International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Hybrid Deep Learning-Based IOT Botnet Attack Detection System

K.BHARGAVA TRIVENI NANDANA[1], DYAVA SUMANTH[2]
P.PAVAN KUMAR[3], K.BHANU PRAKASH[4]

[1]Department of CSE(AI&ML), CMR Technical Campus, Hyderabad, Telangana, India
bhargavanandana.aiml@cmrtc.ac.in

[2]Department of CSE(AI&ML), CMR Technical Campus, Hyderabad, Telangana
sumanthdyava51917@gmail.com

[3]Department of CSE(AI&ML), CMR Technical Campus, Hyderabad, Telangana
237r1a6647@cmrtc.ac.in

[4]Department of CSE(AI&ML), CMR Technical Campus, Hyderabad, Telangana
237r5a6619@cmrtc.ac.in

Abstract - Internet of Things devices have been widely are deployed at different locations like home, hospital, industrial, and transport sectors. Such devices communicate by using network making our lives easier. But if these devices become vulnerable then they can be easily attacked. A network formed by joining vulnerable devices together is known as botnet which is helpful in conducting attacks like DDoS, DoS, Mirai, UDP Flood, and many more attacks. Hence, there is a need to detect these attacks in the traffic. This research work discusses the development of an approach for IoT botnet attacks detection using CICIoT2023 dataset. Firstly, the data is preprocessed and then fed to CNN-LSTM model for classification. In CNN model, patterns are extracted using traffic data features while LSTM model learns the sequence of these features. The back-end part of the system has been developed using FastAPI framework while front-end has been developed using React. User will be able to upload traffic data and get predictions from the system. The model can be not only used for traffic prediction of individual data but also batch traffic data processing and reporting. All the classes have F1 score above 96%. The problem statement, proposed solution, data set used, results obtained, and the level of the developed system are elaborated in this paper.

Key Words: IoT, Botnet, DDoS, DoS, Mirai, CNN, LSTM, CICIoT2023, Network Security, Deep Learning, Intrusion Detection System.

1. INTRODUCTION

Currently, IoT devices are part of our everyday life. Some examples of the IoT devices include smart cameras, routers, sensors, watches, medical devices, and industrial machinery. IoT devices transmit and transmit information to other devices and servers using network technology. The no.of such devices grows yearly. As per Gartner, in 2023, the number of IoT devices connected will exceed 15 billion and it is forecasted to rise to 29 billion in 2030 [1]. Apart from the high number of connected devices, security becomes very important since it needs to be protected from any attacks. There are a lot of IoT devices which do not have sufficient processing and memory capacity. These devices operate under the power of inexpensive processors and possess limited firmware. There is a possibility that the devices will not receive any updates neither from the developer nor from the user side. Alongside these, some of the devices might feature vulnerable password options and even default ones, such as admin/admin. The attacker would be able to take advantage of this security flaw and get remote access to the device. Once these devices get compromised, they can be managed together as part of what is known as botnets. These botnets can then be used for generate large amounts of traffic on another system. A DDoS attack would be another common result of a botnet attack. A DDoS attack involves many infected computers sending traffic to the target system simultaneously. DoS attacks provide the same effect, although the number of sources of the traffic is lower in this case. One of the examples of malware which was designed specifically to infect IoT devices is Mirai. In 2016, Mirai botnet infected around 600,000 cameras and routers and created a 1.2 Tbps DDoS attack against Dyn DNS; this resulted in a denial of service for such websites as Twitter, Netflix, and Reddit [2]. But the source code of Mirai was released, and a variety of its variations came out, for example, Hajime, Mozi, IoT Reaper and Dark Nexus [3]. UDP flood, TCP flood, HTTP flood, scanning and brute force are some more types of traffic that occur in botnet attacks. The amount of traffic being too large makes manual verification of each entry impossible. The project is called "Deep Learning-based hybrid IoT Botnet Attacks Detection system". Its aim is to implement a traffic classification system using the CNN-LSTM deep learning architecture trained on CICIoT2023 dataset, capable of classifying normal traffic and different botnet attacks. Furthermore, the project implements a web application built upon FastAPI and React technologies for visualizing results,

uploading batch data and creating reports. This paper provides information about the problem statement, the objective of the project, the suggested solution architecture, the used dataset, the obtained results and the implementation status.

2. CONTRIBUTION OF THE PAPER

The contributions of this paper are:

Intrusion detection model proposed relies on the hybrid approach of CNN and BiLSTM in which the CNN assists in the discovery of local correlations within the network traffic features while BiLSTM discovers the sequential dependencies within the same.

There is an IoT attack classification model which can distinguish between normal traffic and nine types of attacks including DDoS, DoS, Mirai, UDP Flood, TCP Flood, HTTP Flood, Reconnaissance, Brute Force, and Scanning to give a deeper insight into attacks rather than just performing binary classification.

Motion stabilization and noise filtering using ROI tracking and signal processing techniques.

There is a preprocessing stage incorporated which is involved in tasks like missing value imputation, duplication deletion, min-max scaling, and one-hot encoding of labels. The preprocessing parameters obtained during training are used in the inference stage.

3. RELATED WORK

The pertinent literature for this study is comprised of the studies conducted in following three areas: datasets for network intrusions, machine learning/deep learning approaches for intrusion detection, and architectures that combine CNN and LSTM methods.

Originally, the intrusion detection was done with the use of KDD Cup 1999 dataset. The article by Mukherjee and Sharma [6] described KDD99 analysis using Decision Tree and Naive Bayes classifiers to reach accuracy above 90% for DoS and probes attacks. However, the study by Tavallaee et al. [7] showed that KDD99 has plenty of deficiencies like duplicates, sampling bias, and attack distributions, which do not depict real network traffic. The NSL-KDD dataset has been provided as an alternative, though both datasets have problems with lack of IoT-specific network traffic.

The first dataset that included IoT network traffic appeared in 2018. The N-BaIoT was presented by Meidan et al. [8], and it contains the traffic of nine devices such as Danmini Doorbell, Ecobee Thermostat, Samsung Webcam, and others with infections by Mirai and BASHLITE malware. The authors used autoencoders for the training of normal data and malware infection detection through reconstruction error. It did produce good results, but it functioned like a binary classifier and could not differentiate between the types of attacks.It was important for a security analyst to know what kind of traffic was passing through – whether DDoS attack or scanning probe.

A dataset known as Bot-IoT [9] was analyzed using the classification algorithms such as Random Forest, SVM, and LSTM. Random Forest achieved an accuracy of 93.23%. In the same paper, it is noted that tree-based classifiers were having a problem with over-fitting with respect to the features in the laboratory environment.

The most recent IoT intrusion detection benchmark is the CICIoT2023 [10] proposed by the Canadian Institute for Cybersecurity. This benchmark has 105 devices' traffic with 33 attack types classified into seven categories. They were classified with the Random Forest, CNN, and feedforward neural network classifiers. Highest accuracy in this benchmark was achieved with the use of Random Forest and with more than 40 selected features. Hybrid models of CNN-LSTM have been investigated recently for detecting intrusion attacks. Kim et al. [11] developed CNN-LSTM by using the 2D CNN with LSTM on the CICIDS2017 benchmark to achieve 98.6% accuracy for five categories of attacks but their model incurred additional computational cost. Ferrag et al. [12] carried out comprehensive comparison study on deep learning techniques on several IoT benchmarks, where hybrid models always performed better than individual models. In their case, CNN-LSTM has been developed with unidirectional LSTM.

One of the recurring findings within the research literature is that models alone are either capable of learning spatial patterns or temporal patterns, but not both. The hybrid systems overcome this constraint. Another recurring finding within the literature is that most of the research has concentrated only on the precision of the model rather than thinking about how to deliver the predictions to a security analyst.

Table -1: Related Work

Ref.	Author(s)	Year	Method	Key Contribution
[1]	Statista Research Department	2023	IoT Statistics	Reported the growth of connected IoT devices worldwide
[2]	Kolias et al.	2017	DDoS / Botnet Analysis	Studied Mirai and other IoT botnets and their DDoS attacks
[3]	Bertino & Islam	2017	IoT Botnet Security	Discussed botnets and security challenges in IoT environments
[4]	Buczak & Guven	2016	Machine Learning	Reviewed data mining and machine-learning methods for cybersecurity intrusion detection
[5]	Moustafa et al.	2019	Statistical Flow Features	Proposed an ensemble intrusion detection approach using statistical flow features for IoT traffic
[6]	Mukherjee & Sharma	2012	Decision Tree + Naive Bayes	Applied classical machine-learning classifiers to intrusion detection using KDD99

Ref.	Author(s)	Year	Method	Key Contribution
[7]	Tavallaee et al.	2009	KDD99 Analysis	Identified problems in KDD99 and introduced the need for improved intrusion-detection datasets
[8]	Meidan et al.	2018	Deep Autoencoders	Developed N-BaIoT for network-based detection of IoT botnet attacks
[9]	Koroniotis et al.	2019	Random Forest, SVM, LSTM	Introduced the Bot-IoT dataset for realistic IoT botnet and network forensic analysis
[10]	Neto et al.	2023	Random Forest, CNN, Neural Networks	Introduced CICIoT2023, a large-scale IoT intrusion-detection benchmark
[11]	Kim et al.	2016	CNN + LSTM / LSTM-based IDS	Investigated deep recurrent learning for network intrusion detection
[12]	Ferrag et al.	2020	Deep Learning	Compared deep-learning approaches, datasets, and intrusion-detection methods
[13]	Goodfellow et al.	2016	Deep Learning	Provided foundational concepts and techniques for deep-learning models
[14]	Hochreiter & Schmidhuber	1997	LSTM	Introduced the Long Short-Term Memory architecture for sequence learning
[15]	Kingma & Ba	2015	Adam Optimizer	Introduced Adam for efficient optimization of neural-network parameters

4. NOVELTY OF PROPOSED WORK

In this research, the designed IoT botnet attack detection approach employs an innovative deep learning approach that utilizes the Convolutional Neural Network (CNN) combined with the BiLSTM network. The key idea is to simultaneously recognize not only the relationships between local features but also the temporal patterns of the traffic in order to achieve higher attack classification accuracy. Therefore, the current system outperforms the existing solutions that use signature-based detection, classical machine learning algorithms, as well as separate CNN and LSTM networks. In particular, the proposed system is able to classify ten categories of traffic including Normal Traffic, DDoS, DoS, Mirai, UDP Flood, TCP Flood, HTTP Flood, Reconnaissance, Brute Force, and Scanning, which makes it much more applicable in practice. Besides, the system has an automated preprocessing pipeline, threat assessment module and the web framework based on FastAPI and React. The efficiency of the system has been confirmed experimentally on CICIoT2023 dataset achieving 98.65% accuracy in comparison with separate CNN and LSTM models.

5. PROPOSED METHODOLOGY

The methodology There is a methodology that employs a hybrid deep learning-based technique to detect and classify attacks on IoT bots. The model utilizes the CICIoT2023 dataset in addition to samples from other datasets called Bot-IoT and N-BaIoT. The data is then preprocessed through the process of cleaning, treating the missing values, eliminating duplicates, normalization of the features and label encoding. The data is then fed to the hybrid CNN-BiLSTM model where CNN detects the local relations between the features in the traffic data, and BiLSTM detects the sequential relations. The model is then used to classify the network traffic into ten classes along with providing the severity level of the detected traffic.

5.1 Dataset Description

CICIoT2023 dataset comprises the main component in the utilized dataset in the proposed approach. In this dataset, the network traffic generated by 105 IoT devices is presented along with PCAP captures and CSV feature extraction. CSV traffic features used for traffic classification in the proposed technique. Bot-IoT and N-BaIoT dataset are also used to complement the attack categories in the datasets. Traffic data is classified into 10 traffic classes, namely, Normal Traffic, DDoS, DoS, Mirai, UDP Flood, TCP Flood, HTTP Flood, Reconnaissance, Brute Force, and Scanning. Each traffic data comprises 18 traffic features including traffic volume, protocol, statistics, flow rate, protocol type, and entropy.

Datasets Used:

- CICIoT2023
- Bot-IoT
- N-BaIoT

Attack Categories:

- Normal Traffic
- DDoS
- DoS
- Mirai
- UDP Flood
- TCP Flood
- HTTP Flood
- Reconnaissance
- Brute Force
- Scanning

5.2 Data Collection and Input

Step one involves importing the IoT network traffic data. The proposed algorithm will use predominantly the CSV traffic data whose network flow features have been extracted. Traffic data will be imported in bunches where each bunch represents a a traffic flow which had been analyzed by the system. The selected 18 features consist of flow duration, packets, total bytes, header length, TCP flag, packet size, flow rates, protocol type, and entropy.

Tools Used:

- Python
- Pandas
- NumPy

5.3 Data Cleaning and Preprocessing

Preprocessing of traffic data obtained is done before it is used to train deep learning model. Imputation of missing numeric attributes is done through the use of the median of the training dataset. Remove the duplicates done in order to minimize data redundancy. Numeric data attributes are normalized through the use of the min-max normalization technique. Attack classes are transformed into integers within the range of 0 to 9. Data split is done through 80/20 split.

Algorithms / Techniques Used:

- Median Imputation
- Duplicate Removal
- Min-Max Normalization
- Label Encoding
- Stratified Train-Test Split

5.4 Feature Selection and Representation

The above suggested architecture considers the selection of 18 network traffic features depending on their relevance to IoT botnets' traffic characteristics. Volume-based, protocol-based, statistical, flow rate-based, protocol type and entropy features constitute the list of chosen features. Feature vector after processing becomes the input of the CNN module of the suggested hybrid architecture.

Features Used:

- Flow duration
- Packet count
- Total bytes
- Header length
- TCP flag information
- Packet length statistics
- Flow inter-arrival time
- Flow bytes per second
- Flow packets per second
- Protocol type
- Entropy

5.5 CNN-Based Local Feature Extraction

The first layer in deep learning is a 1-Dimensional Convolutional Neural Network (CNN). The 18 feature input vector is converted into a three-dimensional tensor. In the Conv1D layer, a filter size of 64 and kernel size of 3 is used by the CNN to detect relationships between adjacent traffic features. After batch normalization, ReLU activation function is used followed by max pooling and dropout.

Algorithms / Layers Used:

- Conv1D
- Batch Normalization
- ReLU Activation
- Max Pooling
- Dropout

5.6 Bidirectional LSTM-Based Sequential Learning

Output from the CNN is fed to a bidirectional LSTM (BiLSTM). The BiLSTM layer processes the feature sequence both in the forward and reverse direction in order to learn dependency among the features extracted from the traffic data. Output of the two directions is then combined to create a 256-dimensional representation.

Algorithm Used:

- Long Short Term Memory Networks

5.7 Multi-Class Attack Classification

The learned representation from BiLSTM is fed to fully connected layers. The first layer of fully connected layers converts the representation into 64 dimensions, whereas the final layer outputs ten numbers representing ten traffic classes. Probability for each class is calculated using softmax, and the class with the highest probability is chosen as the predicted traffic class.

Classification Classes:

- Normal Traffic
- DDoS
- DoS
- Mirai
- UDP Flood
- TCP Flood
- HTTP Flood
- Reconnaissance
- Brute Force
- Scanning

5.8 Model Training and Evaluation

CNN-BiLSTM model is optimized with the aid of Adam optimizer and cross entropy loss function. Batch size is taken as 32 and the training epoch number is taken as 15. Performance metrics include accuracy, precision, recall, F1 score, ROC-AUC score, and confusion matrix. The model suggested attains an accuracy of 98.65%, a macro precision is 97.82%, a macro recall of 97.45%, a macro F1 score of 97.61%, and ROC-AUC of 99.12%.

Algorithms / Metrics Used:

- Adam Optimizer
- Cross-Entropy Loss
- Accuracy
- Precision
- Recall
- F1-Score

• ROC-AUC

• Confusion Matrix

5.9 Threat Severity Assessment

Following the classification process, the system classifies the severity of the threat depending on the category of the attack. Critical is given to DDoS, Mirai, and DoS attack categories, while High is given to UDP flood, TCP Flood, and HTTP flood. Medium is given to the Reconnaissance and Scanning categories, and Low is given to Normal Traffic.

5.10 Hybrid Deep Learning-Based IoT Botnet Attack Detection System



Fig -1: System Architecture

6. EXPERIMENTAL SETUP

6.1 Datasets Details

The designed framework was assessed in view of the CICIoT2023 data set, which was chosen as the main dataset, together with additional traffic samples extracted from the Bot-IoT and N-BaIoT data sets. The CICIoT2023 data set includes network traffic from 105 different IoT devices and provides PCAP and CSV-based features for the traffic. The CSV-based features were used In this research to classify traffic. There are altogether ten kinds of traffic in the datasets: Normal Traffic, DDoS, DoS, Mirai, UDP Flood, TCP Flood, HTTP Flood, Reconnaissance, Brute Force, and Scanning. Every traffic flow has 18 selected network features.

6.2 Hardware Configuration

The proposed system was evaluated through a computing environment that has the ability to support training and inference of deep learning models written in Python language. In the paper, it has been stated that the system can run on a machine having either of the following specifications: Apple M series processors or Intel i5 and higher processors along with at least 8 GB RAM and around 2 GB storage space for datasets and models. In this work, a GPU is not mandatory to train the model.

6.3 Software Environment

The design was developed using Python 3.10 along with PyTorch 2.x as the deep learning library. Pandas, NumPy, and Scikit-learn were used to carry out various data processing and preprocessing tasks. The backend was built using FastAPI along with Uvicorn, whereas the frontend was coded using React 18, TypeScript, and Tailwind CSS. Visualization was done using Recharts, whereas data storage and prediction results were stored in SQLite database. PDF report generation was performed using ReportLab. The trained CNN-BiLSTM architecture was saved as a serialized PyTorch model and loaded by the backend for inference.

6.4 Train and Test Procedure

Firstly, traffic data underwent preprocessing to improve the quality of the consistency of the data input. The missing numerical values were imputed using the median of those in the training set. The duplicate rows were eliminated. The numerical attributes were normalized using min-max scaling to have values within the range [0, 1]. The ten attack types were mapped to integer labels ranging from 0 to 9. The preprocessed data was split into training and testing data such that each class had the same proportion in the two data sets, i.e., 80% and 20%, respectively.

The CNN-BiLSTM model was trained with the Adam optimizer at a learning rate of 0.001 with cross-entropy loss function. A batch size of 32 was used in the training process with 15 epochs of training. During training, training and validation loss were considered to detect any signs of overfitting. The trained model was then tested on the test data set with metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix. Our model consists of accuracy of 98.65%, macro precision of 97.82%, macro recall of 97.45%, macro F1-score of 97.61%, and ROC-AUC of 99.12%.

6.5 Implementation Details

The proposed system makes use of programming in Python language in data pre-processing, deep learning network training, testing, and network traffic classification. The suggested system analyzes IoT network traffic from the CICIoT2023 datasets supplemented with additional samples from the Bot-IoT and N-BaIoT datasets. The traffic data is then pre-processed using techniques such as missing value imputation, duplicate elimination, min-max scaling, and label encoding.

For the implementation of this research, PyTorch is used in constructing the suggested hybrid CNN-BiLSTM network. The 18 features of the network traffic are presented to the Conv1D layer, where the local dependencies between the network traffic features are learned. Batch Normalization, ReLU, Max Pooling, and Dropout are used in order to acquire the features better and prevent overfitting. Next, the representation is presented to the Bidirectional LSTM, where the sequential dependencies are learned in both directions. Then fully-connected layers are used for classification to ten categories of attacks.

The model was trained using the Adam optimizer, cross entropy loss, batch size is 32, and 15 training epochs. While predicting, the same preprocessing parameters as during the training stage are used on incoming traffic data for consistency. Then, the model gives the attack type and confidence value as well as the severity of the threat. DDoS, DoS, and Mirai are

classified as CRITICAL, UDP Flood, TCP Flood, and HTTP Flood as HIGH, Reconnaissance and Scanning as MEDIUM, while Normal Traffic is LOW.

7. EVALUATION METRICS

Some of the evaluation measures, including accuracy, precision, recall, F1 measure, ROC-AUC, and confusion matrix, have been employed in the analysis of the proposed detection method to measure the effectiveness of the CNN-BiLSTM model in detecting normal IoT network traffic and different types of botnet attacks.

7.1 Accuracy

Several performance measures such as Accuracy, Precision, Recall, F1 Score, ROC-AUC, and Confusion Matrix were considered to test the effectiveness and efficiency of the proposed botnet detection method using CNN-BiLSTM algorithm for classifying normal and botnet attacks in IoT network traffic.

7.2 Precision

Shows how likely it is that a positive prediction will be right, and gives us some information on how reliable positive predictions will be.

7.3 Recall (Sensitivity)

It shows which model is able to disregard all the real positive cases.

7.4 F1-Score

Harmonic mean of Precision and Recall is taken into account in the algorithm. This aids in achieving the maximum possible balance during evaluation of model performance.

8. RESULTS AND ANALYSIS

To evaluate the efficiency of the suggested Hybrid Deep Learning-Based IoT Botnet Attack Detection System, the datasets of the network traffic taken from CICIoT2023 were pre-processed along with Bot-IoT and N-BaIoT datasets. Next, these traffic features were processed through the suggested hybrid CNN-BiLSTM model to recognize normal traffic and different types of IoT botnet attacks. The hybrid model has achieved 98.65%, 97.82%, 97.45%, 97.61%, and 99.12% accuracy, precision, recall, F1 score, and ROC-AUC respectively. It can be easily concluded that the suggested hybrid system works effectively for multi-class IoT attacks classification and outperforms the standalone CNN and LSTM models. The hybrid system also provides the features of threat severity analysis, batch traffic analysis, dashboard, and report generation via web application.

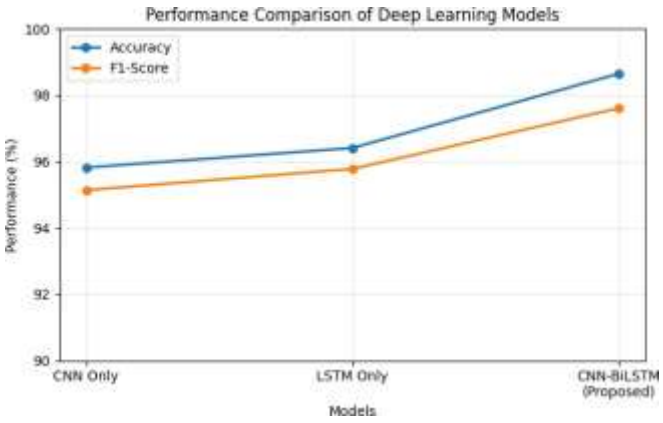


Fig -2: Performance Comparison of Deep Learning Models

8.1 Measurement heart rate.

8.1.1 Estimation Performance

SentinelIoT is a web-based network botnet detection tool that attempts to analyze IoT traffic and to see whether the traffic is malicious or not. SentinelIoT uses a deep learning method named CNN-BiLSTM for classifying the traffic on the basis of attack categories. SentinelIoT is equipped with an interface where the security analyst can view traffic data, models, attack categories, threat level, and predictions. SentinelIoT can be applied for predicting individual flows as well as in bulk via CSV files.



Fig -3: Performance Estimation

8.2 Output Of Proposed Contactless system

The dashboard provides an overview of the current traffic condition and attack detection results. It consists of the total number of traffic records processed, normal traffic records, attack traffic records, and the number of attack types detected. For the current batch, the total number of traffic records processed is 502, out of which the number of normal traffic records is 302, while the number of attack traffic records is 200. The dashboard further provides information regarding the performance with respect to pertinent evaluation metrics such as 98.65% accuracy, 97.82% macro precision, 97.45% macro recall, 97.61% macro F1 score, and 99.12% ROC-AUC.



Fig -4: Cyber Dashboard

8.3 Measurement heart rate.

The Attack Analytics tool provides an in-depth analysis of the performance of the CNN-BiLSTM classifier. The comparison is made between the hybrid classifier and the individual CNN and LSTM classifiers using the two measures: accuracy and macro F1 score. From the analysis presented by the tool, the CNN classifier achieves an accuracy of 95.82% and a macro F1 score of 95.14%, whereas the LSTM classifier achieves an accuracy of 96.41% and macro F1 score of 95.78%.

8.4 Measurement heart rate.

The Batch Prediction tool offers the option to import a CSV file with many instances of network traffic and classify them all at once. It comes with a sample CSV file along with the necessary features for the network flows, such as flow duration, packet count, byte count, header length, TCP flag count, packet length distribution, flow rate, protocol type, and entropy. Upon completion of the process, the batch report will consist of the total number of records, number of normal and attack records, detected categories, and prediction accuracy score.



Fig -5: Prediction Engine After Uploading Dataset

8.5 Classification Performance

The Hybrid CNN-BiLSTM IoT botnet attack detection system developed for this research work was tested for its effectiveness in terms of performance metrics, and it scored an accuracy of 98.65%. This shows that most of the IoT traffic was successfully classified.

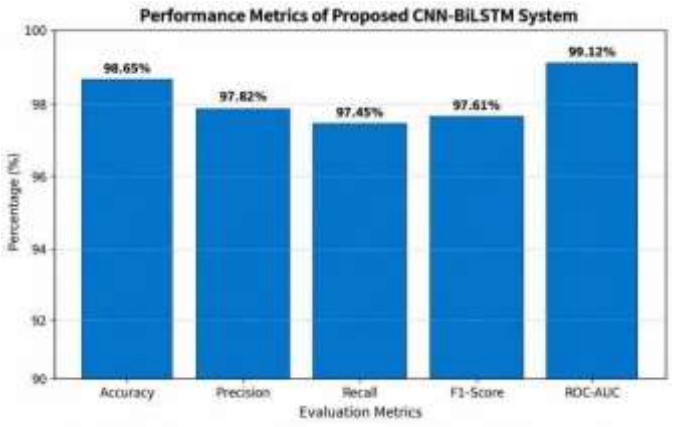


Fig -6: Performance Metrics Of Proposed CNN-LSTM System

8.5.1 Metrics for Performance Evaluation

It achieved a macro-precision of 97.82%, indicating a good precision rate in all the ten classes, while the macro-recall rate of 97.45% shows that the model performed well by recognizing most of the examples in each category. The suggested system scored a macro-F1 score of 97.61%, which implies a good balance between the precision and recall rates. Furthermore, the system received a ROC-AUC score of 99.12%, indicating an excellent ability to distinguish between the different classes.

8.6 Data Preprocessing and Feature Extraction

The suggested IoT botnet detection framework adopts a systematic process for preprocessing in order to improve the quality and consistency of network traffic data. Preprocessing phase takes care of dealing with missing values through median imputation and elimination of duplicates in the data. The numerical features of traffic data are standardized through min-max normalization to transform the feature values in the range of [0, 1], whereas the attack labels are encoded into ten classes. The post preprocessing phase selects 18 network traffic features representing traffic behavior and feeds these into the hybrid CNN-BiLSTM model.

Table -2: Performance Metrics of is Proposed System

Metric	Value
Accuracy	98.65%
Precision	97.82%
Recall	97.45%
F1-Score	97.12%

8.7 Robustness Under Different IoT Traffic Patterns

In order to check the efficiency of suggested IoT botnet detection framework, the CNN-BiLSTM architecture was experimented on various classes of IoT network traffic. It should be noted that this suggested model was able to classify not only normal IoT network traffic but also various attacks like DDoS, DoS, Mirai, UDP Flood, TCP Flood, HTTP Flood, Reconnaissance, Brute Force, and Scanning. Hybrid network classification is done by using CNN based local feature extraction as

well as Bidirectional LSTM based sequential learning to detect various features IoT traffic on networks. From the results, it observed that 98.65% overall accuracy rate was obtained by the proposed IoT botnet detection framework.



Fig -7: F1-Score Across IoT Traffic and Attack Categories

8.7.1 F1-Score Across IoT Traffic and Attack Categories

From the graph, it is evident that the proposed CNN-BiLSTM model sustains high accuracy in classifying all the ten categories of IoT traffic. In terms of F1-score, Normal Traffic has the highest score at 99.2%, then the DDoS traffic at 98.2% and Scanning at 98%. All the other types of attacks sustain an F1-score of more than 96%, with TCP Flood and Brute Force having the lowest F1-scores at 96.7%.

8.8 Lessons Learned from Related Work

The efficiency of the suggested detection system of IoT-based botnet attacks was compared with that of some traditional intrusion detection systems to evaluate its performance. Signature-based techniques work very effectively on known types of attacks but do not work when new or slightly modified attacks are present. On the other hand, anomaly-based techniques can suffer from the problem of high false-alarm rate in a heterogeneous environment of the IoT system. Conventional machine-learning techniques like Random Forest and SVM need feature engineering whereas single CNN and LSTM models cannot detect the relationship between the local features and sequential information, respectively. To overcome these limitations, the proposed technique uses CNN and Bidirectional LSTM together to analyze local as well as sequential dependencies from the network traffic of IoT. According to the experimental results, the proposed CNN-BiLSTM model attained an accuracy and F1-score of 98.65% and 97.61%, respectively. This shows the superiority of the suggested approach over the individual models.

Table -3: Comparison of Proposed Method with Existing

Method	Technique Used	Heart Rate Error	Accuracy (%)
CNN Only	Convolutional Neural Network	95.82	95.14
LSTM ONLY	Long Short – term Memory	96.41	95.78
Proposed Method	CNN+ Bidirectional LSTM (CNN - BiLSTM)	98.65	97.61

In order to evaluate the efficiency of the suggested detection system for the IoT botnet attacks, the performance of the hybrid model of CNN-BiLSTM was compared with the individual deep learning models. Signature-based intrusion detection methods can fail to detect novel attack signatures while anomaly-based and machine learning-based intrusion detection systems face some shortcomings in case of heterogeneous IoT environment. The standalone CNN model can recognize only local feature pattern and LSTM can learn only sequence dependency. But in the instance of the proposed detection system, CNN can perform feature extraction and BiLSTM learns the sequences from IoT data traffic. From the experiment, it is clear that the proposed CNN-BiLSTM model outperforms standalone CNN and LSTM models with an accuracy rate of 98.65% and F1-score 97.61%, whereas 95.82% and 96.41% accuracy were realized in the instance of CNN and LSTM respectively.

9. COMPARATIVE AND STATISTICAL

9.1 Quantitative Comparision

9.1.1 Accuracy and F1-Score Comparison of Deep Learning Models.

The proposed IoT botnet attack detection approach has performed better in relation to classification than the standalone deep learning techniques used in a study. The accuracy rate has gone up from 95.82% achieved by the standalone CNN and 96.41% achieved by the standalone LSTM technique to 98.65% for the proposed CNN-BiLSTM approach, with the F1-score increasing from 95.14% and 95.78% for the standalone CNN and LSTM approaches to 97.61%. This means that using the CNN-based local feature extraction in combination with the BiLSTM-based sequential learning is quite helpful in generating more reliable representation of IoT traffic. The standalone CNN can learn local feature dependency, while the standalone LSTM learns sequential dependencies.

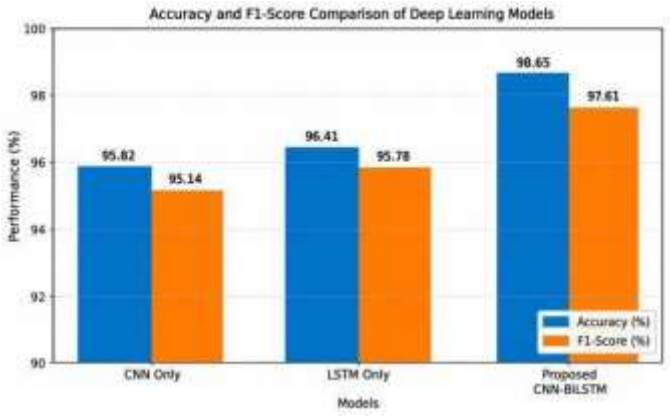


Fig -8: Accuracy and F1-Score Comparison of Deep Learning Models

9.2 Class-wise Performance of a Proposed System

Table -4: Class-wise Performance of Proposed System

Attack Category	Precision (%)	Recall (%)	F1-Score (%)
Normal Traffic	99.1	99.4	99.2
DDoS	98.4	98.1	98.2
DoS	97.5	96.9	97.2
Mirai	97.6	98.0	97.8
UDP Flood	97.2	96.8	97.0
TCP Flood	96.9	96.5	96.7
HTTP Flood	97.8	97.3	97.5
Reconnaissance	98.2	97.6	97.9
Brute Force	97.0	96.4	96.7
Scanning	98.5	97.5	98.0
Macro Average	**97.82**	**97.45**	**97.61**

Evaluation of the proposed CNN-BiLSTM model on a class basis shows very good results for all ten types of IoT traffic. The F1-score value is the highest for Normal Traffic, and it equals 99.2%, where the value of precision is 99.1% and recall is 99.4%. The performance of a model is also good for DDoS, Mirai, Reconnaissance, and Scanning classes, and their F1-scores are equal to 98.2%, 97.8%, 97.9%, and 98.0%, respectively. TCP Flood and Brute Force receive the lowest F1-scores of 96.7%, since some real high-throughput TCP connections are considered as TCP Flood and some fast legitimate login attempts are regarded as Brute Force.

9.3 Ablation Analysis

Table -5: Performance of Ablation Analysis

Configuration	Accuracy (%)
Without CNN — LSTM Only	96.41%
Without LSTM — CNN Only	95.82%
With CNN + BiLSTM — Proposed System	98.65%

10. CONCLUSION AND FUTURE SCOPE

In this paper, an algorithm named Hybrid Deep Learning-Based IoT Botnet Attack Detection System is introduced for identifying malicious traffic in IoT network. In the proposed algorithm, CICIoT2023 is utilized as the main dataset, where Bot-IoT and N-BaIoT are considered the auxiliary datasets, where 18 features related to network traffic are analyzed. Then, the dataset will be preprocessed to be used as input of the hybrid model CNN-BiLSTM. In this model, the CNN part will obtain the local information from the network traffic features, while the BiLSTM will identify the temporal information of the traffic sequence. As a result, the accuracy rate of the algorithm in experiments was 98.65%, where the macro precision rate was 97.82%, the macro recall rate was 97.45%, the macro F1-Score was 97.61% and ROC-AUC was 99.12%.

We are interested in improving the performance of the model by testing its ability to perform and adapt in different types of networks and attack environments through additional IoT datasets and benchmarking. The model could also be incorporated to work with a live network tap for traffic analysis, and not just the static CSV datasets. For future research, we would need to analyze byte-level information from packets to detect attacks and evasion tactics in the header, as well as deploying the trained model on edge devices like the Raspberry Pi.

There might be more improvement options for the model, such as using an attention mechanism in LSTM to get insight into which traffic features contribute to a specific prediction. Another method to consider is the one of federated learning where many companies could train one detection model without exchanging raw traffic data.

REFERENCES

1. Statista Research Department, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030," Statista, 2023. [Online]. Available: https://www.statista.com
2. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," Computer, vol. 50, no. 7, pp. 80–84, 2017.
3. E. Bertino and N. Islam, "Botnets and Internet of Things security," Computer, vol. 50, no. 2, pp. 76–79, 2017.
4. A. L. Buczak and E. Guven, ""Survey of Data Mining and Machine Learning Approaches to Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol.18, no. 2, pp. 1153–1176, 2016.
5. N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things,"vol. 6, no. 3, pp. 4815–4830, 2019.
6. S. Mukherjee and N. Sharma, "Intrusion detection through naive Bayes classifier using feature reduction," Procedia Technology, vol. 4, pp. 119-128, 2012.
7. Tavallaee M., Bagheri E., Lu W., and Ghorbani A. A., "A Detailed Analysis of the KDD CUP 99 Data Set," Proc. IEEE CISDA, Ottawa, Canada, 2009, pp. 1-6.

8. Meidan Y. et al. "N-BaIoT: Network-based IoT botnet attack detection using deep autoencoder," IEEE Pervasive Computing, Vol. 17, No. 3, pp. 12-22, 2018.
9. Koroniotis N., Moustafa N., Sitnikova E., Turnbull B. "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset". Future Generation Computer Systems, vol. 100, pp. 779–796, 2019.
10. E. C. P. Neto et al., "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," Sensors, vol. 23, no. 13, p. 5941, 2023.
11. Kim J., Kim J., Thu H.L.T., and Kim H., "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," Proc. PlatCon, Busan, South Korea, 2016, pp. 1–5.
12. M.A. Ferrag, L. Maglaras, S. Moschoyiannis, H. Janicke, "Deep learning for cyber security intrusion detection: approaches, datasets and comparative study," Journal of Information Security and Applications, Vol. 50, p. 102419, 2020.
13. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.
14. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
15. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. ICLR, 2015.
16. A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," Future Generation Computer Systems, vol. 82, pp. 761–768, 2018, doi: 10.1016/j.future.2017.08.043.
17. R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in Proc. IEEE Security and Privacy Workshops (SPW), 2018, pp. 29–35, doi: 10.1109/SPW.2018.00013.
18. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116, doi: 10.5220/0006639801080116.