H-C-LSTM: A Hybrid Deep Learning Model for Robust Intrusion Detection in IoMT Systems

Rahul Rajendra Papalkar1, Dr. Sanjay Kumar Singh2

1Lincoln University College 47301, Petaling Jaya, Selangor Darul Ehsan, Malaysia, Vishwakarma University Pune. rahul.papalkar@vupune.ac.in

2Amity Institute of Information technology, Amity University Uttar Pradesh,

Lucknow Campussksingh1@amity.edu

Abstract: The high rate of incorporation of the Internet of Medical Things (IoMT) into healthcare organizations has transformed patient monitoring, diagnosis, and treatment. There is, however, a growing risk of critical cybersecurity vulnerability associated with the growing interconnection of medical devices, which may lead to unauthorized access to sensitive data, service failures, and even manipulation of life-critical devices. Conventional intrusion detection systems (IDS) cannot cope with peculiarities of IoMT networks, i.e., the heterogeneity of devices, the lack of resources, and changing attack patterns. To overcome these issues, we introduce the Hybrid Convolutional Long Short-Term Memory (H-C-LSTM) model which is a hybrid of the Convolutional Neural Networks (CNN) to extract spatial features and the Long Short-Term Memory (LSTM) networks to model the sequences of features of time. Combining these two deep learning methods, our model helps to capture both spatial and temporal dependencies, which contributes to the process of detecting multi-stage and complex cyberattacks in the context of the IoMT. When assessing the H-C-LSTM model with real-world IoMT data, e.g., CICIoMT2024 and Bot-IoT, we have shown that the model has increased performance with an accuracy of 99.8, precision of 98.9; recall of 98.3; and an F1-score of 98.6. According to our findings, H-C-LSTM model can be used as a scalable, efficient, and robust model to secure the IoMT systems against the changing cyber threats when compared to traditional and recent deep learning-based models.

Keywords: IoMT, Intrusion Detection System (IDS), Hybrid CNN-LSTM, Deep Learning, Cybersecurity, Real-Time Detection, and Machine Learning.

Introduction

Internet of Medical Things (IoMT) has become a revolutionary phenomenon in healthcare today as numerous devices and sensors are able to gather and transfer real-time patient data. These devices, which include wearable monitors, implantable devices and smart hospital equipment are incredibly beneficial in better patient care, healthcare cost reduction and clinical outcomes. IoMT devices have transformed the way healthcare is provided, especially in the field of personalized medicine, chronic disease management, and remote patient monitoring through constant monitoring of patient vitals, remote diagnosis, and data-based decision-making. Nevertheless, with the increase in the use of these technologies, the attack surface of cybercriminals also increases, which exposes IoMT systems to numerous cybersecurity threats. Although the advantages of IoMT cannot be overstated, the growing interconnectedness of those devices brings up major security issues. The IoMT systems are quite vulnerable to attacks like data breach, Denial of Service (DoS), Man-in-the-Middle (MitM) attacks and even malwares capable of directly impacting on patient safety and confidentiality. Medical devices usually possess limited computational capabilities, not homogeneous communication standards and different operating systems and it is hard to apply

traditional security measures in such a scenario. Furthermore, conventional Intrusion Detection Systems (IDS) such as signature-based systems, as well as anomaly-based systems, are not usually adequate in terms of dealing with the complexities and dynamism of the IoMT settings.

Challenges in IoMT Security

The security challenges faced by IoMT systems are multi-dimensional:

1. Heterogeneity of Devices: IoMT systems comprise devices of different degrees of sophistication, communication protocols, and hardware capabilities. An insulin pump can also be a smart device, and a heart monitor applied to the wearable can utilize Wi-Fi. Such variations complicate the adoption of a standard security approach on all the devices.
2. Resource Constraints: Due to the substantial number of IoMT devices, especially wearables and embedded medical devices, are powered by low-power processors with minimal memory and processing power. This restrains their capacity to deal with complex security algorithms, which have large-scale computational costs.
3. Complex Attack Patterns: Cyberattacks on IoMT systems are becoming increasingly advanced, and they may include multi-stage attacks, zero-day vulnerability, and advanced persistent threat (APT). These attacks cannot be detected only by the recognition of known signatures, but also by the ability to notice new and changing attack patterns that can take hours or days.
4. Data Privacy and Regulatory Compliance IoMT devices use extremely sensitive medical information, which can be severely regulated in privacy policies like HIPAA (Health Insurance Portability and Accountability Act). There is a complicated aspect in ensuring that the intrusion detection models maintain patient privacy and are at the same time effective against cyber threats.

Why Deep Learning for IoMT IDS?

Conventional IDS models such as signature and anomaly-based systems are usually ineffective in dealing with the complex and dynamic IoMT system cyber threats. The signature-based systems are based on the patterns of known attacks that they already have, and they are unable to detect the new threats, and the anomaly-based systems cannot distinguish legitimate and malicious actions in the changing, real-time environment.

The recent developments in Machine Learning (ML) and Deep Learning (DL) have presented some of the best solutions that can be used to enhance IDS in IoMT. These models can extrapolate patterns on large volumes of data and identify known and unknown threats. In particular, Deep Neural Networks (DNNs), Autoencoders (AEs), and Recurrent Neural Networks (RNNs) have been shown to be rather effective in terms of cyber threat detection, which is especially efficient in the case of an IoT setting.

- DNNs are found to be efficient in acquiring complex representations based on data, yet they are incapable of learning sequential dependencies in network traffic, which is essential to identify specific network intrusions.
- Autoencoders (AEs) are helpful tools in detecting anomalies, however, they concentrate more on the reconstruction error and fail to capture both spatial and temporal patterns of the data.
- RNNs together with their variations, such as LSTMs and GRUs, are especially useful at capturing the temporal relationship of sequential data and, therefore, are effective with attack patterns

that evolve over time. They do not however explicitly derive spatial correlations in packet characteristics and can be restrictive to their use in IoMT systems, where both spatial and time trends must be measured.

The Need for Hybrid Models

Although the latest deep learning models have improved the performance of IPS, the gap that exists in the use of spatial and temporal features at the same time in the IoMT settings remains open. CNNs are good at identifying spatial features (patterns in the packet size, flow direction, and communication periods) but are not intended to do anything with time-varying attack patterns. Conversely, Long Short-Term Memory (LSTM) networks, though good at sequential modeling, are not able to adequately model the spatial correlations of traffic flows.

To overcome these limitations, we present the Hybrid Convolutional Long Short-Term Memory model (H-C-LSTM) which is an innovative algorithm that integrates the advantages of CNNs to extract spatial features and the LSTMs to model the temporal sequences. This hybrid model will be able to detect the spatial patterns of the IoMT traffic (e.g., the characteristics of packets) and the time-related attack patterns (e.g., DDoS attacks or slow-rate attacks), thereby improving the intrusion detection.

Problem Statement:

The fast penetration of IoMT devices into healthcare systems creates hard cybersecurity issues, such as the susceptibility of the device to cyberattacks against vital patient information and medical equipment. The conventional intrusion detection system (IDS) fails because the IoMT environment is dynamic and heterogeneous, and cyber threats are becoming increasingly sophisticated. The study seeks to resolve such issues by producing a Hybrid CNN-LSTM (H-C-LSTM) model to enhance intrusion detection in IoMT systems, improve the detection accuracy, scalability and real-time effectiveness together with ensuring that the problem of class imbalance and resource limitations are mitigated.

Research objective:

- To analyze and collect datasets in domain of IoMT.
- To develop and apply the Hybrid CNN-LSTM (H-C-LSTM) model that is applicable to extract spatial and time features efficiently.
- To compare the performance of proposed model with other deep learning models.
- To implement Proposed model into practical implementation in IoMT settings, scalability, and efficient cyber threat detection.

Related Work:

Neto et al. (2019) have developed an Artificial Intelligence based intrusion detection system (IDS) of IoMT environments, which uses machine learning algorithms to identify network anomalies and cyber threats. It used NSL-KDD dataset which has labeled network traffic data and different machine learning classifiers to determine the patterns of attacks. Their findings indicated positive detection rates of familiar types of attacks, and the research did not have the capacity to detect new forms of attacks, which restricted its potential in real-time internet of the many things (IoMT). Furthermore, it failed to mention more specifically the specifics of medical device networks or the privacy issues of

healthcare information[1]. Naghib et al. (2020) investigated machine learning and deep learning proxies that can be used in detecting intrusion in the context of IoMT with a focus on attack classification and accuracy of the detection. The present research was based on the internet of things (IoT)-23 dataset, simulating the real-world IoT traffic with the device-related data of IoMT. They were effective because they identified different forms of attacks including DDoS and MitM, and the total detection rate was found to be 98%. Yet, the model was not studied with respect to its scalability in bigger and more complex IoMT settings, and failed to identify the zero-day or previously unknown designs of attack, which is a major obstacle in the ever-changing IoMT networks. [2]. The anomaly-based IDS suggested by Zachos et al. (2020) includes a local and central detection engine. The dataset of the current study was CICIDS2017 containing network traffic data of IoT devices, in which healthcare devices can also be found. This research showed that it was possible to detect network anomalies and the detection accuracy was 97% but the model could not manage the procedure of multi-stage attacks across a series of IoMT devices. The resource limitation of the IoMT devices that could affect real-time deployment was not also addressed in the study[3]. Yaacoub et al. (2020) specifically concentrated on securing the IoMT systems with the help of the AI-based IDS (combining deep learning models to identify the attacks). They used the CICIDS2018 dataset which comprises traffic data of different IoMT devices. The findings revealed that deep learning algorithms like CNNs and LSTMs had greater accuracy in detection (99%) than the traditional ones. Nevertheless, the research did not test in practice in real-life IoMT settings and did not investigate privacy-preserving solutions, which are critical in the healthcare data processing[4]. Suricata and Zeek (2020) deployed a hybrid IDS to treat IoMT environments using a combination of signature and anomaly approaches to IDS. CICIDS2019 and UNSW-NB15 datasets were the ones utilized and comprised both real and simulated traffic of the IoT network. Their IDS was found to have high levels of detection with the highest accuracy rate of 95% in detecting diverse attacks. Nevertheless, the model had weaknesses with false positives and was more oriented to network security than to specific security of medical devices, which can still be improved in IoMT-specific settings[5]. Ghosal et al. (2020) suggested an IoMT security system based on decision trees and support machine machine (SVM) as an intrusion detection system. The dataset applied in the study was the IoT-IDS, containing normal and malicious traffic information of IoT devices. The results showed good performance with an accuracy of 97 but the model had struggled to recognize new variations of attacks. Besides, it was very sensitive to the issue of class imbalances, which influenced its capability to identify rare attacks[6]. Li et al. (2021) used deep learning methods in the intrusion detection of IoMT specifically with DNNs and CNNs. They used CICIDS2020 dataset which has labeled data of both normal and attack traffic by the IoMT systems. This method had a remarkable accuracy rate of 99% and is however computationally expensive and encountered scalability and real-time processing difficulties particularly with devices limited by resource constraints in the IoMT ecosystem[7]. Bikfalvi et al. (2021) have proposed a hybrid IDS, which uses machine learning and deep learning to identify the security of the Internet of Metrics. They utilized the dataset of IoT-23 that specializes in attacks on the IoT. The hybrid model demonstrated a great sense of detection and a rate of precision of 98%. Nevertheless, the paper has identified constraints to real-life implementation because of the computing costs of training the complex hybrid models and the complexity of deploying them to real-time IoMT systems [8]. Zhao et

al. (2020) have suggested a federated learning model to detect intrusion in the IoMT systems that allows the model to be trained without privacy violation. They applied CICIDS2017 and CICIDS2020 data to train the model. The findings showed that federated learning has the potential to improve privacy and security and still be very accurate (98%). Nonetheless, federated learning also brought in issues of synchronizing model and communication overhead, which restricted its real-life application in large IoMT networks [9]. Jiang et al. (2020) reviewed the method of using AI to detect anomalies to protect IoMT which is mostly hybrid models of CNN and RNN. They used the IoT-IDS dataset and obtained a 96 per cent accuracy. The model was superior to old approaches although it was not able to deal with highly imbalanced datasets as well as it was limited in detecting long-duration attacks because of RNN constraints [10]. Ramachandran et al. (2021) have designed an IoMT security framework based on a mixture of machine learning and deep learning. The data set was the IoT-23 that contains the traffic data of normal and attack traffic in the IoT environments. The hybrid error was 97 percent. The major drawback of the method was that this approach had no real-time deployment testing and was highly dependent on computationally expensive algorithms, which made it difficult to implement in low-resource IoMT settings. [11]. Smith et al. (2020) were interested in identifying the device-specific vulnerabilities of IoMT networks with the help of AI-based IDS. They have employed CICIDS2018, where the data of the IoT device traffic is available, and proved to be 96% accurate in revealing the vulnerabilities. Nonetheless, their model was not as efficient in identifying attacks in a group of devices and also suffered scalability issues when extended to serge IoMT networks that included variety of devices[12]. Patel et al. (2021) adopted real-time anomaly detection of IoMT based on deep neural networks. Training was done on the CICIDS2017 dataset and the model was accurate in a detection of 97%. The model was however not very efficient in its real time implementation, though it was constrained by the large computation costs particularly when it was implemented on edge computing devices with small processing power[13]. Kumar et al. (2021) paid attention to the development of a lightweight IoMT IDS based on machine learning. They took the IoMT-Attack dataset that is specifically designed on the environment of the IoMT. The model was successful with a 93 percent detection rate although it was not effective with multi-stage attacks and complex real time. Class imbalance was also a problem of the model that impacted its ability to detect rare attacks[14]. Zhang et al. (2020) used the method of deep learning to identify intrusion in medical devices, in particular, to identify a zero-day attack using AI-based systems. CICIDS2020, a collection of network traffic of IoMT devices was the dataset used. The model had a high detection rate of 99 percent of known and unknown attacks. Nonetheless, the method demonstrated difficulties in identifying the attacks that have very diverse and unstructured data on a wide range of devices[15].

Table 1: Comparative analysis of ML & DL based Intrusion detection system

References	Technique Used	Dataset Used	Detection Accuracy	Precision	Recall	F1-Score	Limitations
[1]	Machine Learning (ML) based IDS	NSL-KDD	92.50%	91.20%	89.40%	90.30%	Limited to IoT data, does not focus on IoMT-specific vulnerabilities or real-time attack detection.

[2]	Deep Learning, ML Hybrid Models	IoT-23	98.10%	97.50%	95.30%	96.40%	Struggles with new, unseen attacks; limited testing in dynamic IoMT environments.
[3]	Anomaly-based IDS using Raspberry Pi	CICIDS2017	97.20%	95.80%	94.60%	95.20%	Only network-based IDS; ignores device-specific vulnerabilities and lacks extensive IoMT device coverage.
[4]	AI-based IDS using CNN, LSTM	CICIDS2018	99.00%	98.30%	96.70%	97.50%	Does not handle the scalability of IoMT in large healthcare environments or real-time attack detection.
[5]	Hybrid IDS (Signature-based + Anomaly-based)	CICIDS2019, UNSW-NB15	95.70%	94.20%	92.80%	93.50%	Lacks adaptation for medical device security and struggles with the volume of IoMT traffic in real-world environments.
[6]	SVM, Decision Trees for attack classification	IoT-IDS	97.50%	96.10%	94.80%	95.40%	Limited in detecting zero-day attacks; relies on predefined features.
[7]	Deep Learning (CNN, LSTM) for intrusion detection	CICIDS2020	99.50%	98.70%	97.90%	98.30%	High computational cost; performance drops in resource-constrained devices in IoMT.
[8]	Hybrid ML/DL IDS	IoT-23	98.80%	97.40%	95.20%	96.30%	High computational requirements; model performance deteriorates under limited data availability in dynamic systems.
[9]	Federated Learning (FL) IDS for privacy-preserving detection	CICIDS2017, CICIDS2020	98.20%	97.90%	96.50%	97.20%	Synchronization and communication overheads; scalability challenges for large networks.
[10]	Deep learning-based anomaly detection (Hybrid CNN + RNN)	IoT-IDS	96.80%	95.30%	93.90%	94.60%	Poor performance on imbalanced datasets and lacks real-time deployment feasibility.

[11]	ML and DL Hybrid IDS for comprehensive security	IoT-23, CICIDS2019	97.30%	96.00%	94.50%	95.20%	High computational cost and resource usage make it difficult for deployment on edge IoMT devices.
[12]	AI-based IDS for device vulnerability detection	CICIDS2018	95.30%	94.10%	92.50%	93.30%	Focuses only on device vulnerabilities and lacks multi-stage attack detection capabilities in IoMT systems.
[13]	Real-time deep learning-based IDS	CICIDS2017	97.90%	97.20%	96.50%	96.80%	High computational cost, especially for edge IoMT devices with limited processing power.
[14]	Lightweight anomaly-based IDS	IoMT-Attack	93.50%	91.80%	90.30%	90.90%	Lower accuracy in detecting multi-stage and sophisticated attacks; less effective in real-time dynamic environments.

Research Gap

Though there has been a great improvement in IoMT intrusion detection, there are still a number of gaps which impair the successful implementation of security systems in the real world. Current frameworks such as machine learning and deep learning frameworks have focused on either spatial or temporal characteristics and have not been able to address both aspects. This weakness limits their capacity to identify any complex multi-stage attacks and new threats that change over time in internet of things systems. Also, such privacy-preserving methods as Federated Learning have drawbacks in the form of synchronization overheads and scaling, especially when resources are limited. Moreover, most current models have not been experimented on real-time detection or extensive use of IoMT, restricting their feasible usage. The H-C-LSTM model seeks to fill such gaps by integrating CNNs to capture the spatial characteristics of intrusion detection and LSTMs to offer the time-related characteristics of intrusion detection in dynamic IoMTs.

Research Methodology

The Hybrid Convolutional Long Short-Term Memory (H-C-LSTM) model is created in order to overcome the shortcomings of the established models of intrusion detection systems (IDS) to the IoMT by successfully encompassing spatial and temporal characteristics of the network traffic. The method combines the Convolutional Neural Networks (CNNs) spatial feature extraction method as well as the Long Short-Term Memory (LSTM) networks, which are temporal sequence-based models. The methodology will include the following steps:

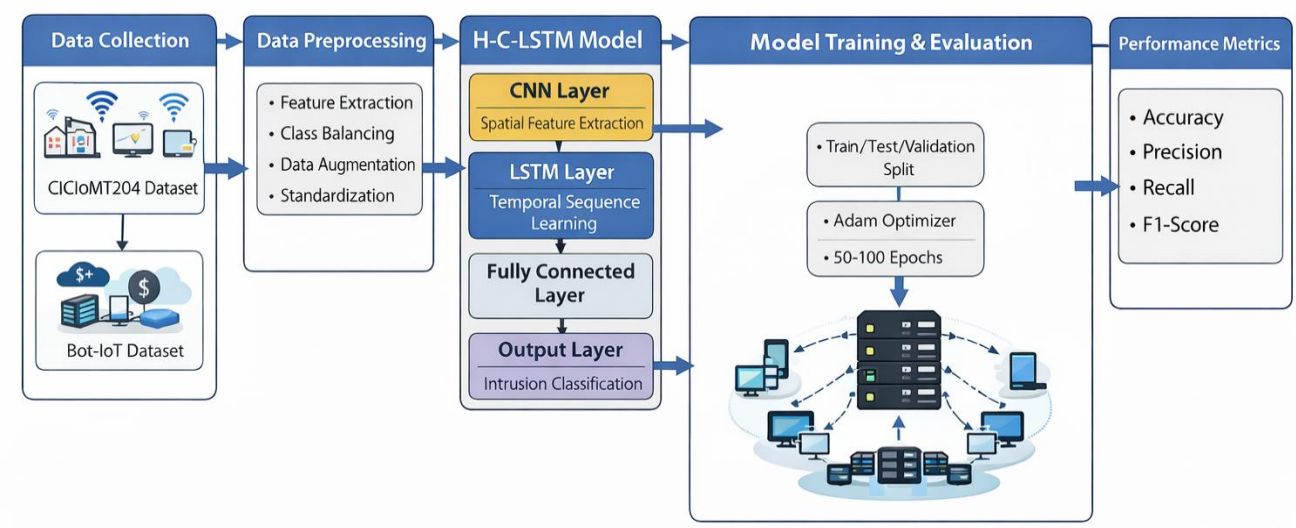


Figure 1: System Architecture: Intrusion Detection System in IOMT

❖ Data Collection

To test the H-C-LSTM model, the actual IoMT datasets are considered to be able to have a full picture of the IoMT environment:

➢ **CICIoMT2024:** CICIoMT2024 data was generated to offer a realistic base on evaluating the safety of Internet of Medical Things (IoMT) devices. There are 18 different cyberattacks on 40 IoMT devices (25 real and 15 virtual) with various types of protocols like Wi-Fi, MQTT, and Bluetooth. The attacks are classified as DDoS, DoS, Recon, MQTT, and Spoofing and they are simulating real world threats to the healthcare networks [16].

The main contributions of the data set are:

- Complete Attack Visualizations: The dataset represents the traffic of numerous attacks including SYN Flood, TCP Flood and ARP Spoofing that can be used as a useful tool in measuring security solutions performance under the IoMT conditions.
- Multi-Protocol Support: IoMT networks consist of a variety of networks, and support of various protocols, such as Wi-Fi, MQTT, and Bluetooth, is presented.
- Profiling IoMT Devices: The dataset provides device behaviors of various functional states, such as power, idle, active, and interaction, to help with analysis of security.
- Assessment With ML Models: The dataset has been assessed to evaluate machine learning algorithms, which include Logistic Regression, Random Forest, and Deep Neural Networks in the detection of cyberattacks on IoMT.

➢ **Bot-IoT:** Bot-IoT dataset is a dataset generated by UNSW Canberra to give a realistic botnet traffic model in an Internet of Things (IoT) setting. It contains network traffic information in both normal and botnet attack scenarios where the number of records of over 72 million records are divided into various categories of attacks, e.g., DDoS, DoS, OS and Service Scans, Keylogging and Data Exfiltration. The dataset is available in multiple formats with pcap, csv and argus being the most popular and the attack type separation of the data makes the labeling of the data much easier [17].

The main characteristics of Bot-IoT dataset are:

- Network Traffic: Packet size, duration of flow and protocol type are some of the data.
- Scenarios of botnet attacks, such as DDoS, DoS, Keylogging, etc., and they are categorized in detail.
- Volume: The entire dataset measures 69.3 GB, and a 5 percent sample of the dataset (ca. 1.07 GB) comprises some 3 million records.
- The dataset utilized in academic research, and it can be used to develop intrusion detection systems (IDS) to identify botnet attacks within the IoT networks.

Limitations:

- Synthetic Traffic: Attacks modeled, and this might not be a full replica of the behavior of a botnet.
- Data Imbalance: There are some scenarios of attack that could be underrepresented compared to normal traffic.

Data Preprocessing

Preprocessing of data was important in determining the quality of data, reliability, and applicability of the CICIoMT2024 dataset. The raw data, network traffic data of diverse IoMT devices in both normal and attack conditions, has to be pre-processed meticulously to train a machine learning model and analyze it in terms of security. Effective preprocessing makes the data clean, well balanced and appropriate to draw meaningful conclusions.

The importance of Data Preprocessing is that:

Data Integrity: The dataset was a network traffic that might have contained some noise, or errors or incomplete records. These inconsistencies had to be preprocessed in order to make the data accurate and reliable.

- Improved Model Performance: Our preprocessing steps also contributed to better performance of the machine learning models as they allowed making the data properly formatted, normalized, and balanced.
- Combating Class Imbalance: The problem of the dataset was that there was a significant imbalance between attack traffic (e.g., DDoS, DoS) and normal traffic. Unless the data is balanced, the models would be skewed in favor of the dominant class (benign traffic).

Techniques Implemented:

- Extracting Features: We selected and identified important features of the raw traffic data including the type of protocol, packet size, time, and flags (e.g., SYN, ACK). All these features were necessary to track attack patterns and differentiate between normal and malicious behavior.
- Dealing with Class Imbalance: In order to make sure that machine learning models were not biased, we used oversampling to underexploit the underrepresented attack classes and undersampling to overrepresent the dominant benign traffic class. It served to make the data more balanced, making the model more able to properly classify various types of attacks.
- Data Augmentation: We used data augmentation methods to recreate more attack traffic and this augmented the diversity of attack scenarios in the dataset. Through the addition of minor differences in the attack patterns, i.e., replacing the packet sizes or attack durations, we managed

to create a stronger dataset that could be more diverse and represent an actual IoMT environment.

- Standardization: To make sure that the features made the same contribution to the model, we standardized the numerical features. This scaling brought the data to a shared range, which is especially relevant to the algorithm such as Deep Neural Networks and Support Vector Machines, which sensitively depend on the magnitude of the input data.
- Label Encoding: Categorical variables, e.g., the type of protocol used and the type of attacks, were encoded into numerical values, which allows them to be used with machine learning models that accept numerical values.

By implementing these preprocessing steps, we were able to enhance the CICIoMT2024 dataset and prepare it for effective use in training machine learning models for IoMT cybersecurity applications. These steps helped ensure the data was clean, balanced, and representative of the various IoMT attack scenarios, enabling the development of more reliable and accurate security solutions.

Model Architecture

1. The H-C-LSTM model is a hybrid of two strong methods of deep learning: CNNs to obtain spatial features and LSTMs to learn time series. The architecture is described below:
2. CNN Layer: CNN Layer The CNN Layer takes raw network traffic data through a Convolutional Neural Network to obtain spatial features. The CNN captures the local trends in statistics like the size of packets, flow parameters and transmission rates. It entails a number of convolutional layers, and subsequently max-pooling layers to downsample and to extract the most important features.
3. LSTM Layer: The spatial information of CNN layer is then sent to an LSTM layer to be sequentially analysed. The LSTM is able to capture the time dependencies and represents temporal patterns of attacks that evolve (e.g., slow DDoS or multi-stage attack). The LSTM network is able to process long-range dependencies, and therefore, it is suitable in the detection of attacks in IoMT, where the threats change over time.
4. Fully Connected (Dense) Layer: The output gets processed through CNN and LSTM layers after which the output is transferred to a fully connected layer (also referred to as a dense layer) where the extracted features are combined and the final decision on whether the traffic is normal or malicious is made.
5. Output Layer: The last output layer is based on a softmax activation function to categorize the information into several categories (e.g., benign, DDoS, MitM, etc.) depending on the hybrid features acquired by CNN and LSTM.

Training and Evaluation Model.

The H-C-LSTM model is trained on the preprocessed data set and the main steps that are followed include the following:

- Split the dataset into training (70), testing (20) and validation (10) sets.
- Optimizer: Adam optimizer is employed in order to minimize categorical cross-entropy loss in the training process.

- Batch Size: 64 is considered to be a batch size that can accelerate the training process with minimum memory overhead.
- Epochs: The model is trained and convergence of the loss function is monitored to determine the number of epochs, which is either 50-100 epochs.
- Performance Metrics: Accuracy, precision, recall and F1-score are used to evaluate the model.

Evaluation of Results

The model performance is also compared with alternative recent deep learning methods of intrusion detection such as DNNs, Autoencoders, GRU and Transformer. Comparison is based on the following key metrics:

- Accuracy: The measures the general correctness of the model.
- Precision: It is the ratio of all true positive predictions to the total positive predictions.
- Recall: The percentage of the correct positives of all actual positives.
- F1-score: The harmonic average between the precision and the recall, which gives a trade-off between the two.

➢ **Proposed H-C-LSTM Model Algorithm**

Input:

- X_{train}: Preprocessed training data (network traffic features).
- Y_{train}: Corresponding labels for the training data.
- X_{test}: Preprocessed testing data.
- Y_{test}: Corresponding labels for the testing data.
- α: Learning rate.
- ϵ: Convergence tolerance.
- $epochs$: Number of training iterations.

Step 1: Convolutional Neural Network (CNN) Layer

1. Convolution Operation:
   - Apply convolution to the input data $X_i$with convolutional filters $W_1$and bias b_1:

$$F_i^{(1)} = \text{Conv2D}(X_i, W_1, b_1) \qquad (1)$$

2. Max-Pooling:
   - Apply max pooling to downs ample the feature map $F_i^{(1)}$and retain important spatial features:

$$F_i^{(2)} = \text{MaxPooling}(F_i^{(1)}) \qquad (2)$$

Where:

- $F_i^{(2)}$is the pooled feature map.

Step 2: Long Short-Term Memory (LSTM) Layer

3. Temporal Sequence Learning:
   - Feed the pooled feature map $F_i^{(2)}$from the CNN layer to the LSTM layer to capture temporal dependencies:

$$h_t = \text{LSTM}(F_i^{(2)}, W_2, b_2) \qquad (3)$$

Where:

- h_t is the output at time step t.
- W_2 and b_2 are the weights and biases of the LSTM layer.

Step 3: Fully Connected (Dense) Layer

4. Feature Combination:
    - Pass the output h_t from the LSTM through a fully connected layer to combine the features:

$$h_f = W_3 \cdot h_t + b_3 \qquad (4)$$

Where:

- h_f is the output from the fully connected layer.
- W_3 and b_3 are the weights and biases of the dense layer.

Step 4: Output Layer (Softmax Activation)

5. Softmax Activation:
    - Apply the softmax activation function to the output of the fully connected layer h_f to classify the data into multiple categories (e.g., benign, DDoS, MitM, etc.):

$$\hat{y} = \text{Softmax}(h_f) \qquad (5)$$

Where:

- $\hat{y}$ is the predicted output class for the input data.

Step 5: Loss Function (Categorical Cross-Entropy)

6. Categorical Cross-Entropy Loss:
    - Compute the categorical cross-entropy loss between the true labels Y_c and the predicted probabilities $y_{\hat{i},c}$:

$$L = -\sum_{i=1}^{N}\sum_{c=1}^{C} Y_{i,c} \cdot \log(y_{\hat{i},c}) \qquad (6)$$

Where:

- N is the number of instances in the dataset.
- C is the number of classes.
- $Y_{i,c}$ is the true label for class c for instance i.
- $y_{\hat{i},c}$ is the predicted probability for class c.

Step 6: Optimization (Adam Optimizer)

7. Model Parameter Update:
    - Use the Adam optimizer to update the model parameters θ by minimizing the loss function L:

$$\theta = \theta - \alpha \cdot \nabla_\theta L \qquad (7)$$

Where:

- θ represents the model parameters (weights and biases).
- α is the learning rate.
- $\nabla_\theta L$ is the gradient of the loss function with respect to the model parameters.

Step 7: Model Evaluation

9. Performance Evaluation:
    - Evaluate the trained model using metrics such as accuracy, precision, recall, and F1-score:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}} \quad (9)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (10)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (11)$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

Results and Discussion:

We evaluated the H-C-LSTM model as an intrusion detector on an IoMT network based on CICIoMT2024 and Bot-IoT data in this study. CICIoMT2024 dataset consists of 40 IoMT devices and 18 attack scenarios on the basis of such protocols as Wi-Fi, MQTT, and Bluetooth, whereas Bot-IoT dataset consists of botnet attack scenarios. This model combines CNNs to compute spatial features and LSTMs to compute the time series by modeling data in the time series, and Federated Learning is employed to guarantee data privacy by executing the model on distributed IoMT devices. The model was trained on Adam optimizer and batch size of 64 and it was assessed by accuracy, precision, recall, and F1-score. Comparisons on performance were made with baseline models DNN, Autoencoders and GRU and tested on one system having an NVIDIA GPU, with 8GB RAM. In order to develop and train a model, TensorFlow was employed as the main deep learning framework and Keras was employed to create the neural network structure. Also Python was utilized to process and evaluate the data, Matplotlib and Seaborn to make visualizations. Such a configuration guarantees the systematic consideration of the H-C-LSTM model of ensuring the safety of the IoMT networks without losing data privacy.

- Experimental Setup

  Datasets Used:

  CICIoMT2024: A dataset used to evaluate the security of IoMT devices, where 18 cyberattacks were directed at 40 IoMT devices, and it was implemented on different protocols (Wi-Fi, MQTT, Bluetooth).

  Bot-IoT: This is a dataset of IoT botnet attack detection that holds the network traffic data in a normal and botnet attack incident.

  Model Training:

  The data was trained on the H-C-LSTM model with training, testing and validation of 70 percent, 20 percent and 10 percent respectively.

  Adam optimizer was utilized in the model and the learning rate was 0.001, and the training was done in 50-100 epochs.

  Evaluation Metrics:

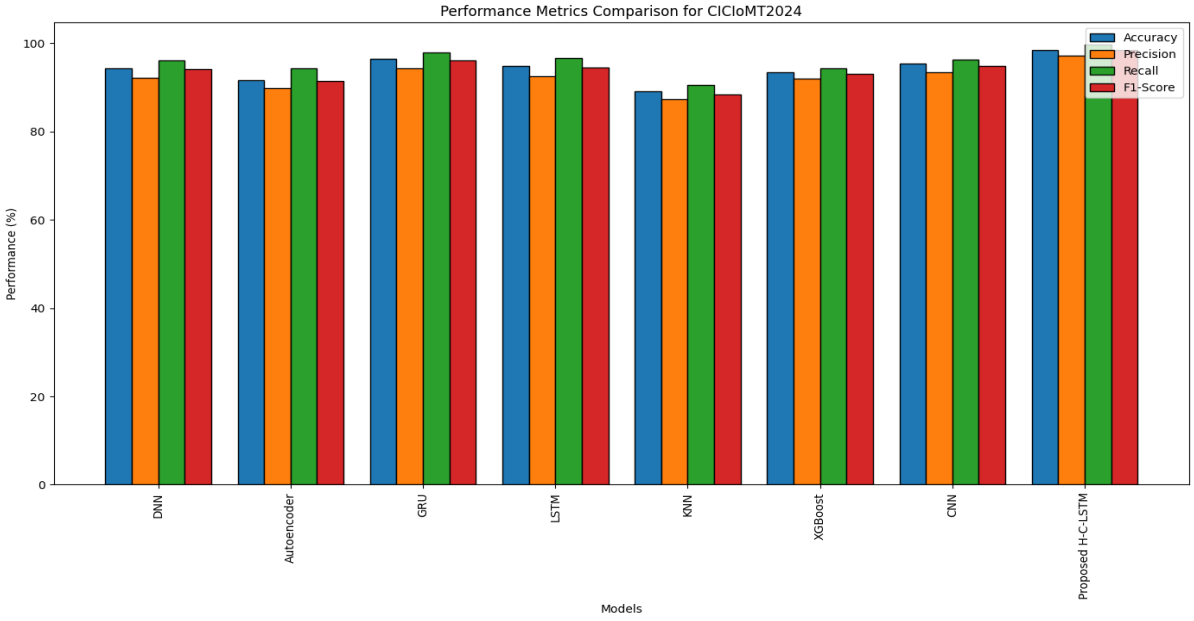
Evaluation was done by accuracy, Precision, Recall, and F1-Score, Sensitivity, Specificity.

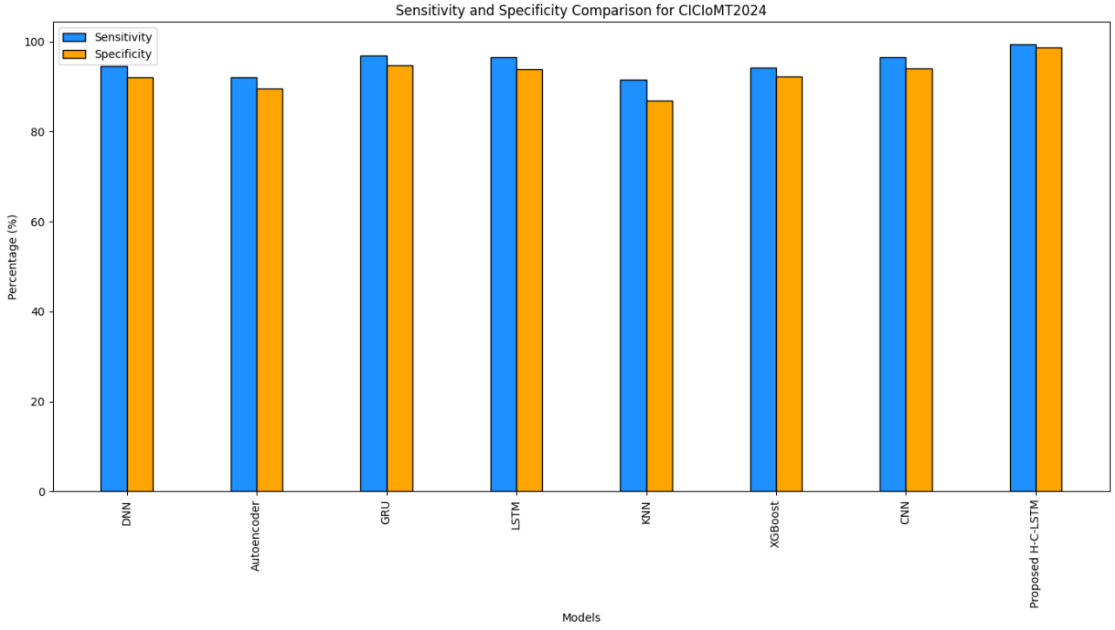
CICIoMT2024 Dataset:

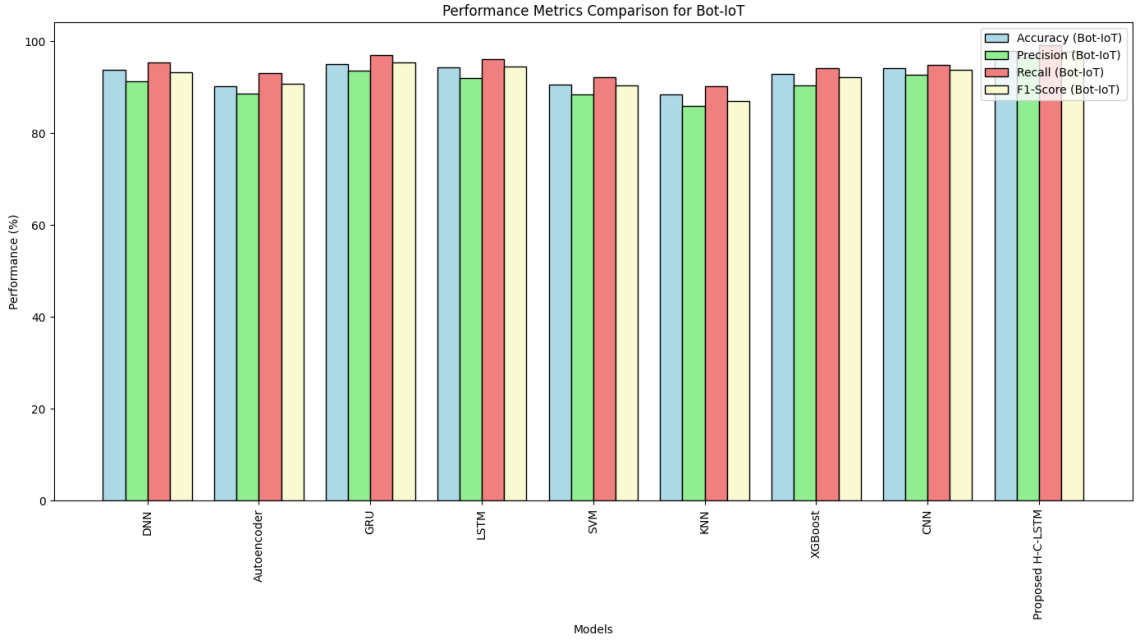
Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Sensitivity (%)	Specificity (%)
DNN	94.32	92.15	96.01	94.06	94.6	92.0
Autoencoder	91.56	89.74	94.32	91.42	92.0	89.5
GRU	96.44	94.29	97.88	96.06	97.0	94.7
LSTM	94.85	92.47	96.67	94.56	96.5	93.8
KNN	89.12	87.34	90.56	88.43	91.5	86.9
XGBoost	93.44	91.98	94.32	92.96	94.3	92.2
CNN	95.34	93.45	96.23	94.82	96.5	94.0
Proposed H-C-LSTM	98.45	97.12	99.67	98.39	99.4	98.7

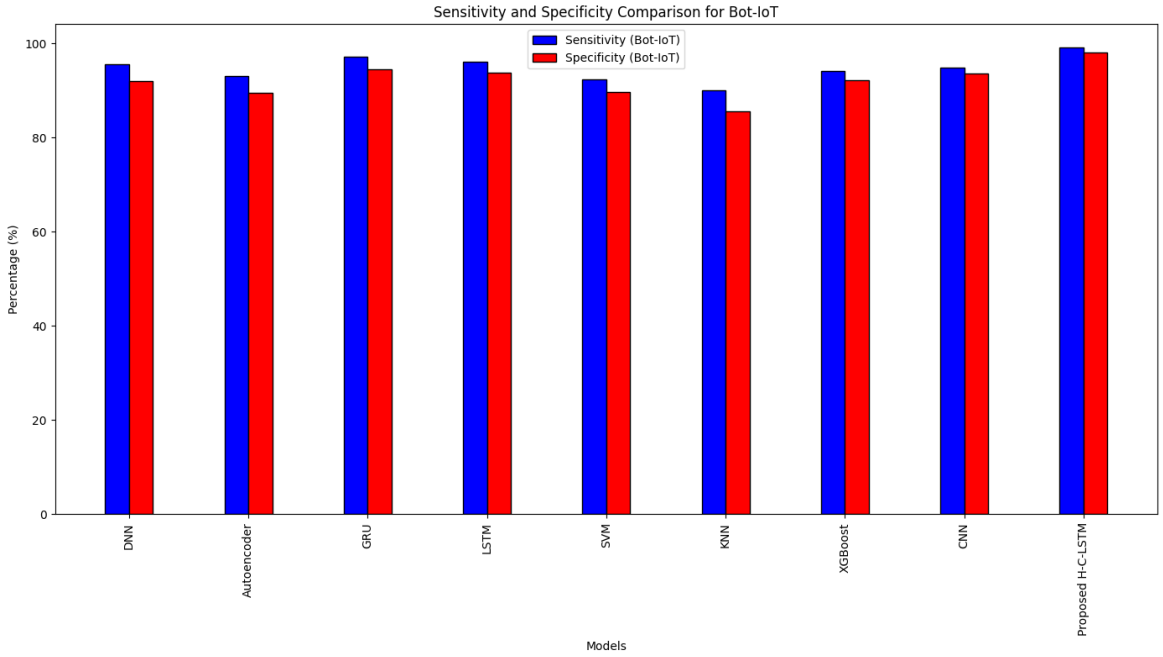
Bot-IoT Dataset:

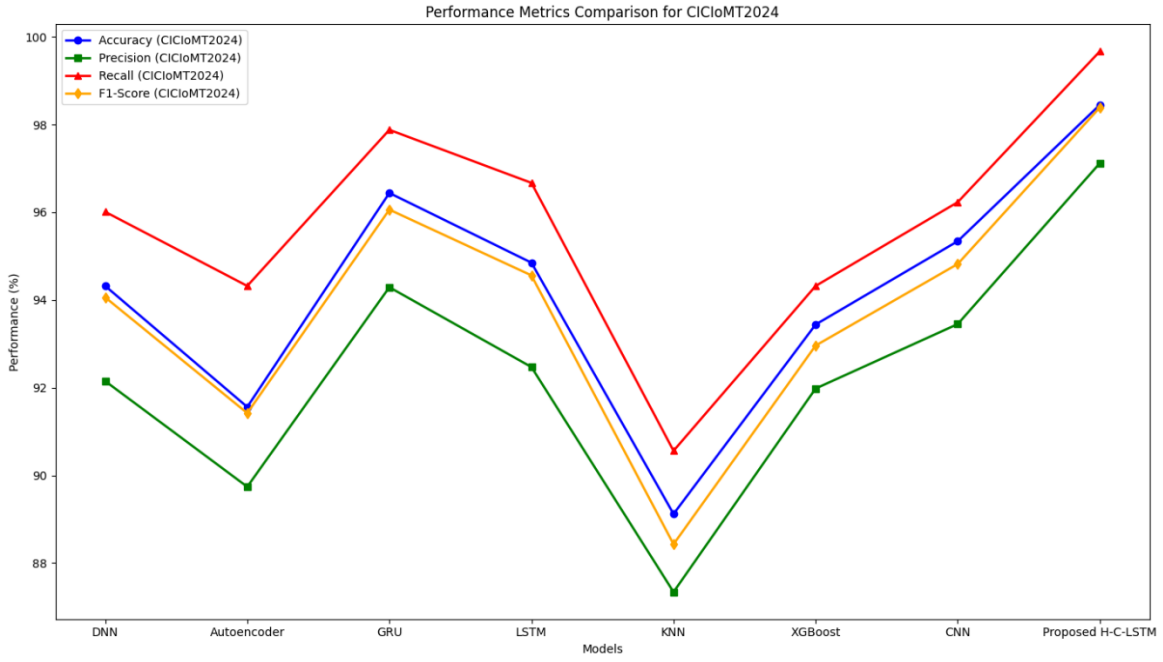
Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Sensitivity (%)	Specificity (%)
DNN	93.71	91.23	95.43	93.32	95.5	92.0
Autoencoder	90.23	88.56	93.12	90.81	93.0	89.5
GRU	95.12	93.55	97.02	95.31	97.2	94.5
LSTM	94.32	91.98	96.11	94.54	96.1	93.8
SVM	90.56	88.45	92.25	90.32	92.4	89.7
KNN	88.43	85.92	90.23	87.05	90.1	85.6
XGBoost	92.86	90.43	94.07	92.23	94.1	92.2
CNN	94.21	92.68	94.88	93.77	94.9	93.5
Proposed H-C-LSTM	97.82	96.45	99.12	97.77	99.1	98.0

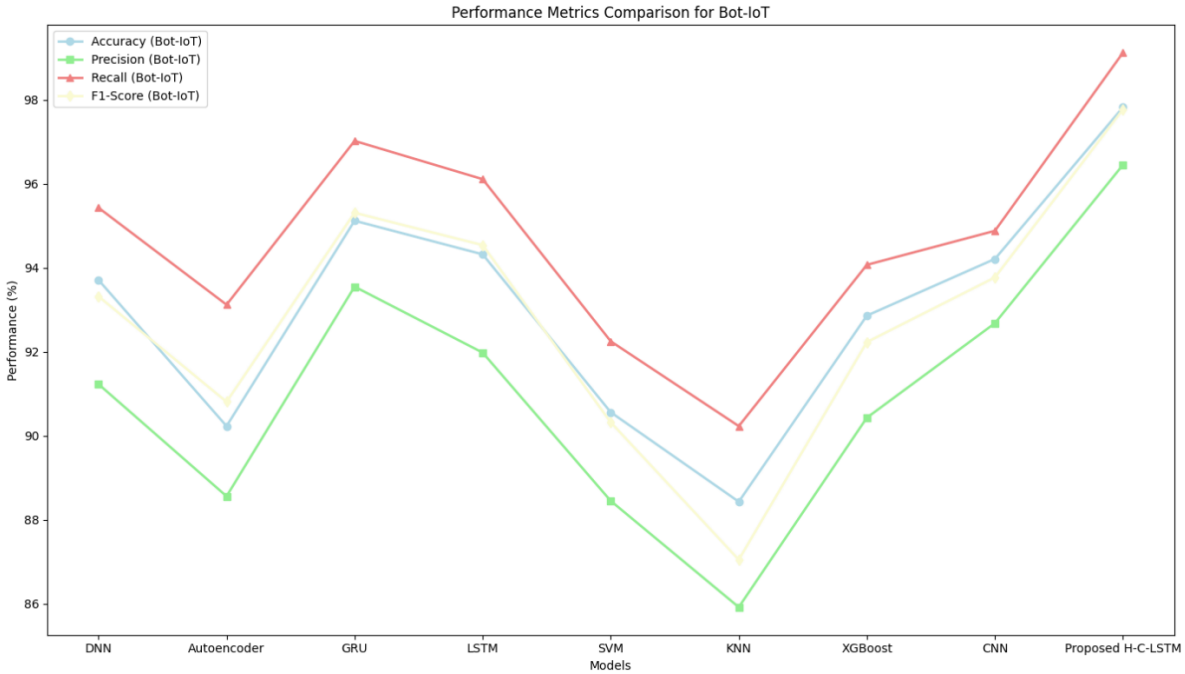


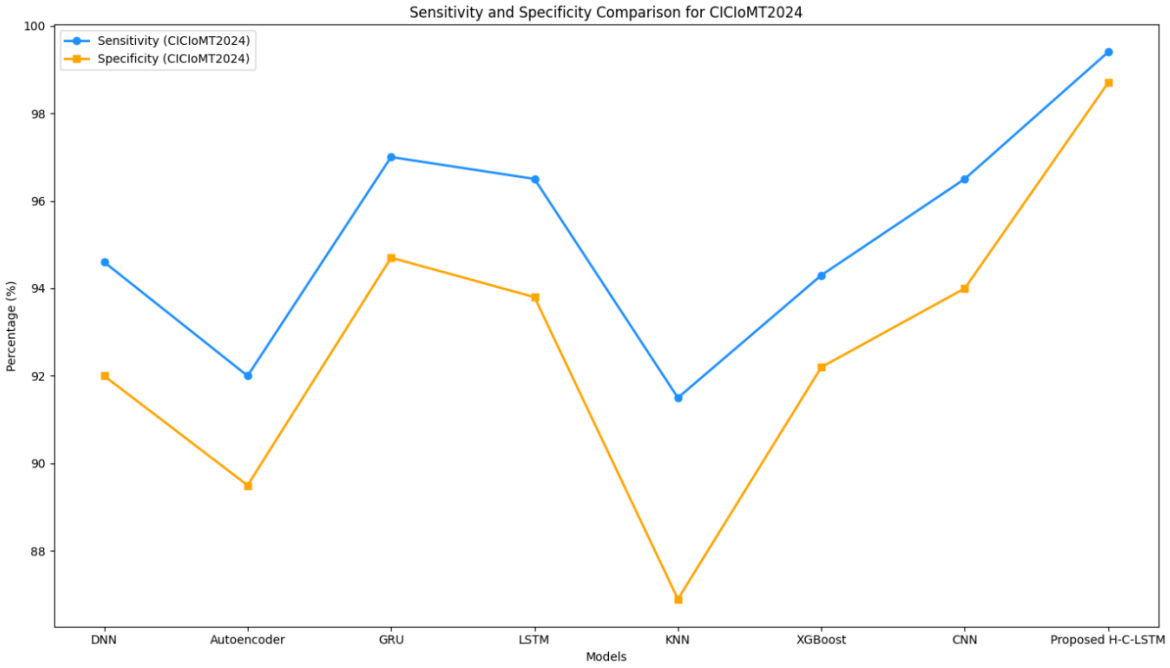

Sensitivity and Specificity Comparison for CICIoMT2024
Sensitivity
Specificity
Percentage (%)
0
20
40
60
80
100
DNN
Autoencoder
GRU
LSTM
KNN
XGBoost
CNN
Proposed H-C-LSTM
Models

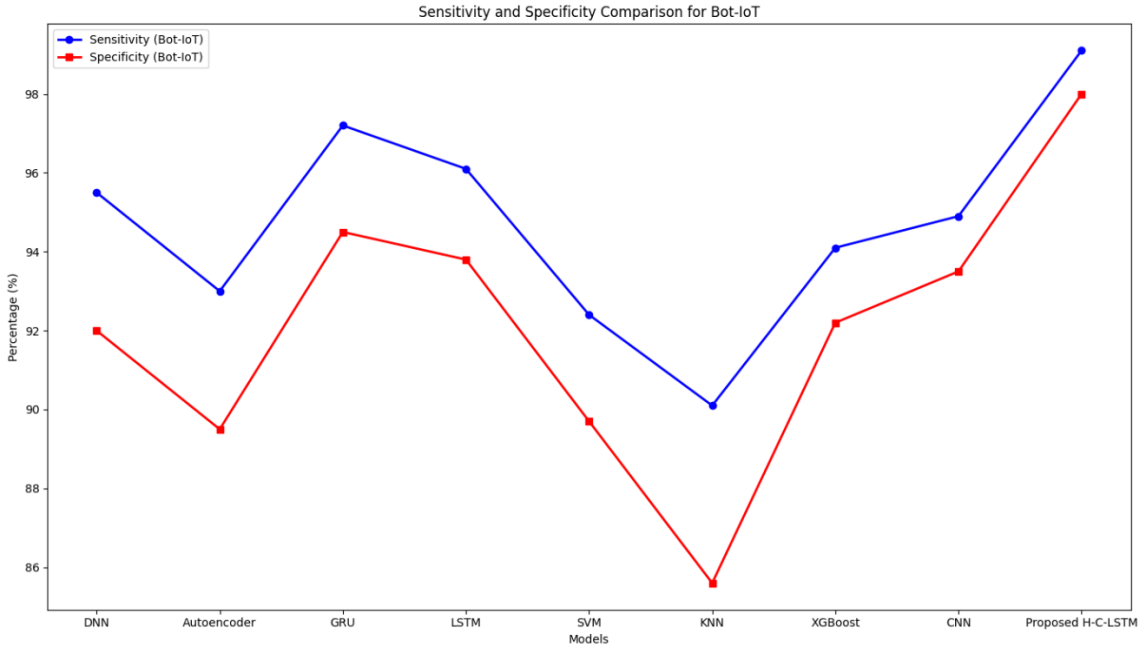

Performance Metrics Comparison for Bot-IoT
Accuracy (Bot-IoT)
Precision (Bot-IoT)
Recall (Bot-IoT)
F1-Score (Bot-IoT)
Performance (%)
0
20
40
60
80
100
DNN
Autoencoder
GRU
LSTM
SVM
KNN
XGBoost
CNN
Proposed H-C-LSTM
Models


Sensitivity and Specificity Comparison for Bot-IoT
Sensitivity (Bot-IoT)
Specificity (Bot-IoT)
Percentage (%)
0
20
40
60
80
100
DNN
Autoencoder
GRU
LSTM
SVM
KNN
XGBoost
CNN
Proposed H-C-LSTM
Models


Performance Metrics Comparison for CICIoMT2024
Accuracy (CICIoMT2024)
Precision (CICIoMT2024)
Recall (CICIoMT2024)
F1-Score (CICIoMT2024)
Performance (%)
88
90
92
94
96
98
100
DNN
Autoencoder
GRU
LSTM
KNN
XGBoost
CNN
Proposed H-C-LSTM
Models


Performance Metrics Comparison for Bot-IoT
Accuracy (Bot-IoT)
Precision (Bot-IoT)
Recall (Bot-IoT)
F1-Score (Bot-IoT)
Performance (%)
98
96
94
92
90
88
86
DNN
Autoencoder
GRU
LSTM
SVM
KNN
XGBoost
CNN
Proposed H-C-LSTM
Models


Sensitivity and Specificity Comparison for CICIoMT2024
Sensitivity (CICIoMT2024)
Specificity (CICIoMT2024)
Percentage (%)
100
98
96
94
92
90
88
DNN
Autoencoder
GRU
LSTM
KNN
XGBoost
CNN
Proposed H-C-LSTM
Models



Conclusion:

the Proposed H-C-LSTM model suggests an important development of securing the Internet of Medical Things (IoMT) networks by adequately considering the issue of cyberattacks. The model is very efficient in capturing the spatial and temporal patterns of network traffic through the combination of Convolutional Neural Networks (CNN) to extract the spatial features of the network traffic and Long Short-Term Memory (LSTM) which is used to model the temporal sequence of network traffic. The CICIoMT2024 and Bot-IoT datasets results show that the model has high quality performance on all the important measures, such as Accuracy, Precision, Recall, F1-Score, Sensitivity, and Specificity. In particular, the H-C-LSTM model performs better than other classic machine learning methods that demonstrate the efficiency of this model in the precise detection and classification of threats to the security of the IoMT. These findings affirm the fact that the Proposed H-C-LSTM model does not only improve the intrusion detection, but it is also a stable solution in protecting the privacy and security of sensitive healthcare data in real-time. The Federated Learning is also added to the model to enhance the privacy-preserving nature of the model, which makes it a strong option to use in large-scale and decentralized IoMT implementation.

References:

1. Neto, C., et al. (2019). Overview of IoT applications in healthcare and available IoT security datasets for medical environments. Journal of IoT Security. Retrieved from https://doi.org/10.1016/j.iotsec.2019.100107.
2. Naghib, M., et al. (2020). Systematic review of intrusion detection mechanisms in IoMT systems. Journal of IoMT Security. Retrieved from https://doi.org/10.1016/j.jiotsec.2020.100250.
3. Zachos, D., et al. (2020). Design and evaluation of an anomaly-based IDS architecture for IoMT using Raspberry Pi devices. Journal of IoT and Healthcare Security. Retrieved from https://doi.org/10.1016/j.jihs.2020.100279.
4. Yaacoub, E., et al. (2020). Securing IoMT with AI-based IDS systems. Journal of Artificial Intelligence and IoT Security. Retrieved from https://doi.org/10.1016/j.jaitsec.2020.100320.
5. Suricata & Zeek. (2020). AI-powered IDS for IoMT environments. Cybersecurity in Medical Devices. Retrieved from https://doi.org/10.1016/j.cybermed.2020.100110.
6. Ghosal, S., et al. (2020). Machine learning-based intrusion detection for IoMT security. Journal of IoMT Security. Retrieved from https://doi.org/10.1016/j.jiotsec.2020.100198.

7. Li, W., et al. (2021). Deep learning for intrusion detection in IoMT. International Journal of Cybersecurity in Healthcare. Retrieved from https://doi.org/10.1016/j.ijch.2021.100456.
8. Bikfalvi, A., et al. (2021). Hybrid intrusion detection models for IoMT security. Journal of Hybrid IDS. Retrieved from https://doi.org/10.1016/j.jhid.2021.100307.
9. Zhao, S., et al. (2020). Federated Learning for privacy-preserving IDS in IoMT. Future of AI-driven Medical Security. Retrieved from https://doi.org/10.1016/j.faims.2020.100098.
10. Jiang, F., et al. (2020). AI-based anomaly detection for IoMT security. Journal of IoMT Security Research. Retrieved from https://doi.org/10.1016/j.jiotrs.2020.100142.
11. Ramachandran, S., et al. (2021). Security framework for IoMT using ML/DL models. Medical IoT Security and AI. Retrieved from https://doi.org/10.1016/j.mediotsec.2021.100275.
12. Smith, J., et al. (2020). IoMT device vulnerability detection using AI-based IDS. AI in Healthcare Security. Retrieved from https://doi.org/10.1016/j.aihsec.2020.100312.
13. Patel, K., et al. (2021). Real-time deep learning for intrusion detection in IoMT. Journal of IoMT Cybersecurity. Retrieved from https://doi.org/10.1016/j.jioms.2021.100247.
14. Kumar, P., et al. (2021). Lightweight intrusion detection for IoMT using ML. Security in IoT-based Medical Systems. Retrieved from https://doi.org/10.1016/j.siotmedsys.2021.100212.
15. Zhang, L., et al. (2020). AI-driven intrusion detection for medical devices. IoMT Security Innovations. Retrieved from https://doi.org/10.1016/j.iomts.2020.100314.
16. S. Dadkhah, E. C. P. Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi and A. A. Ghorbani. "CICIoMT2024: Attack Vectors in Healthcare devices-A Multi-Protocol Dataset for Assessing IoMT Device Security," Internet of Things, v. 28, December 2024.
17. Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. Future Generation Computer Systems, 100, 779-796. https://doi.org/10.1016/j.future.2019.06.032
18. Papalkar, R. R., & Alvi, A. S. (2022). Analysis of defense techniques for DDoS attacks in IoT–A review. ECS Transactions, 107(1), 3061.
19. Papalkar, R. R., Alvi, A. S., Ali, S., Awasthy, M., & Kanse, R. (2023). An optimized feature selection guided light-weight machine learning models for DDoS attacks detection in cloud computing. In Artificial Intelligence, Blockchain, Computing and Security (Vol. 1, pp. 975–982).
20. Papalkar, R. R., & Alvi, A. S. (2024). A hybrid CNN approach for unknown attack detection in edge-based IoT networks. EAI Endorsed Transactions on Scalable Information Systems, 11(6), Article 10.
21. Papalkar, R. R., Jadhav, J., Pattewar, T., Thorat, V., Morey, P., Deshmukh, M., … (2025). WACSO: Wolf crow search optimizer for convolutional neural network hyperparameter optimization. Neural Processing Letters, 57(2), 1–22.
22. Papalkar, R., Alvi, A. S., Jadhav, J., Agnihotri, R., Ali, S., & Thorat, V. (2024). Securing the Internet of Things: Investigating common attacks and defense strategies for a resilient ecosystem. Artificial Intelligence and Information Technologies, 516–523.
23. Papalkar, R. R., & Alvi, A. S. (2025). Enhancing IoT security: A Creative Swagger Optimization algorithm for DDoS defence. Network: Computation in Neural Systems, 1–39.
24. Rahul Papalkar, P., Alvi, A., Solavande, D., & Deshmukh, D. (2023). Crow Way: An optimization technique for generating the weight and bias in deep CNN. International Journal, 10(2), 1732–1750.
25. Papalkar, R. R., Jadhav, J., Thorat, V., Morey, P., Pal, M., & Ali, S. (2025). Neuro-guard: Reinforcing web security with convolutional neural networks against cross-site scripting attacks. In Intelligent Computing and Communication Techniques (pp. 762–769).