult for the Discriminator to distinguish between them. The Discriminator network work hard to examine samples and distinguish between the real data and the generated synthetic data. Its goal is to accurately distinguish between real and fake data and ensuring that generator generates high quality output.

Conditional Generative Adversarial Network is a type of GAN model that has the additional feature of generating based on conditions. These conditions can be class labels, or tags and it is very useful in situations which require controlled generation process. Mehdi Mirza and Simon Osindero first published about cGAN in 2014. After training the cGAN, the real training data and the generated synthetic data are combined to train the BiLSTM model. This updated dataset helps to improve the BiLSTM's ability to detect DDoS attacks more effectively.

In this work, BiLSTM lacks in terms of training and cGAN improves the training quality significantly. cGAN provides improved data variety by generating synthetic data and it excels in case of imbalanced classes. Even in the appearance of rare cases in the dataset, cGAN can create samples and these samples can be used to train BiLSTM to those patterns. The targeted data generation ability of cGAN is the main reason to choose cGAN to integrate with BiLSTM to train the model for DDoS attacks. Figure 3 shows the input and output of generator and discriminator.

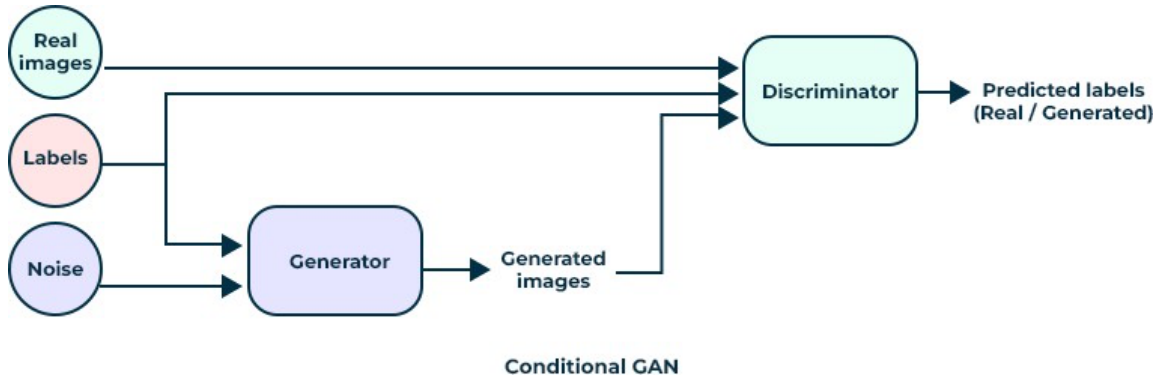

Figure 3: Conditional GAN

3.5 Evaluation

The Bidirectional LSTM model retrained using cGAN is evaluated using different metrices including Accuracy which measures the proportion of true results including true positives and true negatives. Accuracy shows the overall correctness of the model. Precision shows how much of positive indications were actually correct. Recall measures the proportion of actual positives that were correctly identified. F1 Score is the harmonic mean of precision and recall. Confusion matrix is a table used to describe the performance of a classification model. True Positive Rate and False Positive Rate are used to measure the true positive and false positives. Receiver Operating Characteristic (ROC) curve plots the true positive rate against false positive rate and Area Under Curve (AUC) provides an aggregate measure of performance across all possible classification thresholds (Google for Developers, no date). Precision-Recall curve shows the trade-off between precision and recall for different threshold (scikit-learn, no date).

4 Design Specification

4.1 Environment Setup

The Deep learning model is built with the help of Anaconda and Google Colab. Sampling and combining of the dataset were done using Jupyter notebook with Python version. Google collab helped in model building, evaluation and visualization. Python version 3.9 was compatible with all the necessary libraries required to build the model. The libraries like TensorFlow for building and training the components of CGAN and BiLSTM model, Keras along with TensorFlow for simplify the building of the model, numerical algorithms are carried out using NumPy, Pandas for data manipulation and analysis, Matplotlib and Seabone present data and results graphically, preprocessing and metric evaluation of data is done using the Scikit-learn library.

4.2 Deep Learning Model

CICIoT 2023 dataset was selected to train and evaluate the BiLSTM model which utilised cGAN for data augmentation. Features like flow_duration, Header_Length and Protocol Type, duration and rate, Srate (source rate) and Drate (Destination rate), TCPflags (fin_flag_number, syn_flag_number, rst_flag_number), std, IAT, magnitude, radius, covariance, variance, HTTP, HTTPS, DNS are used in building the model for detecting DDoS attacks in IoT devices. The downloaded datasets were combined to create a single dataset, and it was then sampled using the random sampling technique to create a dataset with 5000 samples with equal number of samples from each attack class. DDoS attack class had 11 attack scenarios and thus all DDoS attack scenarios were reclassified as 'ddos' and all the other attack classes were reclassified as 'attack'. The reclassified dataset was named 'processed_dataset' and this dataset is used as the main dataset for this model.

The data is loaded from a CSV file to Google Colab. Preprocessing steps were performed on this loaded dataset. Preprocessing steps include stripping white spaces from column headers, replacing infinite values with 'NaN'. 'LabelEncoder, was used to encode the categorical labels to numerical values, 'StandardScaler' was used to normalize the features. Then the data was reshaped to match the LSTM model's requirement. The data is then split into train and test sets. This Bidirectional LSTM model is a sequential model and it's first LSTM layer has 128 units and 'return_sequences' is True so that full sequence is passed onto the next layer. The second LSTM layer does not have 'return_sequences' setting so that output is passed to the next layer. Dropout layers are added between every LSTM layer to reduce overfitting. Dense layers are added after LSTM layers and those layers have 'ReLU' and 'softmax' activation function. Early stopping was added to stop the training process if the validation loss does not improve for five epochs. The model is then trained extensively as a bath of 64 for 50 passes through training data. The model is then evaluated based on Accuracy, Precision, Recall, and F1 Score.

Conditional GAN has two components, the Generator model is designed to accept a noise vector and a condition label and then generate synthetic data based on the condition. The noise vector is shaped as the 'latent_dim' which in this work is set as 100. The label is then embedded and reshaped to match the noise vector's dimension to make sure the label information is integrated properly with the noise. The noise and label embedding are then concatenated together to form the input to the next layer. Next a dense layer with 256 units and ReLU activation processes the concatenated vector. Another dense layer expands this processed vector, and it is passed through a sigmoid activation function to ensure the output values are between 0 and 1. The output is then reshaped to a sequence format to match the structure of real network traffic data. Discriminator accepts a sample that can be the real or synthetic data samples and label as input. The like generator, the embedding here matches the input data size, and this embedded label and sample data are concatenated together. After concatenation it is processed through dense layers with ReLU and sigmoid activation to output a single probability if the input data is real. The generator is trained to fool the discriminator to classify the fake sample as real. Discriminator is trained to ensure that it classify the real sample as real and generated sample as fake. The cGAN is trained extensively and after this, it is used to retrain the Bidirectional LSTM model to improve the performance of the model. The Bidirectional LSTM model is evaluated based on Accuracy, Precision, Recall and F1 Score. The models, Bidirectional LSTM, Generator, Discriminator, Conditional GAN are saved separately to access them later if needed.

5 Implementation

- The dataset was selected based on the features of the dataset. The downloaded dataset 'CICIoT2023' was split into many pieces, so they were combined to create a single dataset, 'combined_dataset.csv'.
- The obtained single dataset was huge to process for the local machine and Google Colab. So, the dataset was sampled using the random sampling technique. The total size of the dataset was selected as 5000 samples with equal number of attack scenarios.
- There were 11 attack scenarios for DDoS attack class, so all the DDoS attack scenarios were reclassified as 'ddos' and all the other attack classes were reclassified as 'attack'. All the 'BenignTraffic' were left untouched.
- The reclassified dataset was named 'processed_dataset' used as the main dataset for building the BiLSTM model.
- Preprocessing steps were performed on this dataset. Trim the column names by stripping off the white spaces.
- Look for infinite values and replace them with 'NaN'.

```
data.replace([np.inf, -np.inf], np.nan, inplace=True)
```

- Look for missing values and drop all the missing values.

```
data.dropna(inplace=True)
```

- Use 'LabelEncoder' to encode the labels.

```
label_encoder = LabelEncoder()
data['label'] = label_encoder.fit_transform(data['label'])
```

- Standard scaler is used to standardize the values.

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

- Build Bidirectional LSTM model. The first layer of LSTM has 'returrn_sequence' enabled.

```
model.add(Bidirectional(LSTM(128, return_sequences=True),
input_shape=input_shape))
```

- Dropout layers were added after every LSTM layer to reduce overfitting.

```
model.add(Dropout(0.3))
```

- The built model was trained extensively, runs for 50 epochs with batches of 64 samples, and evaluated based on Accuracy, Precision, F1 Score, and Recall.

```
rlstm_model.fit(X_train, y_train_cat, epochs=50, batch_size=64,
validation_split=0.2, callbacks=[early_stopping])
```

- cGAN has two components, Generator and Discriminator. First, Generator network was built based on the condition label.
- Generator network generates fake samples based on the condition label.

```
fake_sample = generator([noise, label])
```

- Build the Discriminator network to identify the fake samples correctly.
- The cGAN is trained extensively, runs for 20,000 epochs with batch size of 64 to achieve high performance.
- The trained cGAN is then used to retrain the Bidirectional LSTM model.
- After training, the final model is then evaluated based on Accuracy, Precision, Recall, and F1 Score.

6 Evaluation

The aim of the research was to develop the model and evaluate through various metrices. The performance of the model was evaluated using Accuracy, Precision, F1 Score, Recall, and ROC and AUC. The classification report and confusion matrix were also obtained for the analysis.

Accuracy shows the overall correctness of the model

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Negatives} + \text{False Positives}} \quad \text{--(1)}$$

High accuracy means that the model performs well across all classes.

Precision shows the accuracy of positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad \text{--(2)}$$

Higher precision means higher percentage of positive identifications was actually correct.

Recall measures how good the model is at finding all the positives.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad \text{--(3)}$$

High recall score shows that the model is good at capturing all positives.

F1 Score is the harmonic mean of precision and recall.

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

True Positives are the correctly predicted positive observations.
True Negatives are the correctly predicted negative observations

False Positive is the incorrectly predicted positive observations.
False Negative is the incorrectly predicted negative observations that are actually positive. Table 1 shows the performance evaluation of BiLSTM model before the integration of cGAN.

Table 1: BiLSTM Model Evaluation

Metrices	value
Accuracy	0.877
Precision	0.8681
Recall	0.877
F1 Score	0.8694

Table 2 shows the classification report of BiLSTM model.

Table 2: Classification Report - BiLSTM

	Precision	Recall	F-1 score	Support
Benign Traffic	0.43	0.11	0.18	27
Attack	0.88	0.93	0.90	613
DDoS	0.89	0.84	0.86	360
Accuracy			0.88	1000
Macro avg	0.73	0.63	0.65	1000
Weighted avg	0.87	0.88	0.87	1000

Model summary provides a brief description of the layers and structure of the model. It includes layer types, output shapes and number of parameters in each layer. The model summary helps to review the architecture of the model. Figure 4 shows the model summary of BiLSTM model.

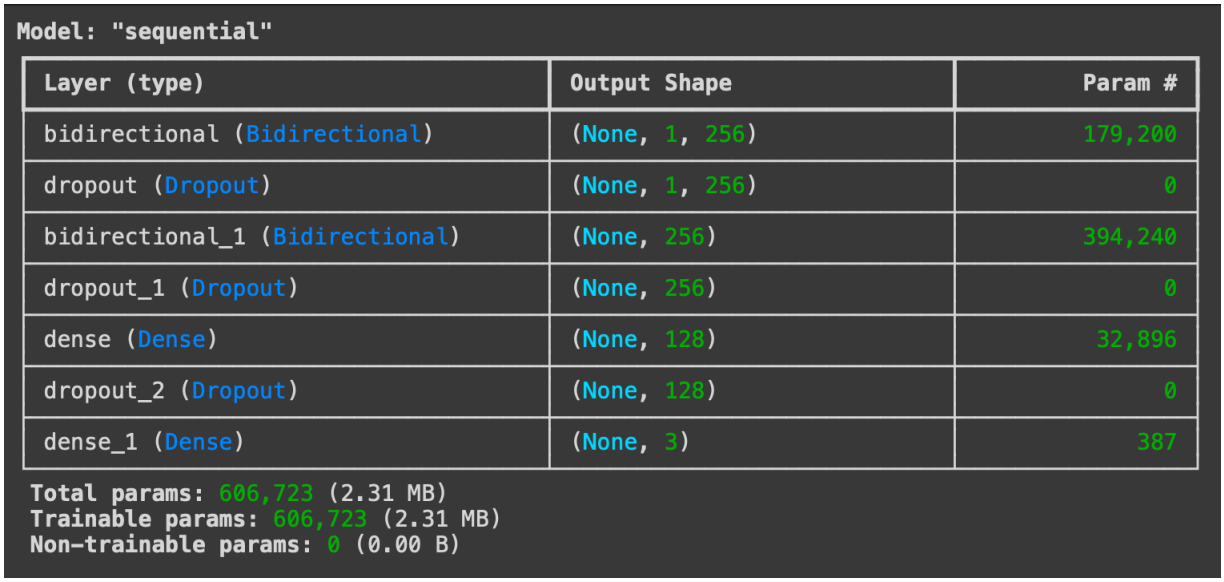

Figure 4: Model Summary

Table 3 shows the performance evaluation of BiLSTM model after integration of cGAN.

Table 3: BiLSTM Model Evaluation after cGAN

Metrices	value
Accuracy	0.886
Precision	0.8865
Recall	0.886
F1 Score	0.8840

Table 4 shows the classification report of BiLSTM model after integration of cGAN.

Table 4: Classification Report – BiLSTM Model after cGAN

	Precision	Recall	F-1 score	Support
Benign Traffic	0.42	0.30	0.35	27
Attack	0.94	0.87	0.90	613
DDoS	0.84	0.95	0.89	360
Accuracy			0.89	1000
Macro avg	0.73	0.71	0.71	1000
Weighted avg	0.89	0.89	0.88	1000

The Bidirectional LSTM model showed significant results before and after the integration of Conditional GAN. Before the integration, the accuracy was 87.8% and precision was 87.4%. And after the integration of Conditional GAN, the results improved, and the accuracy and precision became 88.2% and 87.9% respectively. Any small boost in these metrices can improve the total performance of the model. Table 5 compares the performance of BiLSTM model before the integration of cGAN and after the integration of cGAN.

Table 5: BiLSTM Comparison – Before and After cGAN

	BiLSTM	**BiLSTM after cGAN**
Accuracy	**87.7%**	**88.6%**
Precision	**86.8%**	**88.65%**
Recall	**87.7%**	**88.6%**
F1 Score	**86.9%**	**88.4%**

Confusion matrix visualize the performance of the model in three classes named Benign Traffic, attack, ddos. The rows represent the classifications and column indicate model's predictions. Each cell contains the count of predictions for each class. Top cells shows that 3 benign traffic instances were predicted correctly, 24 benign traffic were predicted incorrectly

as attacks and 0 benign predicted as DDoS attacks. At middle cells shows correctly predicted 531 instances of attacks, 4 attacks incorrectly predicted as benign traffic, and 78 attacks incorrectly predicted as DDoS. Bottom cell represent correctly predicted 348 instances of DDoS attacks, 12 incorrectly predicted DDoS attacks as regular attacks and 0 benign traffic were predicted as DDoS attacks. The high for attack and ddos classed indicates the performance accuracy level here. Figure 5 shows the confusion matrix obtained after integrating cGAN.

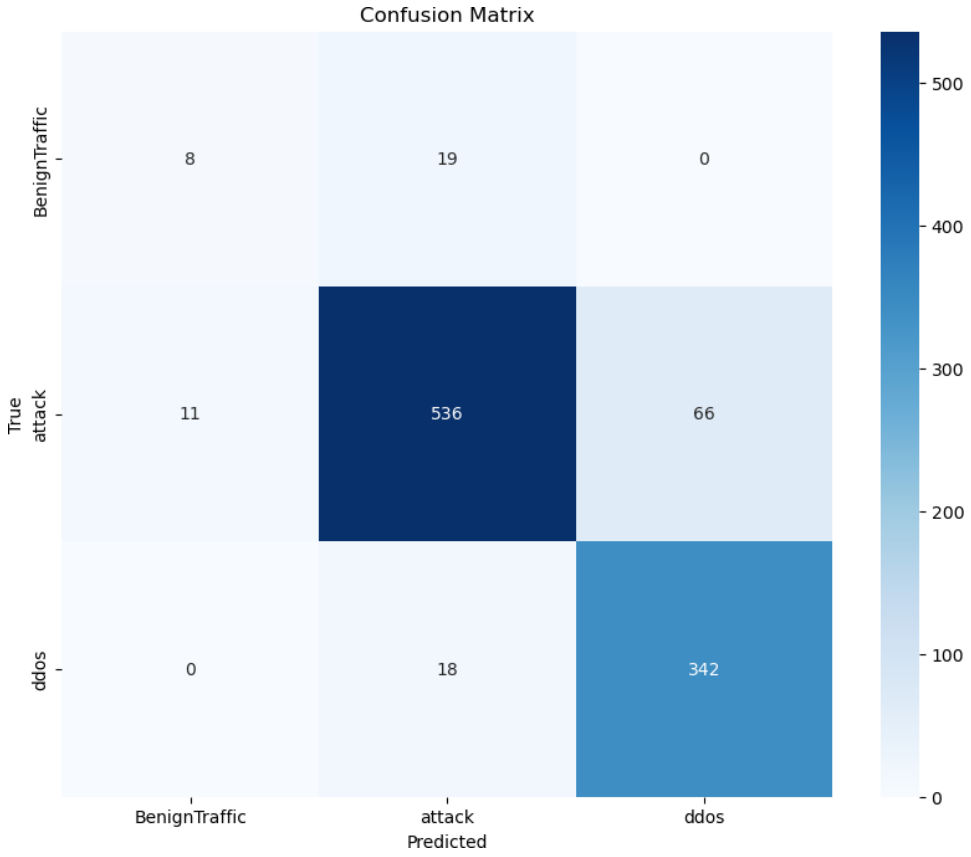

Figure 5: Confusion Matrix

Receiver Operating Characteristic Curve is the graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters; True Positive Rate, and False Positive Rate (Google for Developers, no date). Figure 6 shows the ROC curve obtained after integrating cGAN.

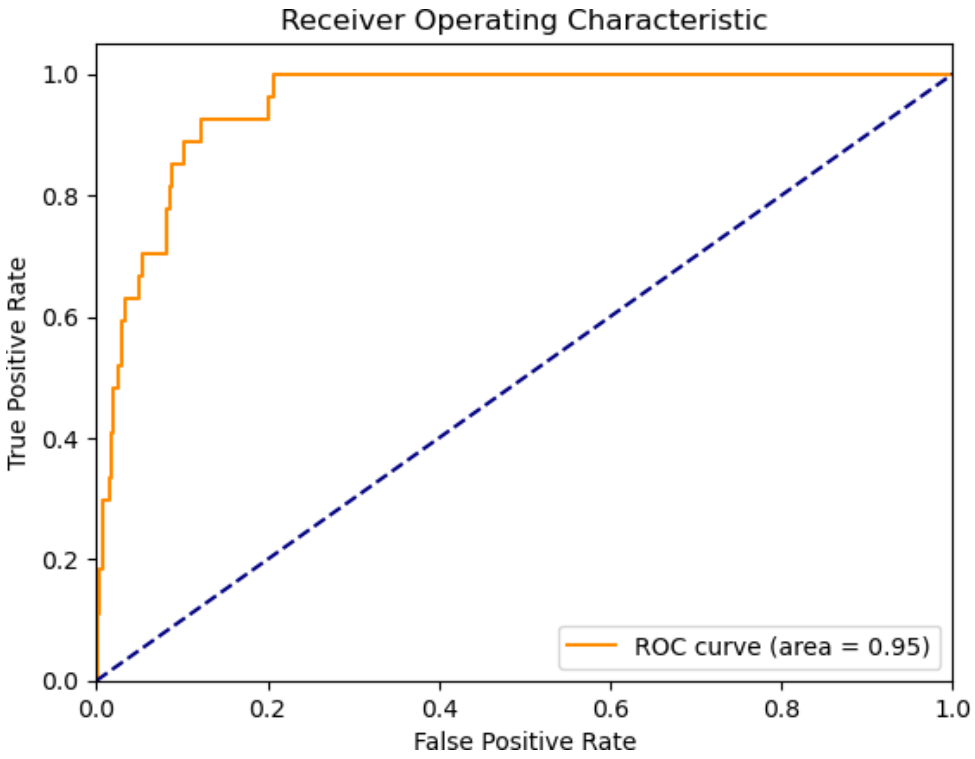

Figure 6: ROC Curve

The precision-recall curve shows the trade-off between precision and recall for different thresholds (scikit-learn, no date). Figure 7 shows the precision recall curve obtained after integrating cGAN.

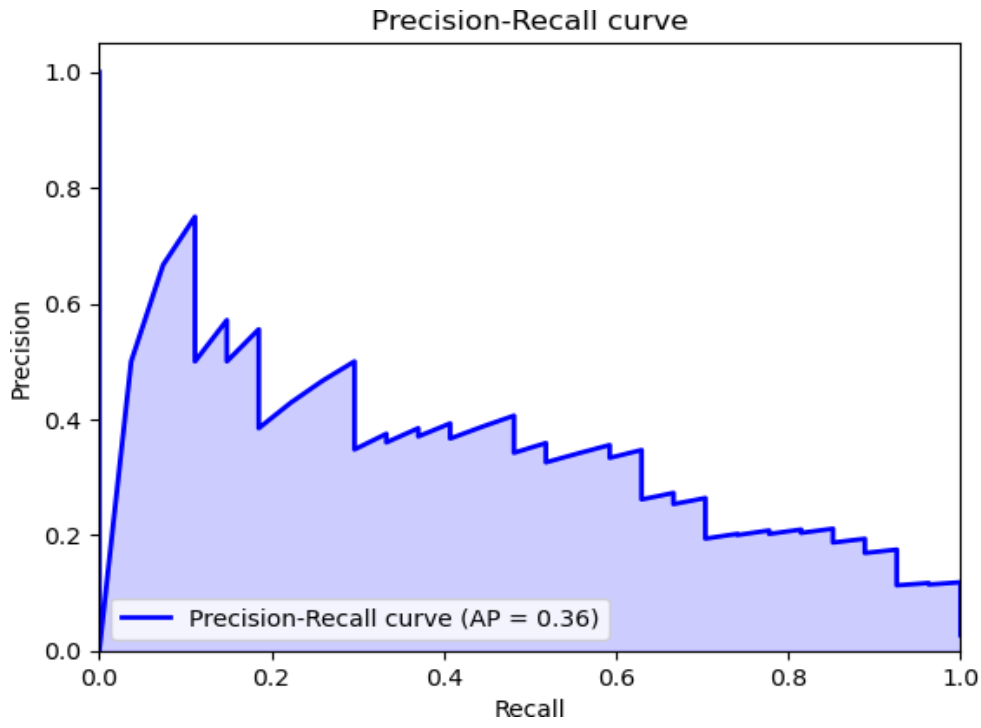

Figure 7: Precision - Recall Curve

7 Discussion

Although ideal results were obtained after this research, there were many limitations during the different stages in the process. Initially, the original dataset was split into 164 datasets, and it was combined to create a single dataset for the better performance of the model, and it was difficult to process due to its huge size. The computational requirement and time to process the dataset was high, so the dataset was sampled to 5000 samples. Initially 1% of the original dataset and a sampled dataset of 10,000 were used as experiment, still it was a slow process. Thus, sample size of 5000 were selected for this work. For the training, the selection of 20,000 epochs was made after experimenting with 5000, 10,000 and 20,000 epochs. Training the cGAN require significant computational requirement and time, 20,000 epochs for training cGAN provide balanced training and better data quality. Overcoming the limitations, the model performance was improved, and the model displayed promising results. The model can be introduced to different real time dataset, which can improve the model performance.

8 Conclusion and Future Work

This research successfully developed the Bidirectional LSTM model, integrated with Conditional GAN to detect DDoS attacks on IoT devices. The evaluation of the model successfully answers the research question how the model detects DDoS attacks in IoT devices by achieving excellent accuracy, precision, recall and F1 score. The objective was to utilise Bidirectional LSTM's capabilities such as increased contextual awareness by utilising past and future data points for accurate anomaly detection. The integration of Conditional GAN to generate synthetic data improved the model's performance. The evaluation of the model showed excellent results with overall accuracy of 88.6%, precision of 88.65%, recall 88.6% and F1 score 88.4%. These promising results prove that this model enhances the security of the infrastructure by detecting DDoS attacks in IoT devices. However, the model shows promising results, training the model consumes long time and require efficient hardware. In future, the model's prediction can be used integrate it with the IDS to implement it in real world scenarios. Also, the model can be trained to detect other types of cybersecurity threats apart from DDoS would make this model much stronger.

References

IoT devices installed base worldwide 2015-2025 (2016) Statista. Available at: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (Accessed: 28 July 2024).

What was the Mirai botnet (2018) Malwarebytes. Available at: https://www.malwarebytes.com/what-was-the-mirai-botnet (Accessed: 28 July 2024).

What is a distributed denial-of-service (DDoS) attack? (no date). Available at: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/ (Accessed: 28 July 2024).

What is LSTM - Long Short Term Memory? (2019) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/ (Accessed: 1 August 2024).

Differences Between Bidirectional and Unidirectional LSTM | Baeldung on Computer Science (2022). Available at: https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm (Accessed: 1 August 2024).

Benaddi, H., Jouhari, M., Ibrahimi, K., Benslimane, A. and Amhoud, E.M., 2022, December. *Adversarial attacks against iot networks using conditional gan based learning.* In GLOBECOM 2022-2022 IEEE Global Communications Conference (pp. 2788-2793). IEEE.

Dunmore, A., Jang-Jaccard, J., Sabrina, F. and Kwak, J., 2023. *A comprehensive survey of generative adversarial networks (GANs) in cybersecurity intrusion detection.* IEEE Access.

Kumar, V. and Sinha, D., 2023. Synthetic attack data generation model applying generative adversarial network for intrusion detection. Computers & Security, 125, p.103054.

Wu, Y., Nie, L., Wang, S., Ning, Z. and Li, S., 2021. Intelligent intrusion detection for internet of things security: *A deep convolutional generative adversarial network-enabled approach.* IEEE Internet of Things Journal, 10(4), pp.3094-3106.

Boukhalfa, A., Abdellaoui, A., Hmina, N. and Chaoui, H., 2020. *LSTM deep learning method for network intrusion detection system.* International Journal of Electrical and Computer Engineering, 10(3), p.3315.

Liang, X., Xing, H. and Hou, T., 2023, August. *Network Intrusion Detection Method Based on CGAN and CNN-BiLSTM.* In 2023 IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI) (pp. 396-400). IEEE.

IoT Dataset 2023 | Datasets | Research | Canadian Institute for Cybersecurity | UNB (2023). Available at: https://www.unb.ca/cic/datasets/iotdataset-2023.html (Accessed: 2 August 2024).

Conditional Generative Adversarial Network (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/conditional-generative-adversarial-network/ (Accessed: 3 August 2024).

Classification: ROC Curve and AUC | Machine Learning (no date) Google for Developers. Available at: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (Accessed: 5 August 2024).

Precision-Recall (no date) scikit-learn. Available at: https://scikit-learn/stable/auto_examples/model_selection/plot_precision_recall.html (Accessed: 5 August 2024).

What is Accuracy, Precision, Recall and F1 Score? (2022). Available at: https://www.labelf.ai/blog/what-is-accuracy-precision-recall-and-f1-score (Accessed: 6 August 2024).