Exploring Machine Learning Approaches for Robust Anomaly Detection and Responsive Security in IoT Frameworks

MSc Research Project
Master of Science in Cyber Security

Albin Shaju
Student ID: 23215496

School of Computing
National College of Ireland

Supervisor: Prof. Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Albin Shaju

Student ID: 23215496

Programme: Master of Science in Cyber Security **Year:** 2024

Module: MSc Research Project

Supervisor: Prof. Jawad Salahuddin

Submission Due Date: Thursday, 12 December 2024

Project Title: Exploring Machine Learning Approaches for Robust Anomaly Detection and Responsive Security in IoT Frameworks

Word Count: 7759 **Page Count** 20

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: Albin Shaju

Date: 12 December 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):	

Exploring Machine Learning Approaches for Robust Anomaly Detection and Responsive Security in IoT Frameworks

Albin Shaju
23215496

Abstract

This paper evaluates machine learning (ML) and deep learning (DL) models for network traffic anomaly detection in IoT devices. Three models were tested: Support Vector Classifier (SVC), Convolutional Long Short Term Memory (ConvLSTM) and Extreme Gradient Boosting (XGBoost) Model. The motivation behind this project is to analyse, examine and evaluate machine learning and deep learning models and their effectiveness with respect to discovering and mitigating network anomalies in IoT frameworks. The accuracy for the XGBoost model is higher than the SVC and ConvLSTM models, with an accuracy of 99.98% as compared to 92.80% and 96.18% respectively. After each attack, the Inference system logs the result, sends email notifications when it detects an attack, and skips blocked IP addresses. Results demonstrate that XGBoost model has the ability to identify the DDoS attacks better, which would be promising for future network security applications in real environments.

1 Introduction

As the networks become more complex, DDoS attacks are becoming more common and damaging and hence detecting such attacks to secure and stabilize networks has become an important task. DDoS attacks, whereby an attacker floods a server with large amounts of traffic to interrupt normal traffic, are becoming more and more complex. Yet, most of the present-day solutions resist in precisely detecting and rectifying such attacks, with specific emphasis on attacks which behave identical to legitimate traffic, or use newer, less understood tactics to launch their attacks. During recent years, the hope has been that newer and more effective detection methods can be developed, especially by the use of machine learning (ML), and deep learning (DL) methods. Although it still has challenges, the detection accuracy is improved for complex attack patterns and false positives are reduced.

The research aims to address the question: **What are the common vulnerabilities found in IoT devices, and what strategies can effectively mitigate these vulnerabilities?** It particularly looks at the use of deep learning and machine learning models for detecting and responding to anomalies in IoT network, specifically protection against Distributed Denial of Service (DDoS) attacks. The goal of this study is to develop machine learning and deep learning models using real world network traffic data, measure performance using accuracy, precision, recall and F1 score, evaluate the effect of feature engineering on detection effectiveness, and provide an inference system by choosing the most effective model. This work also adds to the scientific literature on the applicability of machine learning models and deep learning methods to detect DDoS attacks in IoT network traffic. It also aims to classify attacks more accurately, generate less false positives, and generate alerts faster to network administrators.

Based on its comprehensive listing of 33 attack types and 105 devices, the CICIoT Dataset 2023 which includes IoT network traffic was chosen for the project. The capacity to offer real world scenarios along with a host of its features enhanced its ability to train robust models to develop accurate, scalable, as well as practical solutions to detect and mitigate IoT specific vulnerabilities.

The report is structured as follows: Section 2 reviews related work of machine learning based intrusion detection system for DDoS attack detection; Section 3 explains the research methodology, including how dataset is prepared, how model is trained, and how features are selected; Section 4 explains the system design and architecture to detect the attack and manage the alert; Section 5 explains the implementation and inference; Section 6 explains the evaluation results with classification reports, confusion matrices and model performance metrics. Section 7 concludes our findings and suggests future work to be done.

2 Related Work

This section reviews many literatures on varied techniques which includes anomaly detection, phishing prevention, threat classification and the role of datasets and feature engineering to improve the system's accuracy and resilience to changing cyber threats.

2.1 Intrusion Detection Systems and IoT Security Frameworks

Abusitta et al., (2022) proposed a denoising autoencoder for extracting features resilient to noisy and heterogeneous environments and a deep learning-based anomaly detection framework for IoT systems. The study found9 that the framework is more effective than current models for detecting malicious and benign data. Machine learning algorithms (XGB, MLP, RF) for DoS/DDoS detection in IoT systems were studied by Alabdulatif et al., (2024) for which a prototype real time detection system based on XGB achieved a high accuracy of 99.93%. A review of ML/DL for abnormal behaviors and zero-day attacks detection is presented by Al-Garadi et al., (2020), while Asharf et al., (2020) review IoT vulnerabilities and the role of ML/DL in IoT networks security. Anthi et al., (2019) introduced a three-layer IDS for IoT networks with high performance in the detection of cyber-attacks in IoT devices. Thirdly, Elmasri et al., (2020) proposed three anomaly-based intrusion detection systems (KNN, enhanced KNN, LOF) and used the CICIDS2017 dataset to show the ability of detecting zero-day attacks and the ability of LOF to attain 90.5 percent accuracy.

2.2 Deep Learning and Hybrid Models for Threat Detection

Elzaghmouri et al., (2024) presented a hybrid deep learning model that includes CNN, BLSTM, GRU, and attention for multithreat detection with accuracy above 99.6% in the world of IoT security. A proposal for an IoT intrusion detection system based on deep learning and clustering techniques to enhance detection accuracy and reduce the data dimensionality by 62.45% is presented in Gheni & Al-Yaseen, (2024)Gudala et al., (2019) examines how AI can secure IoT networks and explore anomaly detection, threat classification and self-healing strategies for proactive vulnerability management. Haque et al., (2023) reviewed security measures of IoT networks recently, while more efficient datasets are required along with the use of novel techniques such as federated learning. Jony and Arnob (2024) used an LSTM based model for IoT intrusion detection with scalability and efficiency for larger deployment sizes with 98.75% accuracy. Besides, B Jony et al., (2024) also evaluated some ML algorithms to detect DDoS attacks on the IoT network and discovered that Decision tree and Random Forest outperform other algorithms in terms of accuracy and performance.

2.3 Phishing Detection and URL Classification

Alshingiti et al., (2023) proposed phishing website detection using deep learning based techniques (LSTM, CNN, and LSTM–CNN hybrid) and CNN gives the highest accuracy of 99.2%. LSTM and CNN were used to classify phishing email in Eckhardt & Bagui, (2021), LSTM outperformed CNN with 98.32% accuracy. Khan et al., (2020) used machine learning model for malicious URL detection using AdaBoost which outperforms blacklists and improved the accuracy of zero-day malicious URLs detection. Next, De La Torre Parra et al., (2020) suggested a cloud based distributed deep learning framework for phishing and Botnet attack detection on IoT devices using CNN and LSTM model with an accuracy of LSTM as 94.8%. In the studies we conduct, we find that deep learning can help improve phishing detection accuracy and efficiency on different platforms.

2.4 Datasets and Feature Engineering for IoT Security

To evaluate ML based intrusion detection systems in IoT and IIoT applications in centralized and federated learning modes, (Ferrag et al., 2022) developed the Edge-IIoTset dataset. The data in the dataset were of more than 10 types of IoT devices and 14 different attack types. Khorasgani et al., (2024) studied feature selection and data augmentation in deep learning-based anomaly detection, proposing a framework to optimize its configuration for particular use cases and improve detection performance on two major IoT datasets. Neto et al., (2023) introduced the CICIoT2023 dataset as a complete IoT attack dataset with 33 attack types on 105 devices to aid the development of more robust IoT security models. On similar lines, (Tseng et al., 2024) explored if the deep learning models, specifically the Transformers, could work well for IoT network intrusion detection task and they were able to achieve 99.40% accuracy with multi class classification using CIC IoT 2023 dataset. This thesis also contributes to the importance of having comprehensive datasets and engineering features in order to advance the frontiers of IoT security research.

2.5 Advanced Techniques and Future Directions in IoT Security

An IDS for IIoT networks is proposed by Shtayat et al., (2023) making use of an explainable ensemble deep learning that has more than 99% accuracy. They integrated SHAP and LIME into their system along with providing the increased decision transparency. A novel IoT attack detection framework using CNN and LSTM was developed by Sahu et al., (2021), and their framework performed with 96% accuracy on a recent dataset. In order to explore machine learning techniques for IDS on IoT vulnerabilities, Sharmila et al., (2024) used Decision Tree which had highest accuracy (99.85%). It also highlights the security challenges of IoT growth along with suggestions for solutions like better device security, data security and the creation of security standards. Future work across these studies includes further extending the scalability, interpretability, and efficiency of security models, incorporating privacy preserving techniques, and addressing the changing landscape of security attacks in IoT.

The table (Table 1) given below lists and compares some of the latest research papers that were used for this study and the research that use the same dataset.

Author	Year	Dataset	Technique	Accuracy	Objective	Strengths
Abusitta et al.	2022	Custom Dataset	Denoising Autoencoder, DL	99.93%	Real-time anomaly detection for IoT	High resilience to noise
Alabdulatif et al.	2024	Custom Dataset	XGBoost	99.93%	Real-time DDoS detection	High accuracy, real-world scenarios
Shtayat et al.	2023	RT-IoT2022	Ensemble DL, Explainability	99.85%	IIoT intrusion detection	Transparency with SHAP/LIME
Tseng et al.	2024	CICIoT2023	Transformer	99.40%	Multi-class intrusion detection	High dimensionality handling
Neto et al.	2023	CICIoT2023	Evaluation study	Not Reported	Dataset introduction	Comprehensive attack diversity
Jony & Arnob	2024	CICIoT2023	LSTM	98.75%	Scalability and efficiency in IoT IDS	Effective for large-scale deployment
Elzaghmouri et al.	2024	Custom Dataset	Hybrid Model (CNN + GRU + BLSTM)	>99.60%	IoT threat detection	Advanced hybrid model accuracy
Haque et al.	2023	CICIoT2023	Federated Learning	99.50%	Distributed attack detection	Reduced computational overhead

Table 1: Comparison of Important Research Papers

Several gaps in existing intrusion detection systems (IDS) for IoT networks are revealed by the reviewed studies. However, most frameworks consider detection accuracy over scalability and efficiency for resource constrained IoT environments. Real time detection, an important aspect for the dynamic IoT systems, has often been overlooked. Static datasets limit the adaptability to emerging threats, and generalization for various IoT architectures is an issue. Further, machine learning models’ explainability and interpretability are not sufficiently explored on account of which trust, and practical deployment are lacking.

3 Research Methodology

The research methodology adopted in this work follows a systematic, scientific process to evaluate the effectiveness of machine and deep learning models for detecting network anomalies in IoT environments using the IoT Network Intrusion Dataset (2023) from the Canadian Institute for Cybersecurity. This research methodology outlines the approach used for detecting network anomalies, justifies the selection of the methods and tools used and assesses the capability of these methods and tools to answer the research question. The focus of the study is to use machine learning (XGBoost, SVC) and deep learning (ConvLSTM) models to identify and even mitigate DDoS attacks on IoT networks.

3.1 Research Procedure

Data Collection: The dataset was downloaded from the data set link and contains network traffic data labeled with both benign and malicious traffic types, including Distributed Denial of Service (DDoS) attacks and other IoT-related intrusions.

The CICIoT dataset 2023 was selected, as it incorporated comprehensive coverage of IoT traffic from different devices with 33 different IoT attack types and benign traffic. The proposed models are trained and tested on this dataset, which provides real scenarios to make the models applicable and robust.

Data-set link: https://www.unb.ca/cic/datasets/iotdataset-2023.html

Dataset Characteristics: A mix of labeled benign and attack traffic data. Includes multi-class traffic classification categories like BenignTraffic, DDoS_ICMP_Flood, DDoS_TCP_Flood, and others. The dataset comprises features such as packet size, timestamp, and protocol usage, useful for classification tasks.

In **preprocessing**, we selected the relevant features for traffic classification based on domain knowledge, and we saved them as selected_features.pkl. To improve the model performance, irrelevant or highly correlated features were removed, and the remaining features were normalized using a Min Max Scaler to ensure compatibility with the model sensitivity to input data distribution.

Experimental Setup:

Support Vector Classifier (SVC): A traditional machine learning approach for classification.

Convolutional Long Short-Term Memory (ConvLSTM): Hybrid deep learning model, leveraging convolutional operations, as well as LSTM layers to take into account spatial as well as temporal dependencies of the data.

XGBoost: An advanced algorithm that belongs to the gradient boosting group that is well known by its efficiency and also accuracy for classification tasks. The boosters incorporate boosting techniques, optimized tree-based learning and enable fast training, especially in handling large, high-dimensional, data in the presence of complex patterns and interactions.

Together, the models comprise a wide spectrum of capabilities from classical machine learning to hybrid deep learning.

3.2 Techniques Applied

Data Splitting: To test the performance of the models the dataset was divided into training (80%) validation (10%) and testing (10%) subsets. The class distribution was kept balanced through stratified splitting.

Model Training: The models, XGBoost, ConvLSTM, and SVC, were trained from the CICIoT Dataset 2023. We scaled to enhance performance by performing feature selection and used a MinMaxScaler. The model that achieved the highest accuracy was XGBoost which handles traffic patterns quite well. To optimize XGBoost model, I used RandomizedSearchCV to find the best hyperparameters (n_estimators, learning_rate and max_depth) within a given parameter grid. Then, I performed a search over 10 random combinations of CV parameters (5-fold) using accuracy as a scoring metric. After determining the optimal parameters, the model was retrained using training data and tested on the test set for robustness of the performance. ConvLSTM Model was trained using 10 epochs and 32 batch size. We combined the Conv1D layers that works on capturing spatial features with Bidirectional LSTM layers for temporal dependencies. With a multi class classification problem, the model was compiled with the Adam optimizer and a categorical crossentropy loss function. In order to reduce the learning rate as the validation accuracy plateau's, a learning rate scheduler (ReduceLROnPlateau) which adjusts learning rate to resume convergence was implemented. The SVC model was trained with a linear kernel with regularization parameter, $C = 0.01$, specifically on feature scaled input data to classify the network traffic effectively. All models were evaluated with precision, recall, F1score, and confusion matrices in both models.

3.3 Data Analysis Procedure

Raw Data Compilation:

A thorough cleaning and normalizing process was done upon input traffic data from CICIoT Dataset 2023 using MinMaxScaler to ensure data consistency and compatibility on machine learning and deep learning models. We selected features based on their correlation to the target labels and used for future use. As a ConvLSTM network, the model is able to analyze spatial and temporal dependencies in network traffic, therefore requiring input data to be transformed into sequences. To reduce the level of complexity for structured learning tasks, packet level data were aggregated into flow level statistics, leading to simplified representation in case of SVC and XGBoost models. The dataset was stratified and split into three subsets: training, validation and test, balancing the class distributions across splits.

Evaluation Metrics:

The following metrics were used to assess all models:

- Accuracy: Percentage of correctly classified samples (all classes).
- Precision: Crucial for understanding how reliable our model is in recognizing specific attack types; model's ability to predict true positive (attack) rates versus all predicted positives (positives, attack or non-attack).
- Recall: It's the ratio of the number of actual positives which are actually identified correctly to the number of total positives.
- F1-Score: A simple mean of precision and recall that can be used as a good measure for minority class detection.
- Confusion Matrix: Was used to identify misclassifications and errors in the attack and benign traffic classes.

Accuracy and Loss Plots:

Plots of accuracy and loss of the ConvLSTM model showed how it learns and generalizes over epochs. Initially, looking at validation accuracy, while the loss keeps decreasing steadily with the number of epochs, improvements in the validation accuracy have plateaued by far. These plots demonstrated ConvLSTM could learn complex patterns in sequential data more successfully than SVC in terms of recall and overall detection robustness.

The accuracy plot of XGBoost represents how good it can detect the patterns in a network traffic.

The insights gained by these evaluations shows how well the models detect anomalies and classify network traffic in IoT environments.

3.4 Equipment and Tools

Hardware: Training and testing of the models were done on a system consisting of an AMD Ryzen 7 processor with 24 GB RAM, sufficient to handle big data and train deep learning models of increasing complexity.

Software: For development of model and data preprocessing, Python with support of TensorFlow, scikit-learn, pandas and NumPy libraries were used for the project. For development, visualization, and experimentation Jupyter Notebook was used. We used Yagmail for automated email notifications triggered by detected malicious traffic.

3.5 Contribution and Rationale for Methodology

Contribution of the Study:

This study develops an intrusion detection system (IDS) using three models: XGBoost, ConvLSTM and SVC were selected for their separate strengths and relevance to the research objectives. For efficiency of structured data processing along with robust performance for imbalanced datasets, XGBoost was selected. Based on the papers reviewed as part of the literature review, XGBoost has been widely validated for superior accuracy and scalability. For successfully detecting the complex DDoS attack behaviours in network traffic, ConvLSTM is chosen because it is capable of capturing both spatial and temporal patterns. It uses hybrid architecture which consists of convolutional layer, and the LSTM layer to analyse sequential data comprehensively. To serve as a baseline model against traditional ml model, SVC (even though simpler) is also included. And use of these models is supported in literature on similar works, and they work well for performing anomaly detection tasks. The combination of the research question and computational approaches facilitates a comprehensive survey of various computational approaches, providing comprehensive insights into the question asked. The XGBoost model performs well on structured data and therefore it achieves the highest validation accuracy among the models. In addition, the system has been integrated with real time inference, automated IP blocking and email notifications to be a practical, efficient and scalable cybersecurity solution.

Significance and Usefulness:

This work proposes a system that improves network security by using advanced machine learning and deep learning models for DDoS attacks detection. With its real time response features—CIDR based automatic anomaly detection and IP blocking—along with distinguishing among different DDoS attack types, it easily scales to enterprise level deployments. Email notification is also integrated, to give timely alerts for threat mitigation to the network administrators.

Dataset Selection:

The project uses the CICIoT Dataset 2023 which includes benign traffic and various DDoS (SYN Flood, ICMP Flood, and UDP Flood) attack traffic types. Furthermore, we demonstrate the validation of the proposed framework by using a real world dataset, which guaranties that the models trained and evaluated in the real scenario and contribute to the system be robust to use in the dynamic IoT scenarios. Due to the comprehensive nature of the dataset, the IDS can generalize well and provides an optimal accuracy while detecting DDoS attacks and mitigation.

3.6 Explanation of Methodology

Final Results:

Three models including XGBoost, ConvLSTM, and SVC were compared and found that deep learning and modern machine learning techniques could be effective for anomaly detection. It turns out that XGBoost had the highest accuracy, ConvLSTM captured the temporal patterns quite well, and SVC was a decent baseline comparison. The results further point to a real-world application in IoT security.

Inference System:

Incoming network traffic is passed through pre trained models and predefined features, which further scale the features to the dimensions required by the network input. The changes are detected, logged and actioned with real time response (automated IP blocking, email alerts etc.) making this a practical and scalable cybersecurity solution.

3.7 Limitations

- Feature Selection: There are some concerns in the feature selection for the model as it relies on domain knowledge, which may cause some emergent patterns or features to be undetected.
- Data Representation: Data normalization ensures that our analytic results are consistent, but they may not compliment the data set in real world and its variations.
- Static Dataset: The CICIoT 2023 dataset is comprehensive, however, it is static and neglects the fact that attack patterns change and that IoT environments can mutate.
- Cross-Validation: RandomizedSearchCV optimized the hyperparameters, but we can also test the generalizability of the results obtained by testing on external data sets or real-world data.
- Model Training: ConvLSTM come at huge computational complexity that makes them unsuitable on resource constrained devices like IoT.

4 Design Specification

Intrusion detection system (IDS) is designed with the use of the advanced techniques and models to enable detection and classification of network anomalies in the IoT environment. Three models, XGBoost, ConvLSTM, and SVC, are used which are handpicked based on their characteristics. Gradient Boosted Decision Trees fare best on structured data and high dimensional features and XGBoost uses this. We optimize this model's hyperparameters using RandomizedSearchCV which enhances this model's accuracy. ConvLSTM, which is quite efficient with regard to sequential network traffic analysis, consists of integrating CNN layers which extract spatial patterns and LSTM layers for capturing temporal dependencies. The SVC model is a baseline as it is interpretable and implementable, but easier to compare to.

The system architecture for efficient detection and classification of network intrusions is shown in the figure (Figure 1). The system architecture involves three key components: preprocessing, inference and detection, and response. Data is cleaned, normalized and split into training, validation, and testing subset at preprocessing portion to keep balance of the evaluation. We selected features that have a correlation to target labels, which improves model performance. Real time traffic is processed by an inference system which classifies this traffic as either benign or malicious and logs anomalies. If suspicious traffic is detected, the system dynamically adds the malicious IP to a blacklist and automatically sends email notification to the administrators for action.

To be computationally efficient, it is recommended to run this model on a system with a processor above AMD Ryzen 3/Intel i5 10th gen and with at least 8 GB of RAM. This model was developed using Python with TensorFlow, scikit-learn and other supporting libraries. This IDS is a scalable and robust cybersecurity solution that consequently requires periodic retraining of models to adapt to continually evolving threats.

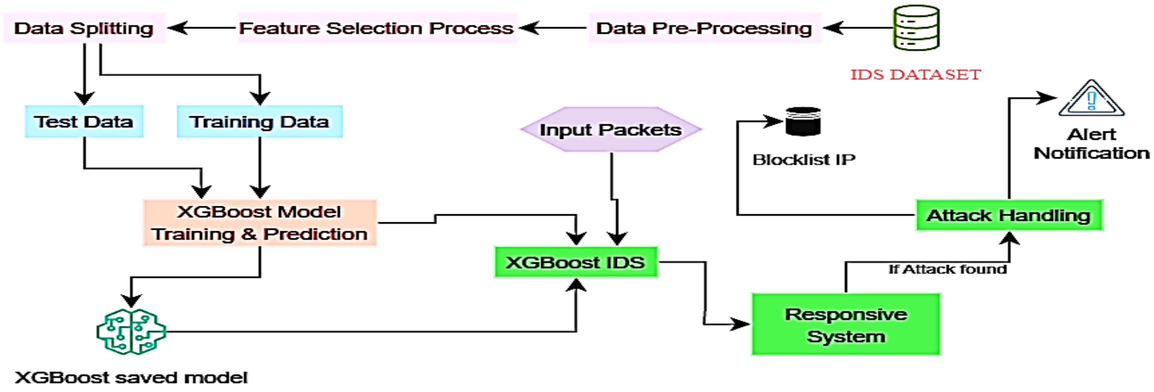


Figure 1: System Architecture

5 Implementation

In this Intrusion Detection System (IDS), we implement the process of preparing the dataset, model training and real time traffic anomaly detection. The process involved in this work includes data preprocessing, feature selection and model training to classify network traffic into categories as normal or attack.

5.1 Data Loading and Preprocessing

Loading Data:

In this step the dataset is loaded from two CSV files (df1 and df2) each having 47 features. Then we import these files into pandas DataFrame for analysis and preprocessing.

Exploring Data:

There are 238,687 records for df1 and 234,745 df2 records in the dataset. The figure (Figure 2) presents the bar plots of the distribution of labels (benign vs attack traffic) used to understand the structure of data and class balance.

Feature Selection and Correlation:

Correlating analysis makes the selection of top features related to the target label. To improve performance of the model, features with the strongest correlations are retained and those without are discarded.

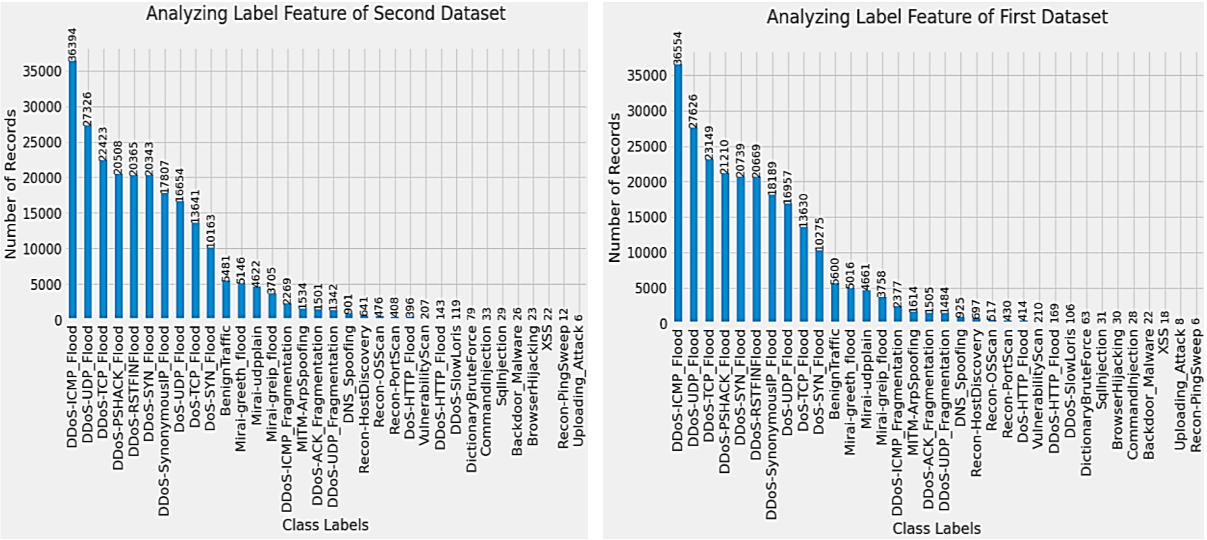


Figure 2: Visualization of class imbalance

Label Filtering:

Multiple DDoS attack types as well as benign traffic are included in the dataset. To ensure diversity and focus, the following labels were shortlisted for the project:

BenignTraffic, DDoS-ICMP_Flood, DDoS-UDP_Flood, DDoS-TCP_Flood, DDoS-PSHACK_Flood, DDoS-SYN_Flood, DDoS-RSTFINFlood and DDoS-SynonymousIP_Flood.

The class distribution is balanced up to 11,081 records for each label, which is effective for training purposes. Labels with fewer records are excluded, and this process is visualized in the figure (Figure 3).

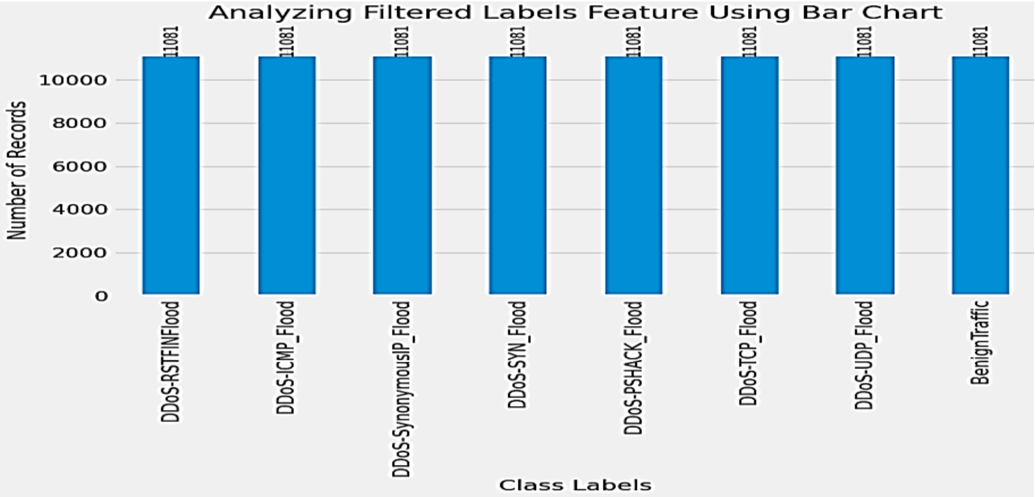


Figure 3: Analysing Filtered Labels Feature Using Bar Chart

Normalization:

We apply MinMaxScaler to scale feature values to [0, 1] to make sure all features provide similar contribution to model performance.

Data Splitting:

The dataset is divided into two subsets: training (80%) and testing (20%) subsets of 71,918 training samples and 17,730 testing samples respectively. On stratified splitting, each resulting set has the same class distribution.

Feature Importance:

A line chart is used to show feature importance based on correlation analysis where the most useful features for classification can be seen.

5.2 Model Training

Support Vector Classifier (SVC):

A linear kernel with regularization parameter of C = 0.01 was used to train the SVC model. We use X_train and y_train to fit the model and use X_test to make prediction. Accuracy, precision, recall, F1-score and confusion matrix are evaluated to measure the model performance across different attack types.

XGBoost Model:

RandomizedSearchCV is used to train the XGBoost model with hyperpararmeters like n_estimators and learning_rate and max_depth optimized. Structured data are well supported by the model, and its performance is evaluated using accuracy, precision, recall, F1 score, and a confusion matrix.

Convolutional Long Short-Term Memory (CLSTM):

The CLSTM model uses Convolutional Neural Networks (CNNs), to extract spatial features, and Bidirectional Long Short-Term Memory (LSTM) layers, to capture temporal dependencies in a sequence. The model is-trained for 10 epochs with batch size of 32. Accuracy, precision, recall, and F1-score are calculated as performance metrics to evaluate classification performance across different attack types, and a confusion matrix is generated.

Model Export:

After training, all models are saved for future use:

- We saved the cleaned, normalized dataset as normalized_data.csv. X_train.csv, X_test.csv, y_train.csv and y_test.csv consisting of the training and testing data are also saved as CSV files.
- SupportVectorClassifier model is saved as SupportVectorClassifier_model.pkl.
- XGBoost_model.pkl is the name used while saving the XGBoost model.
- ConvolutionalLongShortTermMemory_model.h5 is the saving of CLSTM model in HDF5 format.

5.3 Implementing IDS

In order to classify network traffic as normal or anomalous in real time, an Intrusion Detection System (IDS) is designed using the already trained XGBoost model for intrusion detection. This stage includes mechanisms for IP blocking and email notifications to manage a quick and sensitive response to the work of identifying and mitigating malicious traffic. The system guarantees that the classification is correct without losing consistency among the preprocessing we made during the training and the transformations applied to the new traffic data.

Library Imports

To implement the functionality of the inference process, several Python libraries are used. Pandas and numpy are used for good data manipulation and numerical operation on incoming network traffic data. We use pickle library to load the pre trained model, selected features and normalization scalers. To send email notifications when malicious traffic is detected the yagmail library is used. Collectively these libraries provide the tools necessary to construct a robust and efficient IDS.

Loading Pre-Trained Model and Resources

The pre trained XGBoost model saved as XGBoost_model.pkl is loaded using pickle library. During training, this model is trained to classify traffic into various traffic categories (e.g. different types of DDoS traffic and benign traffic). In addition to the model, selected_features.pkl and scaler.pkl are loaded.

Selected Features: Subset of features selected in the feature selection is stored in the selected_features.pkl file. This includes very important features to ensure that the data to be inputted regarding the objective function provides information in a fashion that respects the model requirements.

Scaler: The MinMaxScaler used for feature value normalization during preprocessing is saved as scaler.pkl file. This makes sure that new traffic data gets transformed the same way as old, so that we preserve our model’s predictions.

The system also provides the class labels related to the different traffic types used, e.g., benign traffic, DDoS_ICMP_Flood and DDoS_SYN_Flood DDoS attack types. These labels help the model to predict and take appropriate action depending upon the prediction.

Email Notification Function

An automated email notification function is incorporated in the system which notifies the administrators when malicious traffic is detected. This function uses the yagmail library to connect to the sender’s email account via an app password, and then generates a detailed alert. Details like IP address, predicted attack type, accuracy score are contained in the email. This helps administrators see where they may have problems to head off potential threats. The errors are logged for debugging, if the email fails to transmit. The role of email notification feature is to provide real time alerts for more practical use of intrusion detection system in real life.

Blocklist Verification

The system checks if associated IP address is already in blocklist before processing traffic. This will verify previously identified malicious IPs, so they are not repeated routed through, saving resource and not repeating computation. The first IP address is extracted from the ip_address column in the input data if it exists and otherwise it is set to "Unknown." The system checks the IP against entries in predictions_log.csv, which is the recording of blocked IPs. If a match was found prediction process was skipped and a message will be displayed saying that the IP is already blocked.

Prediction Function

The IDS is centered on the prediction function. Incoming traffic is processed, blocklist checks is handled, features are scaled, and predictions are made using the pre trained XGBoost model. First, IP address is extracted, then the system checks if it hasn't already been included into the blocklist. The function skips the whole flow if the IP is blocked.

The traffic data is normalized for unblocked IPs using the pre-loaded MinMaxScaler to match the preprocessing done during training. And then the normalized data is passed to the XGBoost model, which makes a prediction about the traffic type. Predicted label and its associated confidence score is model's output. They are then mapped to predefine class labels like BenignTraffic or different kinds of DDoS attack types.

However, if the traffic is benign, the system takes no action, saving computational resources. Nevertheless, if the traffic is assessed as malicious, the system puts the predictions in predictions_log.csv with the relevant data, i.e. the IP address, predicted label and confidence score. The log is a record of the threats detected, and it supports analysis further. It also updates the blocklist with the malicious IP found and also triggers the email notification function to notify administrators.

Code Flow

The IDS has been designed such that network traffic is handled efficiently and accurately. The incoming traffic data is then first used with the pre-saved scaler and feature set to preprocess it. The system checks the IP to see if they are on the blocklist. We skip the process for saving resources if IP is blocked. For unblocked IPs, traffic type is predicted by the XGBoost model, benign or malicious traffic is labelled. The system handles the results appropriately: Benign traffic is ignored, and malicious traffic will cause logging, blocking and email notifications. This would theoretically be a structured way to optimize resource utilization and high response rate to threats. When using this approach for decision making, the system does not spend resources on non-critical tasks, continuing as normal to detect and reduce threats.

Real-Time Traffic Handling

The inference system also processes network traffic in real time, allowing for a quick reaction to a possible threat. The system identifies and stops malicious action with the help of fusion of the pre trained XGBoost model and dynamic blocklist update as well as email alert. Real-time preprocessing is used to guarantee that all data is transformed in the same way and blocklist verification ensures malicious IPs are not processed again. These features together enhance system efficiency and reliability.

With its XGBoost pre trained model and automated real time monitoring and notifications, this is a complete intrusion detection package. We found that this system can defend IoT networks in a robust and practical way through the ability to respond to threats quickly and handle network traffic efficiently.

6 Evaluation

A detailed evaluation of the performance of the three models used in this study is provided in this section. The purpose of this evaluation is to test the ability of these models to detect DDoS attacks and differentiate between benign and attack traffic.

6.1 Evaluation of SVC

The validation accuracy of SVC model was 92.80%, making it a good predictor model for network traffic. There were some misclassifications though; mainly in separating some DDoS attack types from benign traffic. These limitations need improvements.

Classification Report

Below is a summary of SVC model classification report. The model achieved an overall accuracy of 92.80%, with the following key metrics:

Macro Average: The model achieved a Precision of (0.95) Recall of (0.93) and F1-score of (0.92).

Weighted Average: 0.95 Precision, 0.93 Recall and 0.92 F1-score.

The model achieved F1 scores of 0.99, 1.00 for BenignTraffic traffic type and some other attack types. However, it struggled with certain attacks such as DDoS_SYN_Flood which had an F1- score of 0.78 and some minor misclassifications. DDoS_SynonymousIP_Flood had the F1-score of 0.62 which is not very good suggesting quite complex classification problem for this type. There were approximately 2216-2217 samples in each class, so the evaluation was fair across all categories.

Confusion Matrix

The confusion matrix for the SVC model is shown in figure (Figure 4). The model exhibited strong performance on most traffic classes:

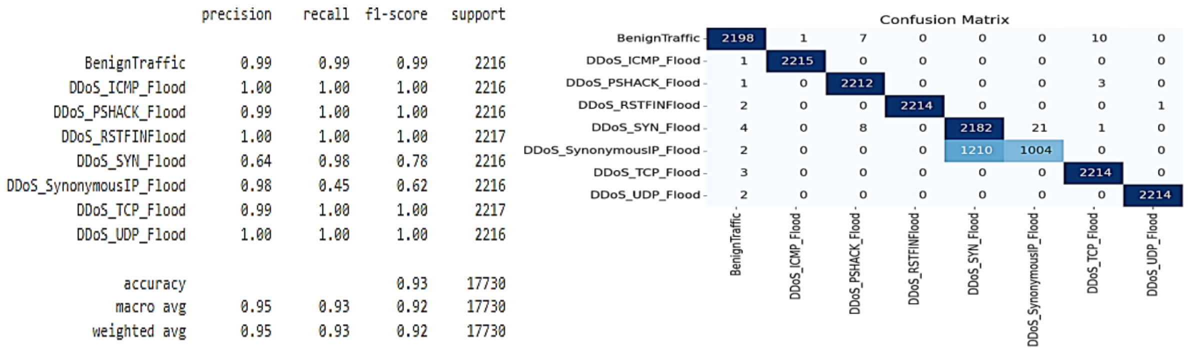
BenignTraffic: Correctly classified 2198 samples, with 10 false positives spread across other classes.

Mostly all other attacks had also similar result except DDoS_SynonymousIP_Flood having 1004 true positives, but 1210 false positives, showing significant overlap with other traffic classes. This confusion matrix highlights the SVC model's strong performance in classifying most traffic types. However, the high number of false positives for DDoS_SynonymousIP_Flood indicates room for improvement in differentiating this traffic type from others.

Insights from the Evaluation

Overall, the proposed SVC model performed well, attaining high precision, recall, and F1 scores for the vast majority of traffic types. As a tool for detecting common attack patterns, it can classify benign and specific DDoS attacks such as ICMP, PSHACK, RSTFIN, and UDP Flood with near perfect accuracy. Nevertheless, its constraints in dealing with

DDoS_SynonymousIP_Flood demand for more enhancement to differentiate, e.g., through integrating additional features or other modelling methods.



```
                         precision    recall  f1-score   support

           BenignTraffic      0.99      0.99      0.99      2216
          DDoS_ICMP_Flood      1.00      1.00      1.00      2216
        DDoS_PSHACK_Flood      0.99      1.00      1.00      2216
         DDoS_RSTFINFlood      1.00      1.00      1.00      2217
            DDoS_SYN_Flood      0.64      0.98      0.78      2216
DDoS_SynonymousIP_Flood      0.98      0.45      0.62      2216
            DDoS_TCP_Flood      0.99      1.00      1.00      2217
            DDoS_UDP_Flood      1.00      1.00      1.00      2216

                 accuracy                          0.93     17730
                macro avg      0.95      0.93      0.92     17730
             weighted avg      0.95      0.93      0.92     17730
```

Confusion Matrix

	BenignTraffic	DDoS_ICMP_Flood	DDoS_PSHACK_Flood	DDoS_RSTFINFlood	DDoS_SYN_Flood	DDoS_SynonymousIP_Flood	DDoS_TCP_Flood	DDoS_UDP_Flood
BenignTraffic	2198	1	7	0	0	0	10	0
DDoS_ICMP_Flood	1	2215	0	0	0	0	0	0
DDoS_PSHACK_Flood	1	0	2212	0	0	0	3	0
DDoS_RSTFINFlood	2	0	0	2214	0	0	0	1
DDoS_SYN_Flood	4	0	8	0	2182	21	1	0
DDoS_SynonymousIP_Flood	2	0	0	0	1210	1004	0	0
DDoS_TCP_Flood	3	0	0	0	0	0	2214	0
DDoS_UDP_Flood	2	0	0	0	0	0	0	2214

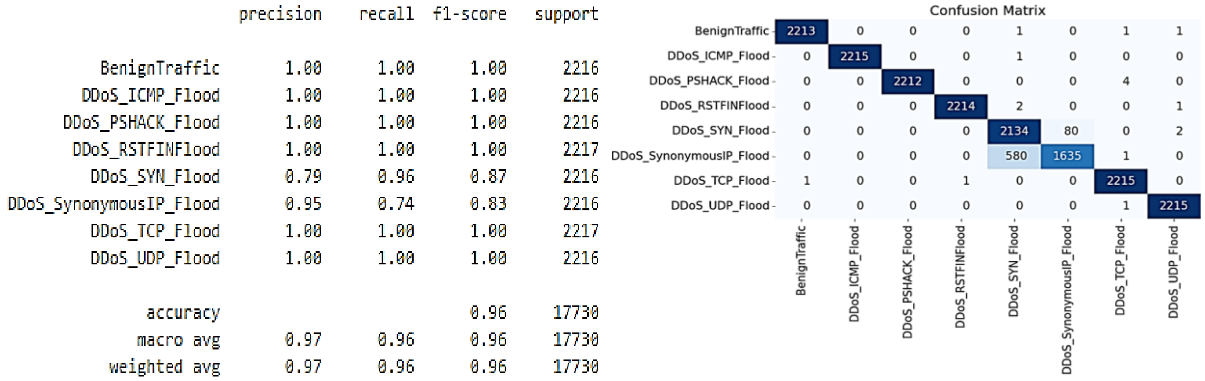
Figure 4: Classification report and confusion matrix for SVC

6.2 Evaluation of CLSTM

ConvLSTM model demonstrated strong performance in classifying network traffic, with validation accuracy of 96.18%, by capturing both spatial and temporal patterns well. Across most traffic classes, it showed excellent precision, recall, and F1-scores, however, we did encounter some challenges regarding the identification of specific attack types.

Classification Report

For Traffic types like BenignTraffic, DDoS_ICMP_flood, DDoS_TCP_Flood and DDoStraggleDdoSTCP_flood the model achieved 1.00 of precision, recall, and F1 score. These results show comparable performance with benign traffic and simple attack. The DDoS_SYN_Flood class, however, had an F1-score of 0.96, with a small loss in precision and recall as the cost of 80 false positives (FP). The model struggled more with DDoS_synonymousIP_flood, with an F1 score of only 0.73 and precision of 0.74, largely because the features overlapped, yielding 580 TP and 1635 FP. In general, the macro and weighted F1-scores are averaged at 0.96, showing that generalization is strong.



```
                         precision    recall  f1-score   support

           BenignTraffic       1.00      1.00      1.00      2216
          DDoS_ICMP_Flood       1.00      1.00      1.00      2216
        DDoS_PSHACK_Flood       1.00      1.00      1.00      2216
         DDoS_RSTFINFlood       1.00      1.00      1.00      2217
            DDoS_SYN_Flood       0.79      0.96      0.87      2216
DDoS_SynonymousIP_Flood       0.95      0.74      0.83      2216
            DDoS_TCP_Flood       1.00      1.00      1.00      2217
            DDoS_UDP_Flood       1.00      1.00      1.00      2216

                 accuracy                           0.96     17730
                macro avg       0.97      0.96      0.96     17730
             weighted avg       0.97      0.96      0.96     17730
```

Confusion Matrix

	BenignTraffic	DDoS_ICMP_Flood	DDoS_PSHACK_Flood	DDoS_RSTFINFlood	DDoS_SYN_Flood	DDoS_SynonymousIP_Flood	DDoS_TCP_Flood	DDoS_UDP_Flood
BenignTraffic	2213	0	0	0	1	0	1	1
DDoS_ICMP_Flood	0	2215	0	0	1	0	0	0
DDoS_PSHACK_Flood	0	0	2212	0	0	0	4	0
DDoS_RSTFINFlood	0	0	0	2214	2	0	0	1
DDoS_SYN_Flood	0	0	0	0	2134	80	0	2
DDoS_SynonymousIP_Flood	0	0	0	0	580	1635	1	0
DDoS_TCP_Flood	1	0	0	1	0	0	2215	0
DDoS_UDP_Flood	0	0	0	0	0	0	1	2215

Figure 5: Classification report and confusion matrix for CLSTM

Confusion Matrix

The confusion matrix shown in the figure (Figure 5) points to near perfect classification for most traffic types among minimum number of misclassifications. Model results were perfect or near perfect for benign traffic, the common DDoS attacks ICMP, RSTFINFlood, and TCP Flood; struggled with DDoS SynonymousIP Flood, where a large number of false positives were observed.

Accuracy and Loss Plots

From the accuracy plot shown in figure (Figure 6) we can see that the training accuracy began at 0.91 and the validation accuracy at 0.89, and by epoch 5 had increased to 0.96 and 0.92 respectively, where they plateau. The training loss plot shown in the figure (Figure 6) shows the training loss steadily dropping and stabilizing at 0.05 after epoch 2, and the validation loss was fluctuating early but converged to 0.10 by epoch 6. These indicators of trends are good learning and generalizations.

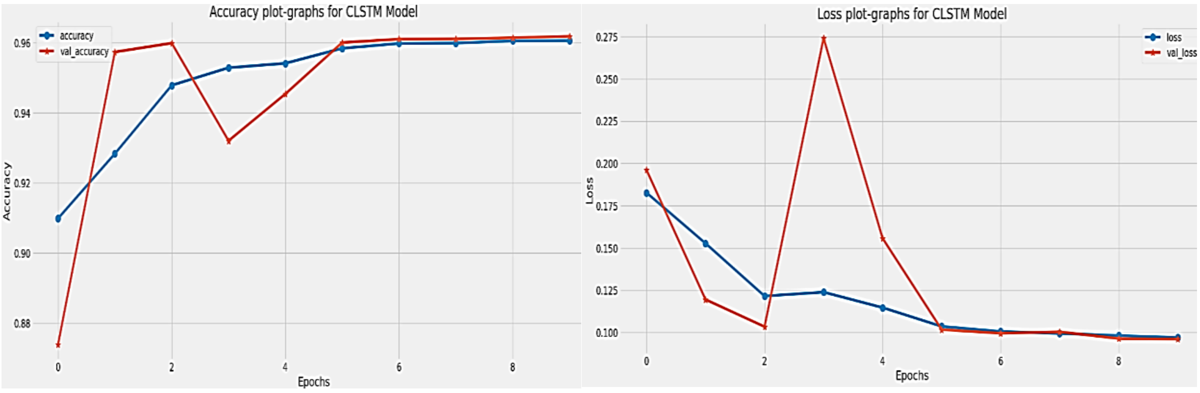


Figure 6: Accuracy Plot & Loss Plot

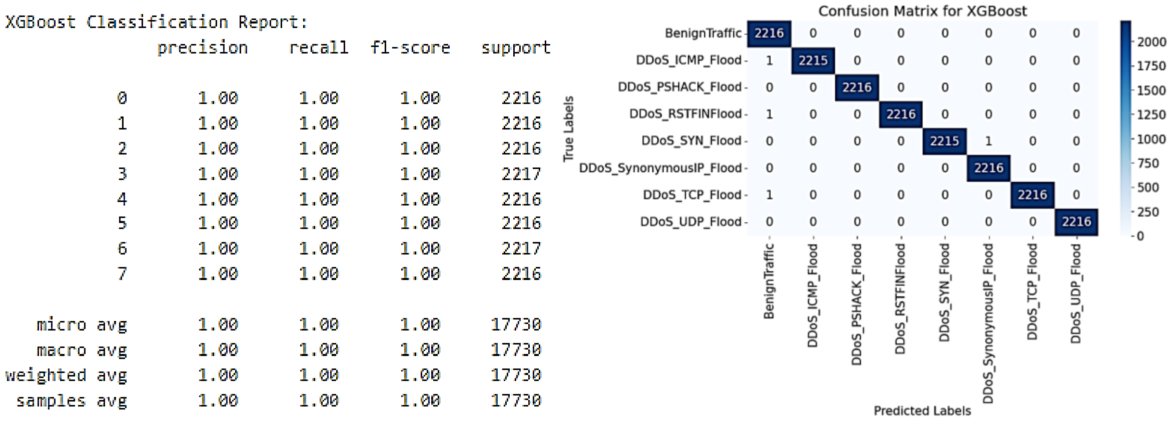
Overall, the ConvLSTM model achieved favourable performance both in terms of accuracy and generalization across majority of traffic types with room to improve in identifying complicated attack patterns such as DDoS_SynonymousIP_Flood.

6.3 Evaluation of XGBoost

Classification of network traffic gave out great results from the XGBoost model with a validation accuracy of 99.98%. This ability to correctly classify all traffic types with minimal misclassifications demonstrate its robustness for intrusion detection tasks. This accurate model is perfect in dealing with structured data and capturing complex patterns in network traffic.

Classification Report

The classification report in the figure (figure 7) show that XGBoost model resulted with 1.00 precision, recall and F1-scores on all traffic classes. This suggests that the model could identify benign and attack traffic without error in a consistent manner. The results were further confirmed by both macro and weighted averages which showed the model was very strong on all metrics.



```
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2216
           1       1.00      1.00      1.00      2216
           2       1.00      1.00      1.00      2216
           3       1.00      1.00      1.00      2217
           4       1.00      1.00      1.00      2216
           5       1.00      1.00      1.00      2216
           6       1.00      1.00      1.00      2217
           7       1.00      1.00      1.00      2216

   micro avg       1.00      1.00      1.00     17730
   macro avg       1.00      1.00      1.00     17730
weighted avg       1.00      1.00      1.00     17730
 samples avg       1.00      1.00      1.00     17730
```

Confusion Matrix for XGBoost

True Labels \ Predicted Labels	BenignTraffic	DDoS_ICMP_Flood	DDoS_PSHACK_Flood	DDoS_RSTFINFlood	DDoS_SYN_Flood	DDoS_SynonymousIP_Flood	DDoS_TCP_Flood	DDoS_UDP_Flood
BenignTraffic	2216	0	0	0	0	0	0	0
DDoS_ICMP_Flood	1	2215	0	0	0	0	0	0
DDoS_PSHACK_Flood	0	0	2216	0	0	0	0	0
DDoS_RSTFINFlood	1	0	0	2216	0	0	0	0
DDoS_SYN_Flood	0	0	0	0	2215	1	0	0
DDoS_SynonymousIP_Flood	0	0	0	0	0	2216	0	0
DDoS_TCP_Flood	1	0	0	0	0	0	2216	0
DDoS_UDP_Flood	0	0	0	0	0	0	0	2216

Figure 7: Classification report and confusion matrix for XGBoost

Confusion Matrix

The classification using the confusion matrix (Figure 7) shows nearly perfect classification of all traffic classes. In particular we achieve precision and recall of 1.0 without false positives or negatives for BenignTraffic, DDoS_TCP_Flood, and DDoS_UDP_Flood. For a few classes, such as DDoS_ICMP_Flood or DDoS_SYN_Flood there were minimal misclassifications. The model is able to accurately differentiate between traffic types, and these minute errors are symptom of a nearly perfect model.

Accuracy Plot

As shown in figure (Figure 8), accuracy plot shows how performance is consistent across multiple test iterations, with mean accuracy close to 99.92%. The model's reliability and its capability of generalizing unseen data is confirmed by this stability across tests.

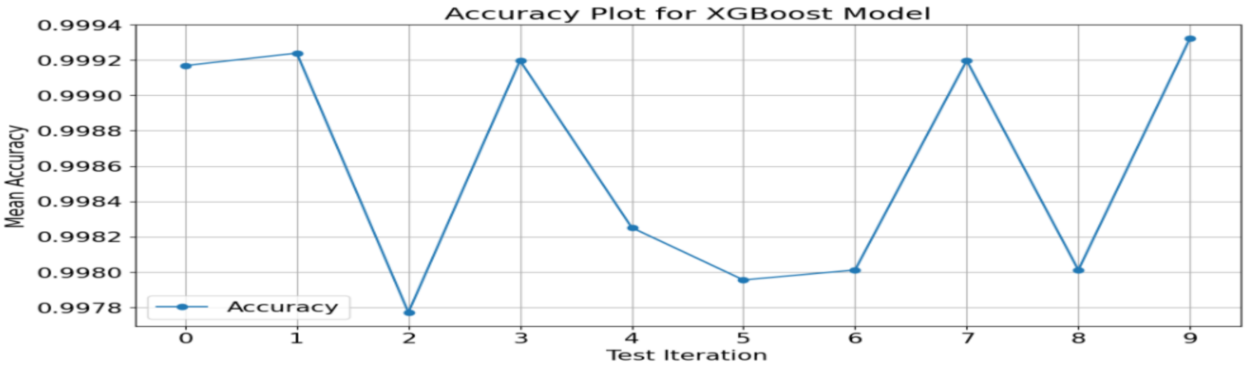


Figure 8: Accuracy Plot

Overall, the best classifier in this study is XGBoost, which approached perfect metrics for all traffic types. Due to its minimal errors and accuracy, this work is suitable for DDoS attack detection and network traffic classification in the real world.

6.4 Discussion

Findings from this study show the comparative performance of three models, SVC, ConvLSTM, and XGBoost, in detecting anomalies in IoT network traffic. All the models showed different degrees of success, with XGBoost as the best performing model with almost perfect accuracy, precision, recall and F1 scores. Although simple, the SVC model established a strong baseline but failed against more sophisticated attack patterns including DDoS_SYN_Flood and DDoS_SynonymousIP_Flood. However, ConvLSTM captured well spatial and temporal patterns at high level and performed well in most traffic types but found difficulty when there was an overlap of feature distribution in DDoS_SynonymousIP_Flood. The findings from this study is shown in the comparison table (Table 2).

Model	Accuracy	Precision	Recall	F1-Score	Strengths	Weaknesses
SVC	92.80%	0.95	0.93	0.92	Good baseline model, interpretable, handles small datasets well.	Struggles with DDoS_SYN_Flood and DDoS_SynonymousIP_Flood. Lower performance in complex attack detection.
ConvLSTM	96.18%	0.96	0.95	0.96	Captures both spatial and temporal patterns,	Computationally intensive, struggles with DDoS_SynonymousIP_Flood.

					excellent for sequential data.	
XGBoost	99.98%	1.00	1.00	1.00	Outstanding accuracy, handles structured data well, minimal misclassifications.	Requires hyperparameter tuning, higher memory usage for large datasets.

Table 2: Comparison table

But despite the promising results, the design has some limitations. Although the CICIoT Dataset 2023 is comprehensive, it is static and cannot fully emulate current evolving real-world attack scenarios. Better generalization might be realized by using real-time traffic datasets or federated learning. Furthermore, XGBoost had amazing results but it is solely built to work with structured data and in that case, it needs extra feature engineering to be deployed in a dynamic environment.

The results of this study also correspond with prior works suggesting that XGBoost is the better choice for structured data and that ConvLSTM is favourable in sequential data. Consequently, techniques such as federated learning or explainable AI, as mentioned in other works, would help to scale up and make it more transparent, which would lead to improved real world deployment.

7 Conclusion and Future Work

This study addressed the research question: What are the common vulnerabilities found in IoT devices, and what strategies can effectively mitigate these vulnerabilities? The focus was to evaluate and compare machine learning and deep learning model's capability to detect and mitigate network anomalies for IoT frameworks against vulnerabilities, such as Distributed Denial of Service (DDoS) attacks. The paper employs SVC, ConvLSTM, XGBoost models to classify different types of IoT traffic, enabling a real time intrusion detection system, and indicates the promise of ML/DL approaches in resolving the IoT security problems.

Moreover, key findings indicate that XGBoost is shown to be the best model as it has a near-perfect accuracy, and it is fit for predicting and resolving such network threats. Results show that ConvLSTM can capture temporal and spatial patterns yet have difficulty differentiating overlapping features in some attack types. While SVC was a reliable baseline, it was not able to handle complex attack patterns. Not only did this implemented system detect threats efficiently but it also responded dynamically with automated IP blocking and email notifications among other things, which prevented vulnerabilities in real time.

While these achievements exist, limitations are still there. The static dataset we used in this research might not effectively represent the dynamically emerged and further emerging IoT vulnerabilities, and the complexity of some models, e.g., ConvLSTM, could cause unfriendliness to resource constrained environments. Moreover, feature engineering depended heavily on domain knowledge and hence may be missing out on emergent patterns.

Future work should investigate how dynamic and real time data sets can make addressing the rapidly changing IoT security threats easier. By introducing explainable AI (XAI) techniques, we could improve the model transparency and trustworthiness making them suitable for more critical IoT systems. And since Federated learning can increase the privacy and scalability, it can also be explored. Finally, the system presents potential integration with cloud platforms or edge computing architectures for increased adaptability in for enterprise level and large scale IoT deployments to further enhance its commercial viability and practical application.

References

Abusitta, A., Silva de Carvalho, G. H., Abdel Wahab, O., Halabi, T., Fung, B. C. M., & Al Mamoori, S. (2022). Deep Learning-Enabled Anomaly Detection for IoT Systems. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4258930

Alabdulatif, A., Thilakarathne, N. N., & Aashiq, M. (2024). Machine Learning Enabled Novel Real-Time IoT Targeted DoS/DDoS Cyber Attack Detection System. *Computers, Materials and Continua*, *80*(3), 3655–3683. https://doi.org/10.32604/CMC.2024.054610

Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., & Guizani, M. (2020). A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys and Tutorials*, *22*(3), 1646–1685. https://doi.org/10.1109/COMST.2020.2988293

Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. *Electronics 2023, Vol. 12, Page 232*, *12*(1), 232. https://doi.org/10.3390/ELECTRONICS12010232

Anthi, E., Williams, L., Slowinska, M., Theodorakopoulos, G., & Burnap, P. (2019). A Supervised Intrusion Detection System for Smart Home IoT Devices. *IEEE Internet of Things Journal*, *6*(5), 9042–9053. https://doi.org/10.1109/JIOT.2019.2926365

Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions. *Electronics 2020, Vol. 9, Page 1177*, *9*(7), 1177. https://doi.org/10.3390/ELECTRONICS9071177

De La Torre Parra, G., Rad, P., Choo, K. K. R., & Beebe, N. (2020). Detecting Internet of Things attacks using distributed deep learning. *Journal of Network and Computer Applications*, *163*. https://doi.org/10.1016/J.JNCA.2020.102662

Eckhardt, R., & Bagui, S. S. (2021). Convolutional Neural Networks and Long Short Term Memory for Phishing Email Classification. *International Journal of Computer Science and Information Security*, *19*(5). https://doi.org/10.5281/ZENODO.4898109

Elmasri, T., Samir, N., Mashaly, M., & Atef, Y. (2020). Evaluation of CICIDS2017 with qualitative comparison of machine learning algorithm. *Proceedings - 2020 IEEE Cloud Summit, Cloud Summit 2020*, 46–51. https://doi.org/10.1109/IEEECLOUDSUMMIT48914.2020.00013

Elzaghmouri, B. M., Jbara, Y. H. F., Elaiwat, S., Innab, N., Osman, A. A. F., Ataelfadiel, M. A. M., Zawaideh, F. H., Alawneh, M. F., Al-Khateeb, A., & Abu-Zanona, M. (2024). A Novel Hybrid Architecture for Superior IoT Threat Detection through Real IoT Environments. *Computers, Materials & Continua*, *81*(2), 2299–2316. https://doi.org/10.32604/CMC.2024.054836

Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access*, *10*, 40281–40306. https://doi.org/10.1109/ACCESS.2022.3165809

Gheni, H. Q., & Al-Yaseen, W. L. (2024). Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset. *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, *9*, 100673. https://doi.org/10.1016/J.PRIME.2024.100673

Gudala, L., Shaik, M., Venkataramanan, S., & Sadhu, A. K. R. (2019). Leveraging Artificial Intelligence for Enhanced Threat Detection, Response, and Anomaly Identification in Resource-Constrained IoT Networks. *Distributed Learning and Broad Applications in Scientific Research*, *5*, 23–54. https://dlabi.org/index.php/journal/article/view/4

Haque, S., El-Moussa, F., Komninos, N., & Muttukrishnan, R. (2023). A Systematic Review of Data-Driven Attack Detection Trends in IoT. *Sensors 2023, Vol. 23, Page 7191*, *23*(16), 7191. https://doi.org/10.3390/S23167191

Jony, A. I., Jony, A. I., & Arnob, A. K. B. (2024). A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset. *Journal of Edge Computing*, *3*(1), 28–42. https://doi.org/10.55056/jec.648

Khan, F., Ahamed, J., Kadry, S., & Ramasamy, L. K. (2020). Detecting malicious URLs using binary classification through ada boost algorithm. *International Journal of Electrical and Computer Engineering*, *10*(1), 997–1005. https://doi.org/10.11591/IJECE.V10I1.PP997-1005

Khorasgani, A. T., Shirani, P., & Majumdar, S. (2024). An Empirical Study on Learning Models and Data Augmentation for IoT Anomaly Detection. *2024 IEEE Conference on Communications and Network Security, CNS 2024*. https://doi.org/10.1109/CNS62487.2024.10735681

Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors 2023, Vol. 23, Page 5941*, *23*(13), 5941. https://doi.org/10.3390/S23135941

Sahu, A. K., Sharma, S., Tanveer, M., & Raja, R. (2021). Internet of Things attack detection using hybrid Deep Learning Model. *Computer Communications*, *176*, 146–154. https://doi.org/10.1016/J.COMCOM.2021.05.024

Sharmila, B. S., Nandini, B. M., Kavitha, S. S., & Srivatsa, A. (2024). Performance Evaluation of Parametric and Non-Parametric Machine Learning Models using Statistical Analysis for RT-IoT2022 Dataset. *Journal of Scientific and Industrial Research*, *83*(8), 864–872. https://doi.org/10.56042/JSIR.V83I8.7437

Shtayat, M. M., Hasan, M. K., Sulaiman, R., Islam, S., & Khan, A. U. R. (2023). An Explainable Ensemble Deep Learning Approach for Intrusion Detection in Industrial Internet of Things. *IEEE Access*, *11*, 115047–115061. https://doi.org/10.1109/ACCESS.2023.3323573

Tseng, S. M., Wang, Y. Q., & Wang, Y. C. (2024). Multi-Class Intrusion Detection Based on Transformer for IoT Networks Using CIC-IoT-2023 Dataset. *Future Internet 2024, Vol. 16, Page 284*, *16*(8), 284. https://doi.org/10.3390/FI16080284