National College of Ireland

DDoS Defence in IoMT: A Hybrid CNN-LSTM approach for SNORT based Intrusion Detection

MSc Research Project
MSc Cybersecurity

Misha Rose Kambakaran Mathew
Student ID: 22159851

School of Computing
National College of Ireland

Supervisor: JOEL ALEBURU

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	MISHA ROSE KAMABAKARAN MATHEW
Student ID:	22159851
Programme:	MSc Cybersecurity **Year:** 2024
Module:	MSc Cybersecurity Research
Supervisor:	Joel Aleburu
Submission Due Date:	16/09/2024
Project Title:	DDoS Defence in IoMT: A Hybrid CNN-LSTM approach for SNORT based Intrusion Detection

Word Count: **8048** **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: MISHA ROSE KAMABAKARAN MATHEW

Date: 16/09/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

DDoS Defence in IoMT: A Hybrid CNN-LSTM approach for SNORT based Intrusion Detection

Misha Rose Kambakaran Mathew
22159851

Abstract

The advancement of IoT has expanded its application in various sectors including health sector, which leads to rise of IoMT. IoMT also transfer large amount of data between devices, therefore it is highly susceptible to cyber-attacks. This research mainly focuses on mitigation of Distributed Denial of Service (DDoS) attacks on IoMT devices by developing a hybrid Convolution (CNN) and Long Short-Term Memory (LSTM) model integrating with SNORT Network Intrusion Detection System (NIDS). The CNN-LSTM model utilize CNN for feature extraction and LSTM for pattern recognition, which provide best framework for detecting DDoS attacks in IoMT devices. The model was trained and evaluated using IoT-based ICU scenario, including both normal and malicious traffic data offering a broad view of network behaviour in healthcare. After deep learning hybrid CNN-LSTM model achieved 95% accuracy in identifying DDoS attacks with 95% of precision, 94.95% of Recall,94.99% of F1 score and 95% of Malicious score. A SNORT rule was formulated by integrating the deep learning predictions and it was tested by customized IoMT traffic.

Keywords: DDoS, IoMT, CNN, LSTM, NIDS.

1 Introduction

The Internet of Things (IoT) has emerged as a rapidly growing and essential type of network that connects to the internet through specific protocol and data sensing devices which enable the sharing of data, identification, tracking and monitoring (Saheed, Y. K., & Arowolo, M. O, 2021). Devices of daily use to industrial purposes are now being connected to each other for transfer the data and increased performance. As IoT usage grows exponentially, so does data generation, with projections indicating that by 2025, data output could reach approximately 73.1 Zettabytes (ZB) (S. Al-Sarawi, M. Anbar, R. Abdullah and A. B. Al Hawari). The Internet of Medical Things (IoMT) is a stream of Internet of Things (IoT), where number of healthcare gadgets like glucometer, fall detector, infusion pumps, intelligent pacemaker, pulse rate monitor are linked to each other to share the sensitive medical healthcare information which is further used by medical entities to provide medication and support (Saheed et al, 2021). The confidential data collected from these devices is stored in data centers, processed by gateway systems, and then transmitted to relevant end-users. The large amount of data handling makes it more vulnerable to cyberattacks. Numerous attack vectors are exploited to obtain control over these medical devices. (Saheed et al, 2021) While most of the IoT vulnerabilities apply to IoMTs, the most successful cyberattack is the Distributed Denial of Service (DDoS) attack **(**Rana Abubakar, Abdulaziz Aldegheishem, Muhammad Faran Majeed, Amjad Mehmood, Hafsa Maryam, Nabil Ali Alrajeh, Carsten Maple, and Muhammad Jawad. 2020). A DDoS

attack sends malicious traffic to a specific node or number of nodes through a bot system which has large number of computers that is controlled by the attacker **(**Rana et al., 2020). The machine that is compromised by the attacker is used to send large number of packets to the target for flooding the network and block services. The infected device used for executing DDoS attack can be IoT based system. These devices are connected in a particular way, so that they continuously transfer large number of packets without any interruption and shows like it is from legitimate user. During the attack different network packet services are used as source and mostly self-controlled trojans will be controlled by the attacker site. DDoS architectures are divided into two models such as proxy (Zombies) and Internet Relay-Chat (IRC) model **(**Rana et al., 2020). During the attack communication between zombie will be hidden since the channel will be encrypted, which makes it difficult to identify the DDoS architecture. Attackers use spoofing methods to hide the IP/MAC address by overriding the address with other random IP/MAC address. The identification of these attack is difficult unless the author publishes them, or it get accidentally identifies by the third party **(**Rana et al., 2020). Several studies are made to mitigate the arising DDoS attacks, with the improved ability of the attacker the demand for the research and design of a defense architecture for DDoS attack is increased. Conventional security measures cannot be considered for addressing these attacks. The conventional methods like encryption, authentication and trust-based system are difficult to apply and it does not guarantee the full security as well, therefore it demands a new defense mechanism. According to the studies the most effective methods for mitigating the DDoS attacks in IoMT are usage of Intrusion Detection/ Prevention systems like SNORT and Deep Learning Techniques. The identification of DDoS attacks can be done using predefined SNORT rules using K-mean clustering and MITRE ATT&CK framework and the SNORT inline mode with custom rule blocks these attacks (Lai, Y.-C., Yu, C.-L., Liao, M.-L., Lin, Y.-S., Chang, Y.-C., & Chen, J.-L. 2023). An There are several deep learning models that can be applied for Identification of DDoS attacks in IoMT, combining two or more models which are highly capable of DDoS detection overcome the limitations of individual approaches. Most of the previous research and studies focuses on general IoT security which demands a mechanism to identify and mitigate the DDoS attacks in Medical IoT devices. These limitations in previous research prompt the following research question.

Research Question:

"How can a hybrid CNN-LSTM model, integrated with SNORT NIDS, identify and mitigate DDoS attacks on Medical IoT devices?

Objectives:

- To develop a system using hybrid deep learning model for the identifying DDoS attacks in IoMT devices.
- To generate customized SNORT rules based on deep learning results and implementing these rules for detecting the DDoS attack.
- To evaluate the performance of the developed hybrid CCN-LSTM model using different metrices.

The objective of this paper is to develop a hybrid model combining deep learning methods like Convolution Neural Networks (CNN) and Long Short-Term Memory (LSTM) with the SNORT-NIDS for identifying and mitigating the DDoS attacks specifically on IoMT devices. Using a real time based synthetic dataset the hybrid CNN-LSTM model is trained to identify the DDoS attack patterns and trained model output is fed into the SNORT-NIDS. According to the studied pattern from the trained model, customized rules must be developed. Utilizing these rules SNORT-NIDS should throttle the suspicious traffic to mitigate the DDoS attacks. Evaluation of this hybrid model should be done by performance analysis on given dataset using different evaluation metrices like accuracy, precision, recall, F1 score, TPR, FPR and Area under ROC (AUC).

The remainder of this paper is structured as follows: Section 2 discusses related works on DDoS attack methodologies in IoT, Section 3 outlines the research methodology, Section 4 details design specifications, Section 5 describes the implementation of our proposed model, Section 6 presents the results from various evaluation metrics, and Section 7 concludes the paper and suggests future research directions

2 Related Work

A variety of research methods have been proposed and analyzed to efficiently detect and mitigate Distributed Denial Service (DDoS) attacks. One of the common approaches are Machine Learning and Deep Learning, many studies on the use of SNORT NIDS for identifying the DDoS have also made. In this section we thoroughly analyse the previous related works and understand how deep learning-based SNORT NIDS helps to identify and mitigate DDoS attacks in Internet of Medical Things IoMT, and how does it overcome the limitations of the defence mechanisms presented by different researchers.

Lai., *et al*. propose a defense mechanism that generates automatic SNORT firewall rules using machine learning and genetic algorithms. This system utilizes the Cowrie honeypot to capture attack packets targeting SSH connections. It employs the K-mean algorithm for unsupervised learning to preprocess the data, categorizes attack behaviors based on payload data features, and labels each group using TTPs from the MITRE ATT&CK framework. Perl Compatible Regular Expressions, crafted through genetic algorithms, generate SNORT rules that are later tested using Python scripts simulating attack packets. This approach achieves a 99.5% accuracy rate in categorizing attack behaviors and a packet coverage rate of 98% for SNORT rules filtering various attack patterns initiated by Cowrie. While this paper effectively categorizes and generate rules in controlled environment, the real-world application to different IoMT devices is limited and real-world traffic pattern maybe more complex and vulnerable beyond SSH attacks. The evaluation of the system is also done under controlled conditions, which lacks evaluation under dynamic and varied IoMT networks. And in the paper, it is also not clear whether this genetic algorithm-based rule generation is adaptable or not. This paper poses significant gaps of real-world application, adaptability to new vulnerabilities, and evaluation under dynamic conditions, which requires an advanced DDoS defence mechanism.

Abdulrezzak, S., & Sabir, F. A. *et al.* (2023) uses SNORT's inline mode and iptables to enhance the intrusion detection system. The author proposes a method where DoS attacks and brute-force attacks are blocked by utilizing SNORT's inline mode and creates new SNORT rules which drop the packets that seems to be like the attack signature. To analyze the live traffic and improve the system performance three virtual Linux machines are installed, where first machine sends malicious network, in second machine SNORT is configured in inline mode with Netfilter Que (NFQ) Data Acquisition Library (DAQ) and third machine is the victim machine with Apache Server and My SQL. And author creates new SNORT rules to block and identify the attack packets, these rules try to match source IP addresses, ports, protocols and packet flags. The effectiveness was evaluated by SNORT's output and number of packets analysed and blocked. Around 99.64% of malicious traffic is blocked by this system where 149982 malicious packets were sent in rate of 94 packets/sec. This paper provides a framework for assessing network security by SNORT in the proposed method. Where Snort will be installed in a victim machine and attacker machine will be simulating malicious traffic and sending to it. The rule generation and the blocking mechanism of the SNORT can be evaluated in this virtual environment. Even though this paper focus on developing an advanced defence mechanism, it lacks certain features like evaluation of SNORT in real-world IoT environment with variable traffic patterns and device types. It is not clear whether SNORT' rule generation will adapt new attack signatures or techniques. And paper also demands a need for further investigation on the impact of SNORT's inline mode on network performance under high traffic. All these demands construction of an architecture which integrate SNORT with other security measures for obtaining vulnerabilities other than DoS and brute-force attacks.

Rana Abubakar, *et al*. suggest a DDoS protection method which utilize traffic analysis, protocol validation and machine learning. The system monitors the anomalies behavior of the traffic, filters the malicious traffic and reroute it away from the target. The combination techniques like traffic behaviour analysis, packet header validation, protocol validation and traffic matching dataset. Traffic patterns are classified based on decision tree algorithm. For anomaly detection the system implements individual packet threshold per protocol and uses Simple Network Management Protocol (SNMP) for communication between agents which enable distributed detection and information sharing. Each agent shares logs of detected malicious attack with central controller. This controller update signature database and sends SNMP alerts to all devices in the network with the updated Access Control Lists (ACLs) for filtering the traffic, Iptables firewall rules and routing information for the system is integrated with SNORT and iptables firewall rules for filtering malicious traffic. The implementation of the method mentioned in this paper, is highly complex and it is highly challenging to maintain it in large network. The network performance may get affected by the continuous traffic analysis and anomaly detection during the peak traffic time. The proposed system is highly depended on SNMP for communication between agents which requires an advanced network environment. This paper is focusing the mitigation of DDoS attacks in general, which demands a need of mitigation of DDoS attack in specific applications. This paper asks for an alternative communication protocol to replace or explore SNMP for improved operations in IoT environments.

Kauhsik, B., Nandanwar, H., & Katarya, R. (2023) *et al*. makes a study about the weakness in the existing techniques and suggest new the methods to overcome them. Deep Learning seem to be most effective approach for intrusion detection and prevention and this paper makes an analytic study by taking four research papers which accomplish IoT security using Deep Learning and Machine Learning techniques. Author also conducts review on latest publications and propose a combined measure to provide solution to the intrusion detection in IoT devices. According to this paper the integration of Machine Learning and Deep Learning allow to identify and mitigate malicious activity through analysis of network traffic patterns and anomaly detection. The data created by IDS can be utilized for training DL models which in turn increases the accuracy in identifying threats. The whole paper discusses benefits of using hybrid machine learning and deep learning techniques for feature selection on dataset with feature selection feature. But the model proposed here has limited adoption and demands a need for a hybrid model which integrate two model together for obtaining better results. There is a need for targeted research on different intrusion detection methods mainly for IoT devices. Paper also highlights the challenges and need of combining ML and DL models into existing IoT infrastructure with high efficiency and real-time responsive.

Idrissi, I., Azizi, M., & Moussaoui, O. *et al*. (2020) conducts a systematic literature review in IoT security and vulnerabilities by categorizing security threats according to Cisco IoT reference model architecture. This paper also reviews various existing works, mainly focusing Intrusion Detection System (IDS) based on Deep learning technique and these review and its findings will help in future research. According to the review certain results and findings are obtained, like deep learning method with wrapper based-feature extraction achieve 99.66% accuracy for binary classification and 99.77% for multiclass classification using UNSW-NB15 dataset.85.95% accuracy for binary classification and 81.33% for multiclass classification in multi-CNN fusion for Industrial IoT intrusion detections-based malicious traffic classification obtain nearly perfect accuracy with different dataset. 91.05% accuracy is obtained with KDD CUP 99 dataset in IoT data feature extraction and Intrusion detection using Deep migration learning. And the real-time attack detection with deep learning is reported almost 87.1% accuracy. These studies done by the author gives a clear picture about how application of deep learning model detects IoT security threats with high accuracy rate at different dataset. The main drawbacks of this system are limited evaluation of Deep Learning based IDS in real-world IoT with various device types and networks, limited validation of real-world performance and scalability of the model for IoT intrusion under variable conditions and generalization of the result also demand a detailed explanation. These drawbacks demand further research on how deep learning-based IDS can be validated for real-world scenarios and how can we do the practical implementation of this model.

Reddy, K. P., Kodati, S., Swetha, M., Parimala, M., & Velliangiri, S. *et al.*,2021 propose a hybrid neural network architecture for particularly detecting DDoS attacks using deep learning models. Mainly two deep Learning models named Gradient Boosting Decision Tree (GDBT) are used to extract spatial features of traffic flows and temporal features of traffic flows are classified using Convolution Neural Network (CNN). At first the raw network traffic packets are pre-processed to generate flow and based on the IP addresses distinguish between legitimate

and illegitimate traffic is done preserving packets of same flow. The output that is obtained from the two deep learning classifiers is merged using Add () function to create this hybrid model and Keras is utilized for this process where combined classification results predict DDoS attacks (1) or normal traffic (0) on the value of output tensor. From the research GBDT model achieve 93% and 91 % accuracy on training and testing dataset, CNN achieve 95% and 93% accuracy on training and testing and proposed hybrid model obtain 98% and 965 accuracies on training and testing. This paper gives a direction on how to build a hybrid model using two deep learning techniques and how to merge and integrate the results obtained from both techniques, where final prediction indicates the DDoS attack. After detailed analysis of this paper, it was clear that there are challenges in practical implementation and scalability of hybris-model. And the paper also poses a lack of comparative study between proposed hybrid model and other DDoS detection methods. The effectiveness of hybrid model to variable IoT network traffic and attack patterns also need further research. All these downsides ask for the performance metrices for evaluating effectiveness of hybrid models like GBDT-CNN in real time IoT DDoS detection.

Each paper reviewed identify how IoT security and intrusion detection can be done through various methods, but there are certain areas which remain unaddressed in this research. Mostly the research is done on controlled environment and urges a system which works in real-time environment and demand a performance evaluation of each system. Therefore, in this paper different mitigation techniques and solutions from the related works are thoroughly studied and tried to develop a hybrid model with high accuracy detection. Hybrid CNN-LSTM model training is done on the selected dataset to detect DDoS attacks and trained model output is fed into SNORT NIDS and customized rules are developed based on the results obtained. The performance of the model is evaluated using various evaluation metrices.

3 Research Methodology

This section provide knowledge about the research methodology that will be employed to develop a DDoS detection mechanism using CNN-LSTM hybrid model and SNORT NIDS. Methodology consists of details and steps that must done during the process like data collection, data preprocessing, development of hybrid deep learning model, integration of SNORT NIDS and analysis of mechanism. At the first stage of data collection the downloaded dataset was combined and sampled which was used for model analysis and evaluation. For model development different libraries were installed, and the collected data was pre-processed. The CNN-LSTM model was evaluated using different performance metrices during the model development. Later using this model predictions are made based on which SNORT rule was generated. Capability of SNORT in capturing IoMT traffic with the help of new generated SNORT rule is tested at final stage.

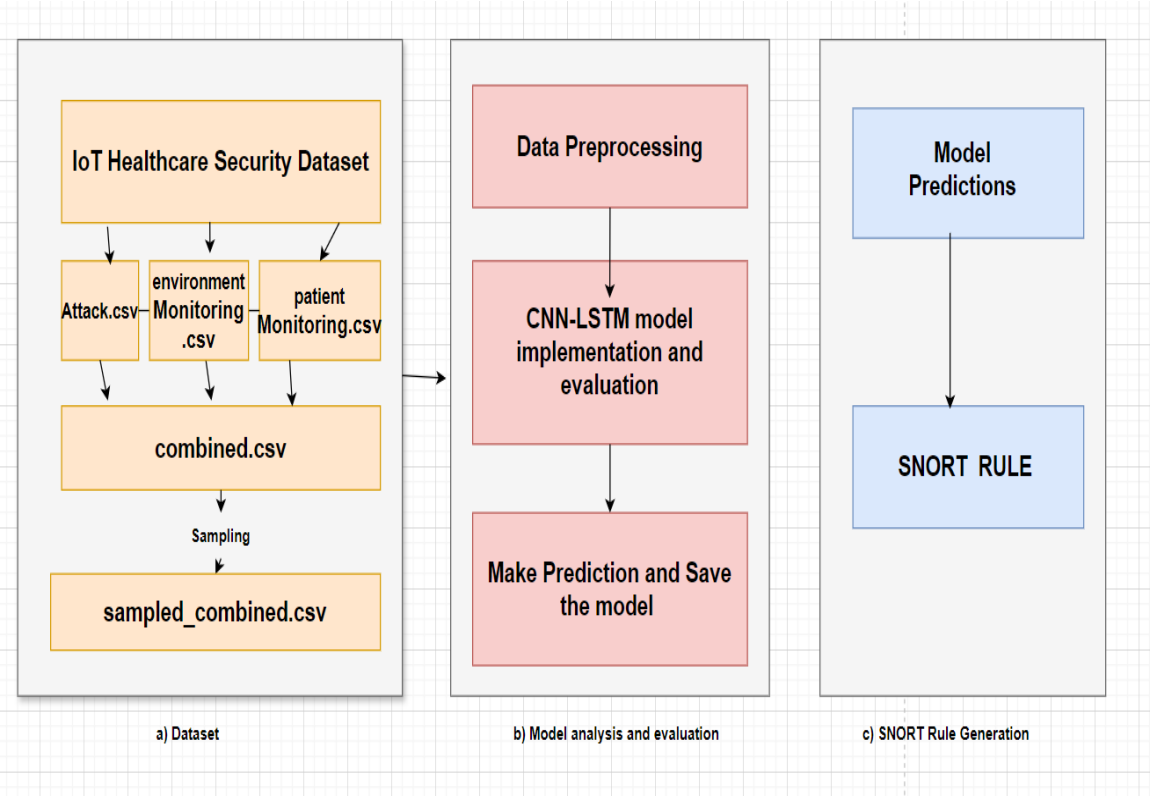

Figure 1: Flow chart of Methodology

3.1 Data Collection

Dataset was downloaded from Kaggle IoT Healthcare Security which is a synthetic data created for the research purpose using open-source tools. The use of synthetic data protects the patient privacy and ensure the compliance with data protection rules and regulations like HIPPA and GDPR. This dataset include data from different medical IoMT devices in which a use case of IoT-based ICU with capacity of 2 beds with each bed containing 9 patient monitoring devices and sensors and one control unit named Bedx-Control Unit. All devices have been created using IoT-Flock tool, which is an open-source tool for IoT traffic generation which support IoT application layer protocol like MQTT and CoAP. Dataset contains three sets of data, Attack.csv which consist of cyber-attacks traffic in healthcare, environmentMonitoring.csv containing environment sensors with normal traffic, patientMonitoring.csv containing ICU patient monitoring sensors with normal traffic. This dataset is downloaded and pre-processed for analysing certain features required for DDoS detection model are extracted (F. Hussain, 2023).

Features from this dataset helps to understand nature of network communication in IoMT devices which helps to identify patterns like DDoS attacks. 'frame.time_delta' represent time differences between consecutive frames indicate the sudden surge in traffic because during DDoS attack the time between packets decrease as attacker floods the network with large packets, 'frame.time_relative' indicate time from the beginning of capture helps to identify the time of unusual activity which allow to relate traffic hike with specific time and identify the attack window, 'ip.src' and 'ip.dst' represent source and destination IP addresses, these can identify the source of attack and targeted IoMT device, 'tcp.srcport' and 'tcp.dstport' is the source and destination ports in TCP communication. Because during

DDoS attack certain ports may be targeted more frequently which in turn identify the patterns of the attack and distinguish between normal and malicious traffic., 'tcp.flags' contain flags that control or identify various state of TCP connection these can identify SYN flood attack which is a type of DDoS attack. 'tcp.len' is the length of TCP segment, Abnormality in TCP segment length represent the attempts to flood the network or exploited vulnerabilities. 'tcp.ack' acknowledgment number in TCP communication, analysing this number helps in detecting anomalies in connection handshakes that represent attempts to disrupt normal communication. ‘tcp.connection.syn', 'tcp.connection.fin' and 'tcp.connection.rst' indicates establishment, termination and reset of TCP connections to identify unusual patters like high number of SYN or RST packets, these values suggest SYN floods. ‘tcp.window_size_value' represent window size used for flow control in TCP communication representing the attempts to manipulate flow control with flooding the network with data and degrading the performance of the IoMT devices, 'tcp.payload' is the actual data transmitted in TCP segment, by analysing this payload suspicious patterns designed for exploiting vulnerabilities related to DDoS attack can be detected. 'tcp.hdr_len' length of TCP header, anomalies in header length indicate attempts to exploit protocol weakness or inject malicious payloads that disrupt normal traffic. 'ip.proto' are the protocol used in IP packets which helps in identifying the sudden spike in rarely used protocol, during DDoS often less common protocols are used to avoid detection. 'ip.ttl’ represent time-to-live value of IP packets because unusual TTL value indicate traffic designed to avoid detection. All these features required for training CNN-LSTM model to identify the DDoS attacks in IoMT network traffic are extracted from this dataset (F. Hussain, 2023).

3.2 Data Preprocessing

The three sets of data in the dataset: Attack.csv, patientMonitoring.csv and environmemntMonitoring.csv were combined into a single data frame and saved as combined.csv. After combining the data, there were around 200,000 which made the deep learning more complex therefore from the combined data 3000 samples were selected and from each sample 3 classes where opted. For getting an evenly distributed classes, 3000 samples were selected, because there were 3 classes inside the combined dataset which include attack, environment monitoring and patient monitoring. After dividing the number of samples per number of classes, even distribution of classes was obtained. This type of sampling is important for training models that can generalize across different categories. During sampling, check whether all the labels environment Monitoring, patientMonitoring and attacks are presented in equal number and around 1000 samples each were selected from these datasets. The selected dataset contains both numerical and non-numerical features and these are pre-processed separately. StandardScaler can be used for preprocessing numerical features which standardize data to improve the performance and speed of deep learning models (Scikit-learn). After the scaling csr_matrix used for converting data into sparse matrix format which is efficient for large dataset. OneHotEncoder used for non-numerical features which converts categorical data into formal useful for deep learning, this data can also be stored in sparse matrix format for efficiency (Wojtek , 2023).

3.3 Hybrid CNN-LSTM Model Analysis and Selection

In this study, the strengths of both CNN and LSTM are combined for detecting DDoS traffic. The proposed model built with the help of IoT Healthcare security dataset, was first directed to CNN where spatial features are extracted and then to LSTM network for analyzing temporal patterns. The anomalies inside individual packets and sequence of packets such as repeated packet header or specific payloads which indicated DDoS attacks can be identified using CNN. Whereas LSTM capture long term patterns in sequential data which identify the attack through increase in traffic volume or change in packets intervals. The outputs of both CNN and LSTM layer are concatenated into one for combining both spatial and temporal features.

3.2.1 CNN

Convolution Neural Network is a of deep learning neural network architecture in computer vision (GeeksforGeeks, 2024). CNN is a version of artificial neural network (ANN) which is used for extracting features from grid like matrix dataset where there will be certain patterns inside the data (GeeksforGeeks, 2024). In the dataset each network session can be represented as grid where row corresponds to packets and column to features like tcp.len, tcp.flags etc. CNN consist of various layers like input layer, convolution layer, Max pooling layer, Dense layer and Output layer. Convolution layers apply filters to the input to extract features, within each samples this layer detect the local patterns. Maxpooling layer down sample the data and reduce the dimensionality of feature generated by convolution layer, For DDoS detection CNN utilizes each layer for different purposes, Input layer prepare the matrices correspond to a batch of packets with a dimension [batch _size, nume_packets, num_features]. Convolution layers detect the spatial patterns across the input which indicate the attacks. Pooling layers reduce the sensitivity to noise and network generalization. The fully connected layer inside CNN integrates the extracted features by convolution layer to predict whether the traffic is normal or malicious. The use of dense layers with sigmoid activation gives the probability of traffic being an attack.

3.2.2 LSTM

LSTM is a subset of RNN with memory cells and three gating mechanisms, it is effective in classifying the long data sequence of malicious network traffic. In the three-gating mechanism, first gate is the input gate, second gate is forgetting gate and last is output gate, these gates control the flow of information (GeeksforGeeks,2024). The important information is stored in the memory cell for a longer period using the gates and it can also be erased or replaced if required. All these features enable the model to detect the intrusion by using their capability of learning from high complex data, which is the reason behind choosing LSTM for deep learning model (GeeksforGeeks, 2024). In IoMT traffic data is processed as sequence of packet over time which allow the LSTM to analyse the feature changes. They identify long term dependencies and correlations in the data for detecting DDoS patterns. Ability of LSTM to maintain memory of previous inputs helps them to adapt to changes in traffic patterns. While CNN focus on extracting features from individual packets, LSTM analyse how these features evolve over time. The effectiveness of LSTM depends on sequence length for input, selection and preprocessing of input features enhance

LSTM's ability to learn the patterns from traffic. The advanced features of both LSTM and CNN are combined to build a hybrid CNN-LSTM model. Before considering the CNN- LSTM model, SVM-CNN model was considered to detect the DDoS in IoMT devices, but the novelty of the model was very low. Support Vector Machine (SVM) are mainly designed for static data, and do not capture the evolving patterns, therefore application of SVM for intrusion detection will not perform well (GeeksforGeeks, 2023). Other drawbacks of SVM are it is computationally expensive, and output provided are binary classification outputs, when these are combined with CNN it will become a complex model. To overcome these issues hybrid CNN-LSTM model was opted.

3.2.3 Hybrid CNN-LSTM model

The hybrid CNN-LSTM model is trained using processed data where CNN model extracts spatial patterns from the sequential using features like 'frame. time_delta', 'tcp. flags', 'tcp.len', 'tcp.ack', 'tcp.connection.syn', 'tcp.connection.fin', 'tcp.connection.rst', 'tcpwindow_size_value', 'tcp. payload', 'tcp.hdr_len', 'ip.proto', 'ip.ttl'. LSTM extract the temporal and long-term patterns in the data through 'frame. time_relative', 'ip.src', 'ip.dst', 'tcp.srcport', 'tcp.dstport', 'ip.proto'. Using both the models together complex and dynamic IoMT traffic can be managed including diverse protocols and changing data flow. The combined model reduces the false positives and negatives which make it efficient in handling the changes. Each model is compiled with Adam optimizer and binary cross-entropy loss function and then trained for making predictions on the test data.

3.4 Model evaluation

The hybrid CNN-LSTM model is evaluated using different metrices. Accuracy measures the ability of the model to identify DDoS attack and identify how well the model perform across the given dataset, Precision and Recall evaluate models' performance in distinguishing malicious traffic and legitimate, Precision focus on models' ability to correctly identify DDoS attacks without any false positives, which reduces the false alarms. Recall will also measure how precisely the DDoS attacks are detected, ensure that the system is sensitive to the threats and make sure no threats are undetected. F1 score combine both Recall and Precision into single metric for the analyzing the performance of the model. For imbalanced classes it provides balanced view of model's performance which in turn helps our study to achieve best of precision and recall. To measure percentage of the true positives and false positives out of actual DDoS attacks True positive rate (TPR) and False positive rate (FPR) are used. To identify the severity of the malicious activity the malicious score can be used. The model output produces a probability score which indicates the packet being malicious, this probability can be directly calculated using this malicious score. It also gives the model utility in IoMT devices for detecting DDoS attack detection. The summaries of overall classification on different thresholds can be obtained from Area under ROC AUC, it measures the model's ability to distinguish between classes and provide summary that reflect the trade-off between TPR and FPR. These metrices gives an overview of different error handling which is important in IoMT devices. These metrics also ensure that model is not only accurate but also deployable in real-world scenario.

3.5 SNORT NIDS Integration and monitoring

SNORT is selected for detecting the DDoS attacks since it is open-source and can customize extensively. It can detect the real-time traffic and deployed for both prevention and detection. Which makes it perfect choice for the research. The saved combined model can be applied on a live traffic, which can be captured using traffic capturing devices. The saved model predicts the probability of each data being malicious and if the model predicts an instance of malicious traffic, SNORT rules are generated. Generated SNORT rule may look like:

Tab 1. SNORT RULE

alert <protocol><src_ip> <src_port>-> <dst_ip> <dst_port> (msg:"<message>"; sid:<sid>; rev:1;)	
alert	action that SNORT will take when the condition of the rule is met.
<protocol>	Specify the protocol that rule apply and define which type of network traffic is analyzed by the rule.
<src_ip>	Source IP address from which traffic is generated.
<dst_ip>-	Destination IP address where traffic is generated
<dst_port>	Destination port number for traffic specify the port number
msg	specify the message that will be displayed
Sid	SNORT ID, unique identifier assigned to the rule.
rev:1	Revision number indicating the version of the rule, to track changes and updates to rules

Generated rule was saved in local.rules inside /etc/snort/rules. According to the generated SNORT rule the SNORT configuration is edited. For testing SNORT a traffic is generated in the kali Linux and send to the ubuntu where SNORT has already setup. Using custom python script with ‘scapy’ a IoMT traffic is generated. SNORT monitor the traffic, and the output or logs are observed to verify whether it identifies the DDoS attack or not.

4 Design Specification

4.1 Environment Setup

For performing the deep learning latest version of Anaconda is installed and Jupiter Notebook is used for coding, data analysis and visualization. Python Libraries like TensorFlow and pandas NumPy scikit-learn tensorflow scapy was installed for building andtraining deep learning CNN-LSTM model and for customizing SNORT rules. scikit-learn for data preprocessing, model evaluation and metrices calculation, pandas for data manipulation and analysis, numpy for numerical computations, scapy for creating customized network traffic for testing.

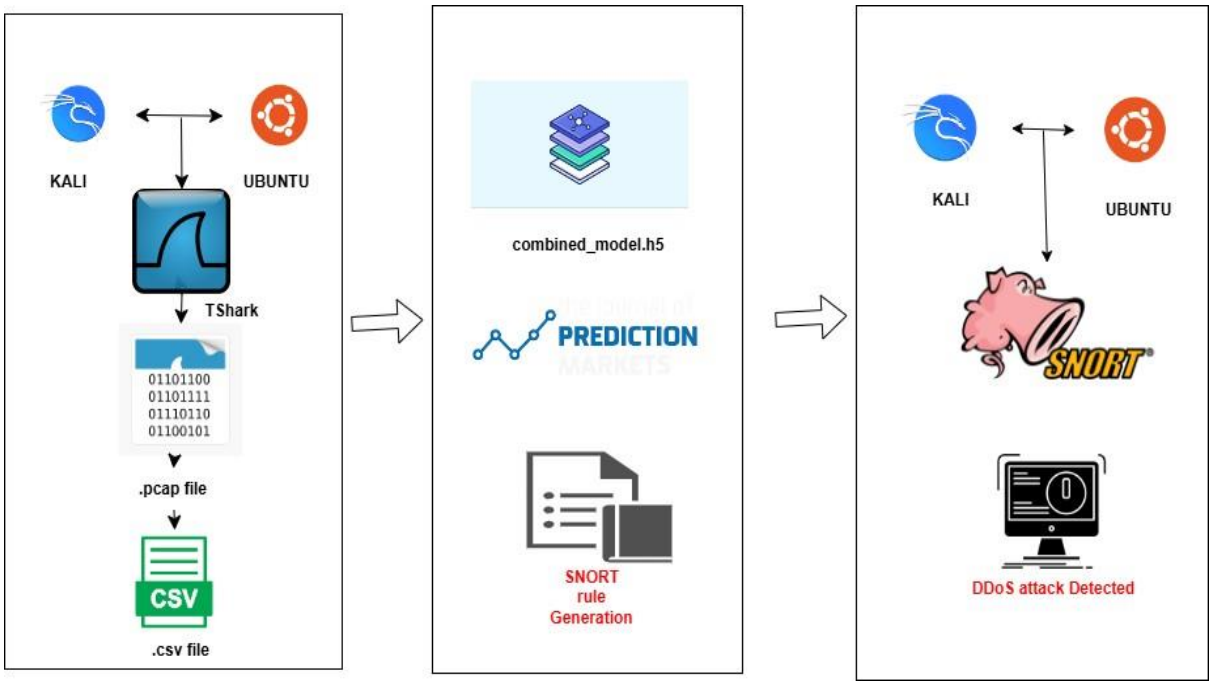

Figure 2: Design Specification

4.2 Deep Learning Model

Data combining

For Deep Learning**,** an IoT Healthcare dataset is selected. Datasets consist multiple datasets, therefore combined multiple datasets to obtain single dataset for making the analyzation and processing efficient. For combining the dataset several steps were done, at first directory consisting of all three datasets attacks.csv, environmentmonitoring.csv, and patientmonitoring.csv were retrieved into a specific path using 'os.listdir(folder_path)' function. Then an empty list named 'dataframes' was initialized to store the dataframe objects, and each dataframes were appended to it. By 'pd.concat(dataframes)' all the dataframes in the 'dataframe' were combined into single DataFrame. The combined dataframe were written into a CSV file.

Data Sampling:

To reduce the computational time and increase the accuracy rate from the combined dataset sampling was done. As a first step total number of samples selected as 'n_sample', and 3000 samples were selected with each sample getting almost 'samples_per_calss' as 1000. The availability of the class was checked and if any label was missing an error was raised to ensure the dataset can provide samples for each category. For each class randomly samples were selected, the 'random_state' ensure the reproducibility of samples. Each dataframes were filtered to separating each class before sampling. The sampled data from each class were combined into single dataframe 'sampled_data'. The combined data were shuffled to ensure random ordering, which is important for preventing any order-based pattern. The sampled and shuffled data was then saved to a single CSV file.

Data Preprocessing

Libraries like ColumnTransformer, Standarscaler, OneHotEncoder, to_categorical required for data processing were imported. All the numerical and categorial features are defined

under ‘num_features’, and ‘cat_features. With the help of ‘StandardScaler’ the numerical features are standardized, ensuring the data has a mean of 0 and standard deviation of 1 which helps the model learning process to be stable. ’OneHotEncoder’ is used to convert all the discrete variables into encoded format. With ‘ColumTransformer’ the operation of the numerical and categorical features was combined into one. The ‘preprocess_data’ function take the dataframe and apply the transformation using the ‘preprocessor’. After this, feature is transformed, and labels are extracted and converted to categorical format for the classification. The output of the OneHotEncoder may have sparse matrix that make difficulties for reshaping the data for neural network, therefore it is converted into dense layer. The data are reshaped because the LSTM expect input in 3D shape (batch_size, timestep, features). Reshaping is done with the dimensions like (X_train.shape[0], 1, X_train.shape[1]). X_train.shape[0] helps the model to maintain batch size ensuring each batch contain processed number of samples. Here ‘1’ represent the time step; third dimension ‘X_train. shape [1] correspond to number of features per samples. This is unchanged because each sample’s feature is processed as single time step by LSTM. After all these processes the data split for testing and training process. The test-train splitting is done is 80:20 ratio. 80 for training and 20 for testing.

CNN-LSTM Model Building

The combined CNN-LSTM model was developed that detect DDoS in IoMT devices using Keras. A sequential model is initialized in keras which means each layer has one input tensor and one output tensor.

Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(1, X_train.shape[2]), padding='same')

- Conv1D- Convolution layer helps in extracting spatial features,
- ‘filters=64’ number of filters use in convolution layer, where each filter learns different details of the data.
- ‘kernel_size=3’ size of convolution window. Kernal size 3 indicate layer consider 3-time steps at a time
- ‘activation=relu’ The Rectified Linear Unit (ReLU) activation function introduce non-linearity into the model, which helps to learn complex patterns.
- To ensure output dimension to be same as input dimensions without reducing the size of features map, padding is done.

- MaxPooling Layer is added to reduce dimensionality of features by taking maximum value over window of size 2, helps in reducing computational complexity. Two LSTM layer is added into the model, for first LSTM layer number of memory cells taken is 50 which allow the model to learn complex data and it return full sequence of outputs useful for the model and second LSTM layer with 50 units return only last output in output sequence. For preventing overfitting, a ‘Dropout’ layer is added between each LSTM layer and the value taken is 0.5. Flatten layer convert 2D matrix into 1D vector which prepare the data for fully connected layers, to learn complex patterns Dense layer with 100 neurons and ReLU activation and output Dense Layer for binary classification is used. Hybrid model is trained to predict classes for test data and the predicted probability is converted into binary class, then the model is saved as combined_model.h5 so that it could be supported across different

platforms and easy to transform model between host machine to the VMware the SNORT is configured.

4.3 SNORT Integration

VMware is used to setup SNORT and traffic generation. Ubuntu OS is installed on VM for SNORT and Kali Linux is installed for creating customized traffic for testing. Latest version of SNORT is installed in Ubuntu and configured it so that whenever an abnormal activity occurs it can detect. Python Scripts is customized using 'scapy' library and saved in Kali Linux to generate different medical IoT network traffic which mainly include TCP Flood, MQTT flood etc. At first the malicious traffic and normal traffic were sent to the Ubuntu machine and SNORT didn't detect any IoMT traffic, then this traffic was captured using tshark. The captured traffic contained various types of traffic which was saved in pcap file and converted to CSV file. On the new captured traffic dataset, the model was loaded, and the predictions are made. According to the predictions a SNORT rule was generated directly inside the local.rules. After this SNORT was turned on and again traffic was sent from the Kali to Ubuntu machine and this time SNORT correctly identified the IoMT traffic and displayed the message 'An IoMT traffic detected' was displayed in the log.

5 Implementation

- Deep learning model was created using the IoT healthcare dataset in Anaconda Jupyter Notebook and during model creation each model was build and evaluated its performance metrices using various features.
- Using tshark the real-time traffic to the ubuntu was collected and converted the pcap file into csv file. And this dataset was used as the input to the saved model and generated the predictions and customized rule based on it.
- The SNORT was installed and configured in the Ubuntu inside VM ware. Inside the SNORT folder it contained SNORT rules and SNORT Configuration files, this snort.config. file copy was made and saved, so that if any data is lost, we could retrieve from it. Inside the snort.config.file, the network variables are updated that represent the IP address and network configuration to our environment. The preprocessor for detecting specific traffic patterns and protocols is enabled. The output settings are made, and an output are saved in snort.log for future use. Updated the custom classification and priorities for detecting DDoS attacks using the new generated rule. Include $RULE_PATH/local.rules inside the config.file.
- To share the file from the host machine to the VM ware, the open-vm-tools packages which provide functions like are folder support, drag and drop and clipboard sharing between host and VM are installed. A directory was created, and the shared folder was mounted. The 'fstab' file, which is mainly used to define how disk partitions, other block devices, remote filesystem should mount into the filesystem. And to this 'fstab' file a line is added to configure the system to automatically mount the shared folder. The saved model was transferred using this method.
- The customized rule was directly added to the local.rules, and this rule alert the home network about tcp traffic from any devices coming to IoMT devices.

- A python script for IoMT traffic was crafted using the Scapy library in text editor and saved as .py file in Kali Linux. Several scripts were made to generate different types of floods, in one of the script a MQTT packets were defined and a function construct MQTT publish packets consisting of IP, TCP and MQTT layers were constructed. The script is made to generate random heart rate value between 60 to 100 beats per minute and random blood pressure values with systolic value between 110 and 1130 and diastolic values between 70 to 90. Then device/heartrate and devices/blood pressure are defined and associate these with corresponding payload. For each payload pair an MQTT packet is created and sent it using Scapy's send function.
- SNORT in the Ubuntu was made activated, so that at any time when the OS starts working it starts to scan all the external traffic that comes to the system. When the DDoS traffic was sent to the system, SNORT instantly identified the DDoS attack in the network and gave an alert.

6 Evaluation

One of the aims of the research was to evaluate the performance of the hybrid CNN-LSTM model. This evaluation was done by obtaining the model summary and metrices like Accuracy, Precision, Recall, F1- Score and ROC AUC of CNN and LSTM model separately. The Classification Report, Malicious Score, Confusion Matrix, ROC curve and Precision-Recall curve of combined model is also obtained for evaluation.

Tab 2. Evaluation Table

Accuracy	**(TP+TN) / Total Number of Samples**	**0.9510**
Precision	**TP / (TP+FP)**	**0.9503**
Recall	**TP / (TP+FN)**	**0.9495**
F1 score	**2 * ((Precision*Recall) / (Precision + Recall))**	**0.9499**
Malicious Score		**0.9544**

Tab 3. Classification Report

	Precision	**Recall**	**F-1 Score**	**Support**
0	**0.95**	**0.96**	**0.96**	**573**
1	**0.95**	**0.94**	**0.94**	**427**
Accuracy			**0.95**	**1000**
Macro avg	**0.95**	**0.95**	**0.95**	**1000**
Weighted avg	**0.95**	**0.95**	**0.95**	**1000**

6.1 Combined Model Summary

Tab 4. Hybrid CNN-LSTM model summary

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 1, 64)	155,008
max_pooling1d_1 (MaxPooling1D)	(None, 1, 64)	0
lstm_4 (LSTM)	(None, 1, 50)	23,000
dropout_4 (Dropout)	(None, 1, 50)	0
lstm_5 (LSTM)	(None, 50)	20,200
dropout_5 (Dropout)	(None, 50)	0
flatten_1 (Flatten)	(None, 50)	0
dense_4 (Dense)	(None, 50)	2,550
dropout_6 (Dropout)	(None, 50)	0
dense_5 (Dense)	(None, 2)	102

Total params: 200,860 (784.61 KB)
Trainable params: 200,860 (784.61 KB)
Non-trainable params: 0 (0.00 B)

6.2 Confusion Matrix

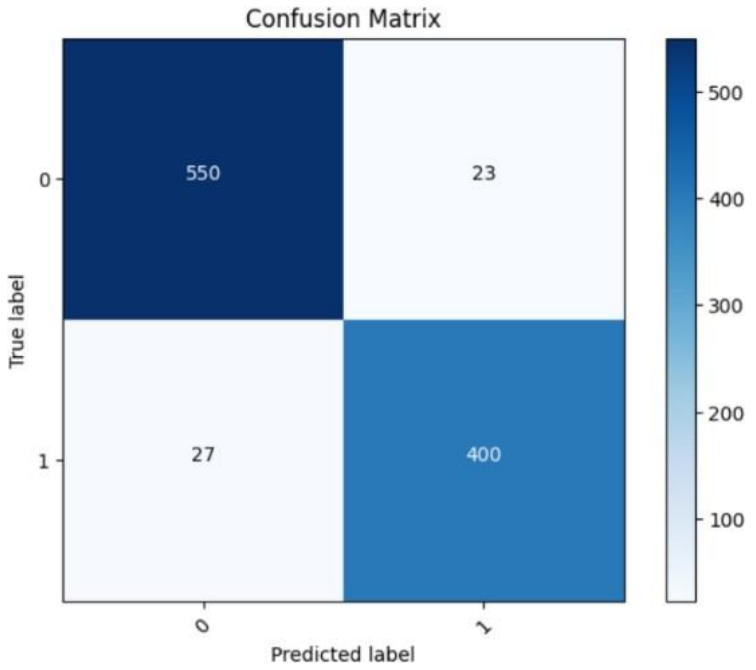

Figure 3: Confusion Matrix

The visualization of the performance was obtained using confusion matrix. The confusion matrix compares true label with predicted label and summarizes the results TN=550, FP=23, FN=27 and TP=400.

6.3 Receiver Operating Characteristic (ROC) Curve

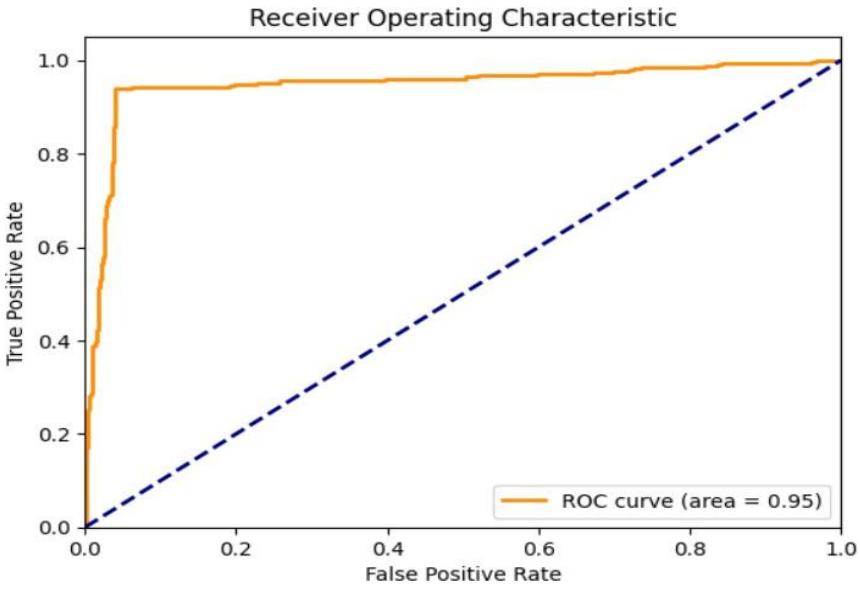

Figure 4: Receiver Operating Characteristic (ROC) Curve

Diagonal blue line represents random classifier which has AUC of 0.5 and is compared against ROC curve. ROC curve in graph is 0.95 which indicate there is a 95% chance that model will correct differentiate between true positives and true negatives.

6.4 Precision-Recall Curve

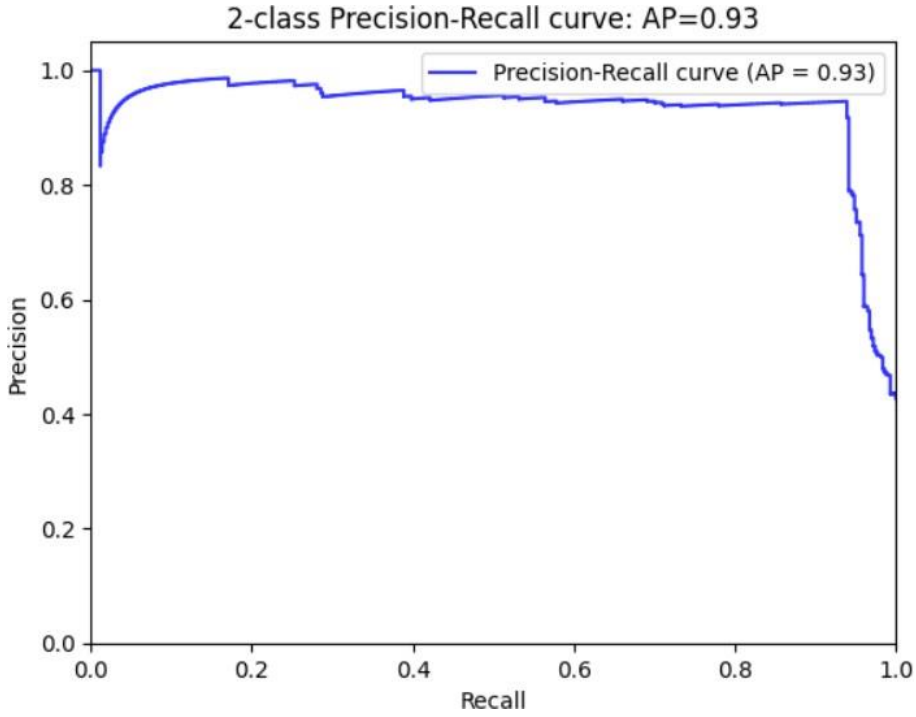

Figure 5: Precision-Recall Curve

The Y-axis represent the precision, X-axis is Recall-the true positive rate. The are under Precision-Recall curve denoted as Average Precision (AP) and it is valued as 0.93.

6.5 SNORT Rule Generated

```
## Snort rule based on CNN-LSTM Model
alert tcp any any -> 192.168.136.129 1883 (msg:Detected MQTT publish packet to sensors/temperature; sid:1000001; rev:1;)
alert tcp 192.168.136.130 20 -> 192.168.136.129 1883 (msg:Detected traffic in IoMT device; sid:10000073453453; rev:1;)
```

Figure 6: SNORT Rule Generated

Two SNORT rules are generated here, the first rule detect MQTT packets that has been sent to temperature sensors, which indicate the monitoring of traffic to IoT based temperature sensors. Second rule focus on traffic between specific devices which monitor malicious traffic patterns that may cause security issues, intrusion or data leak.

7 Conclusion and Future Work

This research successfully developed hybrid CNN-LSTM model integrated with SNORT NIDS for identifying DDoS in IoMT. The study answers the research question of how this model identify the DDoS attacks in IoMT by achieving perfect accuracy, precision, recall and ROC-AUC score. These results highlight the capability of the model to handle the complex IoMT traffic and detect the traffic pattern. Integration of the SNORT improve model's applicability by enabling the real-time attack detection through customized SNORT rule. The study was mainly done to address three objectives, the first objective was achieved by developing the hybrid model and the generation of customized rule based on the developed model helped to complete the second objective as well. Third objective was to evaluate the performance of the hybrid model using various metrices and while evaluating the model 95% accuracy, precision, Recall, F-1 score, ROC-AUC, malicious score it was obtained. The confusion matrix, ROC curve and Precision-Recall curve also helped in completing this evaluation. This research also had certain limitations, like usage of synthetic data for maintaining the privacy and compliance with data regulations. If the model is overly tuned its ability to generalize real-world conditions may compromised. Since model performance was evaluated in controlled environments, it would be better if the testing is done in Real-world with varying traffic patterns and device types. Here the dataset chosen is synthetic dataset, in future dataset from real-world devices can be used for the building of the models. After research it was clear that IoMT devices are highly vulnerable to DDoS attacks and there is a need of advanced security system in protecting the IoMT system. The successful implementation of this hybrid model addresses this security challenge effectively and relevance of this research extend to all healthcare institutes and IoT device manufactures.

References

Kauhsik, B., Nandanwar, H., & Katarya, R. (2023) 'IoT Security: A Deep Learning-Based Approach for Intrusion Detection and Prevention' *in 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)*. Bengaluru, India, 20-21 October 2023**,**pp. 1-7.

S. Al-Sarawi, M. Anbar, R. Abdullah and A. B. Al Hawari, "Internet of Things Market Analysis Forecasts, 2020–2030," 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 2020, pp. 449-453.

Saheed, Y. K., & Arowolo, M. O. (2021)' Efficient Cyber Attack Detection on the Internet of Medical Things-Smart Environment Based on Deep Recurrent Neural Network and Machine Learning Algorithms'.

Rana Abubakar, Abdulaziz Aldegheishem, Muhammad Faran Majeed, Amjad Mehmood, Hafsa Maryam, Nabil Ali Alrajeh, Carsten Maple, and Muhammad Jawad. (2020).'An effective mechanism to mitigate real-time DDoS attack 'IEE Access.

Lai, Y.-C., Yu, C.-L., Liao, M.-L., Lin, Y.-S., Chang, Y.-C., & Chen, J.-L. (2023)'An Intelligence Defense System with SNORT Rules' *in Proceedings of the 25th International Conference on Advanced Communication Technology (ICACT)***.** *Pyeongchang, Korea,19-22 February 2023,* pp. 249-254.

F. Hussain, (2023)"IoT Healthcare Security Dataset," *Kaggle.* Available at**:** https://www.kaggle.com/ds/2852100. DOI: 10.34740/KAGGLE/DS/2852100.

Abdulrezzak, S., & Sabir, F. A. (2023) *'Enhancing Intrusion Prevention in Snort System*' *in 2023 15th International Conference on Developments in systems Engineering (DeSE).* Baghdad & Anbar, Iraq, 09-12 January 2023**,** pp. 88-93**.**

Idrissi, I., Azizi, M., & Moussaoui, O. (2020)' IoT security with Deep Learning-based Intrusion Detection Systems: A systematic literature review' *in 2020 Fourth International Conference On Intelligent Computing in Data Sciences* ***(ICDS)*****.** Fez, Morocco21-23 October 2020, pp. 1-10.

Reddy, K. P., Kodati, S., Swetha, M., Parimala, M., & Velliangiri, S. (2021)' A Hybrid Neural Network Architecture for Early Detection of DDOS attacks using Deep Learning Model*s' in 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*. Trichy, India, 07-09 October 2021, pp. 323-327.

Faruqui, N., Yousuf, M.A., Whaiduzzaman, M., Azad, A., Alyami, S.A., Liò, P., Kabir, M.A. and Moni, M.A. (2023) 'SafetyMed: A novel IoMT intrusion detection system using CNN-LSTM hybridization', Electronics, 12(17), p. 3541.

GeeksforGeeks, (2023) ‘Support vector machine in Machine Learning’,*(2023)* Available at: Support vector machine in Machine Learning - GeeksforGeeks [Accessed 02 April, 2024]

Scikit-learn ‘StandardScaler’ Available at: StandardScaler — scikit-learn 1.5.1 documentation [Accessed on June 12, 2024]

GeeksforGeeks. (2024) ’Introduction to Convolution Neural Network’ Available at: Introduction to Convolution Neural Network - GeeksforGeeks [Accessed on June 12, 2024]

GeeksforGeeks, *(*2024)*’What is LSTM-Long Short Term Memory’* Available at: What is LSTM - Long Short Term Memory? - GeeksforGeeks[Accessed on June 12, 2024]

Priyadarshi, P. (2020)*'Calculating Number of Parameters in a LSTM Unit/Layer'*, Medium. Available at: https://medium.com/@priyadarshi.cse/calculating-number-of-parameters-in-a-lstm-unit-layer-7e491978e1e4 [Accessed 02 April 2024]

Medium.(2023)*'Deep Learning Course Lesson 11: Model Evaluation Metrics'*, Available at: https://medium.com/@nerdjock/deep-learning-course-lesson-11-model-evaluation-metrics-d85d0b85bcca [Accessed 02 April 2024]