electronics

IMPACT FACTOR 2.6

CITESCORE 6.1

Article

SafetyMed: A Novel IoMT Intrusion Detection System Using CNN-LSTM Hybridization

Nuruzzaman Faruqui, Mohammad Abu Yousuf, Md Whaiduzzaman, AKM Azad, Salem A. Alyami, Pietro Liò, Muhammad Ashad Kabir and Mohammad Ali Moni

Special Issue

Data Analytics and Visualization in Health Informatics

Edited by

Dr. Ashad Kabir, Dr. Shariful Islam, Dr. Enamul Hoque, Dr. Mufti Mahmud and Dr. Naeemul Hassan

MDPI

https://doi.org/10.3390/electronics12173541

electronics

MDPI

Article

SafetyMed: A Novel IoMT Intrusion Detection System Using CNN-LSTM Hybridization

Nuruzzaman Faruqui [1,2], Mohammad Abu Yousuf [2,*], Md Whaiduzzaman [3], AKM Azad [4], Salem A. Alyami [5], Pietro Liò [6], Muhammad Ashad Kabir [7] and Mohammad Ali Moni [8,9,*]

1 Department of Software Engineering, Daffodil International University, Daffodil Smart City, Birulia, Dhaka 1216, Bangladesh; faruqui.swe@diu.edu.bd
2 Institute of Information Technology, Jahangirnagar University, Savar, Dhaka 1342, Bangladesh
3 Faculty of Science, School of Information Systems, Queensland University of Technology, 2 George St., Brisbane, QLD 4000, Australia; md.whaiduzzaman@qut.edu.au
4 Department of Mathematics and Statistics, Faculty of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia; kazad@imamu.edu.sa
5 Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11564, Saudi Arabia; saalyami@imamu.edu.sa
6 Department of Computer Science and Technology, The University of Cambridge, Cambridge CB2 1TN, UK; pl219@cam.ac.uk
7 School of Computing, Mathematics, and Engineering, Charles Sturt University, Bathurst, NSW 2795, Australia; akabir@csu.edu.au
8 Artificial Intelligence & Data Science, School of Health and Rehabilitation Sciences, Faculty of Health and Behavioural Sciences, The University of Queensland, St Lucia, Brisbane, QLD 4072, Australia
9 Artificial Intelligence and Cyber Futures Institute, Charles Stuart University, Bathurst, NSW 2795, Australia
* Correspondence: yousuf@juniv.edu (M.A.Y.); m.moni@uq.edu.au or mmoni@csu.edu.au (M.A.M.)

Abstract: The Internet of Medical Things (IoMT) has become an attractive playground to cybercriminals because of its market worth and rapid growth. These devices have limited computational capabilities, which ensure minimum power absorption. Moreover, the manufacturers use simplified architecture to offer a competitive price in the market. As a result, IoMTs cannot employ advanced security algorithms to defend against cyber-attacks. IoMT has become easy prey for cybercriminals due to its access to valuable data and the rapidly expanding market, as well as being comparatively easier to exploit.As a result, the intrusion rate in IoMT is experiencing a surge. This paper proposes a novel Intrusion Detection System (IDS), namely SafetyMed, combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to defend against intrusion from sequential and grid data. SafetyMed is the first IDS that protects IoMT devices from malicious image data and sequential network traffic. This innovative IDS ensures an optimized detection rate by trade-off between False Positive Rate (FPR) and Detection Rate (DR). It detects intrusions with an average accuracy of 97.63% with average precision and recall, and has an F1-score of 98.47%, 97%, and 97.73%, respectively. In summary, SafetyMed has the potential to revolutionize many vulnerable sectors (e.g., medical) by ensuring maximum protection against IoMT intrusion.

Keywords: internet of medical things; intrusion detection system; convolutional neural network; long short-term memory; response mechanism; IoMT; IDS; CNN; LSTM

Citation: Faruqui, N.; Yousuf, M.A.; Whaiduzzaman, M.; Azad, AKM; Alyami, S.A.; Liò, P.; Kabir, M.A.; Moni, M.A. SafetyMed: A Novel IoMT Intrusion Detection System Using CNN-LSTM Hybridization. *Electronics* **2023**, *12*, 3541. https://doi.org/10.3390/electronics12173541

Academic Editor: Manohar Das

Received: 27 May 2023
Revised: 9 August 2023
Accepted: 14 August 2023
Published: 22 August 2023

1. Introduction

The Internet of Medical Things (IoMT) is a sophisticated network that seamlessly integrates Internet-connected medical devices and corresponding software applications to exchange healthcare-related information to facilitate treatment and patient observation over the Internet [1]. The global IoMT market will experience a Compound Annual Growth Rate (CAGR) of 18.5% between 2021 and 2027, resulting in a valuation of US $284.5 billion [2]. Moreover, the value of IoMT data is considered 50 times more than data of other sectors [3].

At the same time, IoMTs are resource-constraint devices. That is why the quality of the utility gets higher priority than the advanced security by the manufacturer [4]. The rapid market growth, data value, and security vulnerabilities attract cybercriminals to exploit the weaknesses of IoMT devices. This paper proposes SafetyMed, a novel Intrusion Detection System (IDS) that ensures the integrity, confidentiality, and security of IoMT-enabled services, facilitating safer and more reliable personalized healthcare services.

SafetyMed employs a groundbreaking LSTM-CNN hybrid architecture specifically designed for intrusion detection. This distinctive architecture enables the system to proactively defend against intrusions using both image and non-image data. The hybrid model has been meticulously crafted through rigorous mathematical analysis, ensuring the conceptual integrity of the system. The revolutionary design of SafetyMed incorporates an edge server interposed between the IoMT network and the Internet access point. Furthermore, it houses a highly effective classification algorithm capable of identifying the twelve most commonly occurring IoMT intrusions. To the best of our knowledge, this surpasses the capabilities of any existing IDS in terms of the number of detectable intrusions. Half of these intrusions are non-images, with the remainder being image-based. SafetyMed is designed with the aim of detecting intrusions with substantially high accuracy, thereby safeguarding the IoMT nodes from subsequent intrusions of a similar nature. In summary, SafetyMed is a unique IDS offering both intrusion detection and prevention and boasts the following novel features:

- Architectural Novelty:The proposed SafetyMed is the first of its kind of IDS, to our best knowledge, hybridizing the CNN and LSTM to defend against intrusion from sequential and grid-structured data.
- Innovative Optimization Scheme: The SafetyMed is the first IDS that incorporates an optimization scheme that uses the trade-off between Detection Rate (DR) and False Positive Rate (FPR).
- Effective Application: Unlike most IDSs studied in the literature review, SafetyMed employs an additional layer of protection by detecting and preventing further malicious traffic from compromised sources.
- Unique Classification Algorithm: A unique classification algorithm using the SafetyMed technology has been developed in this paper, contributing to the proposed IDS's outstanding performance.

The remainder of this paper is organized as follows. Section 2 presents existing research and its limitations. Section 3 outlines the methodology employed in the study, detailing the various steps and techniques used. Section 4 covers the implementation of the proposed SafetyMed system and analyzes its response mechanism. Section 5 provides a performance evaluation of SafetyMed, considering different contexts and scenarios. Section 6 discusses the paper's limitations and explores potential avenues for future research. Finally, Section 7 concludes the paper, summarizing the key findings and contributions to the field.

2. Related Work

The IoMT threat detection system with the explainable model developed by I. A. Khan et al. [5] supports the problem statement explored in this paper. A survey by S. Ahmed et al. [6] on Machine Learning (ML)-based intrusion detection systems shows the dominance of text-based IDSs. After detecting intrusion in the IoMT network, the decision-making seems to be gaining little attention. The proposed IDS pays equal attention to intrusion detection and its application. The article by I. Idrissi et al. [7], F. Khan et al. [8], and S. A. Wagan et al. [9] agrees with the findings of this paper, stating that IoMT intrusion is rising. These studies focus on the development of the IDS, enhancing their performances. However, the improved detection rate is underutilized in these papers. At the same time, these papers have not analyzed the detection rate and delay. The proposed paper addresses these issues, enhances the performance of the IDS, and focuses on detection rate and detection time. This is how SafetyMed stands out from the rest of the IDSs.

A. K. Kumar et al. [10] propose an IoMT intrusion detection system by hybridizing CNN, BiLSTM, and Gated Recurrent Units (GRU). While this methodology achieves 98.34% accuracy, it detects only Botnet attacks. The Botnet attacks are usually made through sequential data [11]. An LSTM or BiLSTM network is enough to detect this attack accurately [12]. CNNs are designed for grid-like data structures representing images [13]. This observation suggests that hybridizing with CNN does not serve the purpose but introduces additional network complexity. The proposed SafetyMed has adopted the idea of hybridizing CNN and LSTM networks, making the extended capabilities productive. Including Botnet attacks, this paper detects eleven other types of attacks. This is the first IDS capable of detecting these many intrusions. It utilizes the CNN for detecting intrusions through a grid data structure, which is missing in the paper of A. K. Kumar et al. [10]. At the same time, SafetyMed uses LSTM to detect six types of intrusions from the sequential, where A. K. Kumar et al. [10] detected only Botnet attacks.

R. Chitra [14] developed the XGBoost classifier-based malware detection system of IoMT, which achieves 97% accuracy. However, this study did not develop any response mechanism after developing the classifier to take appropriate action. SafetyMed proposes an effective algorithm to defend the intrusion after detection and prevent further attacks from the malicious source. This classifier is optimized with Genetics Algorithm (GA), which optimizes the learning process. However, they did not optimize the detection rate to reduce the false alarm rate, which has been conducted in our proposed system. S. Karagiannis et al. [15] conducted a study on developing mobile applications to analyze the security vulnerability in IoMT. It is a unique research. However, it does focus on intrusion detection and protecting the IoMT devices, whereas SafetyMed is dedicated to protecting IoMT devices. An innovative experience-driven threat-defense system developed by B. Tahir et al. [16] addresses the impact of False Data Injection Attacks (FDIA). While it accurately defends this attack, the other frequently attempted attacks are ignored, leaving the system vulnerable to other frequently attempted attacks. The proposed IDS considers the risk of being exploited by various other types of intrusions. That is why it has been designed to defend against twelve intrusions.

The Gradient-Boosted-Tree-based IoMT intrusion detection system developed by W. Lu [17] classifies intrusions with 95.4% accuracy. However, it has been trained using 11 features only, leaving a question mark on the system's reliability. The intrusion pattern rapidly changes because of the availability of high-performance computing [18]. A system trained with 11 features cannot detect complicated patterns embedded in non-malicious data. The proposed SatefyMed has been trained on 78 non-image and numerous image features, which is more than 97% accurate. Because of using the CNN-LSTM hybrid network, it is capable of identifying intrusions from complex patterns, ensuring better security. An explainable DL framework for Industrial IoT (IIoT) developed by I. A. Khan et al. [19] uses Fully Connected (FC) architecture and segments the data sequence using the Sliding Window (SW) method and converts them into fixed sequences. This methodology agrees with the approach introduced in this paper. The Swarm-Neural Network-based IoMT intrusion detection system developed by J. B. Awotunde et al. [20] secures health-related data by incorporating an edge server. The proposed IDS uses an edge server as well but with well-designed IDS architecture with an aim to outperform these methods.

Research Gaps

A thorough review of the literature reveals a significant research gap in intrusion detection systems based on image data. The preponderance of existing studies concentrates on intrusion detection within non-image data. Contrary to this trend, our proposed system, SafetyMed, distinguishes itself by tackling both image and non-image data. Remote patient monitoring is a prevalent service within the Internet of Medical Things (IoMT) domain. The vulnerability of these imaging device-based systems to intrusion via malware-infused images presents a substantial risk. Additionally, the transmission of medical imaging reports through IoMT services exposes another potential area of vulnerability [6]. Our

proposed intrusion detection system proactively identifies and mitigates these threats by detecting intrusions from both image and non-image data sources.

Over-sensitive IDS generates false alarms even if the Machine Learning (ML) model exhibits high accuracy. Blocking network traffic based on false positive prediction interrupts the seamless IoMT communication. It also degrades the service quality. Emphasizing the model's accuracy and ignoring its usability is a significant research gap in the existing literature [21]. Training and optimizing ML models to improve IDS's performance draws the attention of the recently published papers. Developing sophisticated network architectures, incorporating various optimization algorithms, and introducing more effective features highlight the current IDS research trends [22]. Using the prediction to develop reliable and usable IDS is a research gap explored in the proposed SafetyMed. It abridges the gap by devising a novel algorithm called SafetyMed Classification (SC) to utilize the predictions from the system and secure IoMT devices by defending current and future intrusions.

3. Methodology

The overview of the proposed methodology is illustrated in Figure 1. It is the simplified process of the overall system, where Figure 1a presents the network training. It involves network selection, dataset processing, and training phases. Figure 1b is the simplified SafetyMed overview that runs in an edge server. Finally, Figure 1c presents the communication module where the router communicates with the internet. A Router Access Control (RAC) unit is controlled by the SafetyMed Classification (SC) algorithm. The algorithm detects the intrusion and protects the IoMT devices by blocking further packets from malicious sources through the RAC.

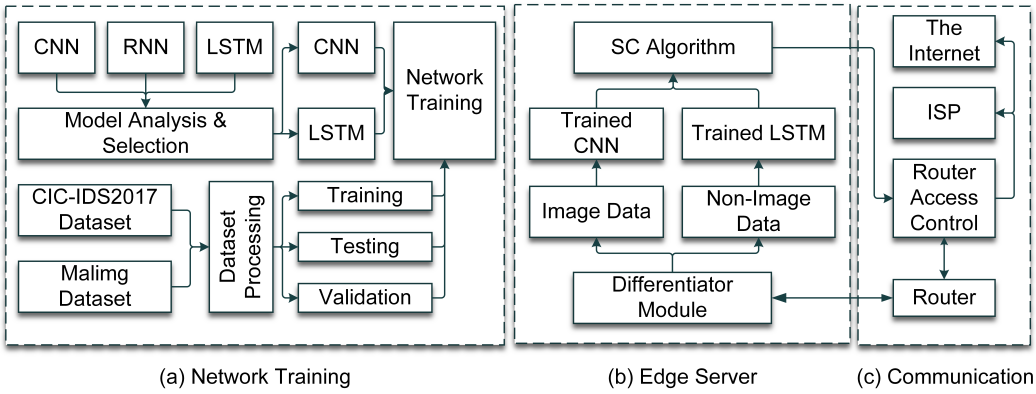

Figure 1. The overview of the methodology.

3.1. Deep Learning Model Analysis and Selection

Deep Learning (DL) technology uses the Multilayer Artificial Neural Network (ANN) algorithm [23]. As a result, it can learn complex patterns from different data and make accurate predictions. IoMT receives complex network data where numerous parameters and their values are combined. The data pattern is complex; thus, DL models are perfect for intrusion detection at the IoMT node. However, hundreds of DL models and more are frequently added. This rapid development in DL technology has made choosing an appropriate model challenging.

3.1.1. Model Analysis

We have performed an exploratory analysis to identify the most effective DL models to detect intrusions at the IoMT node. The findings of our analysis show that the Convolutional Neural Networks (CNNs) [24], Recurrent Neural Networks (RNNs) [25], and Long Short-Term Memory (LSTM) networks [26] are more appropriate than other models for intrusion detection. Despite the potential scope of application, the Bidirectional Encoder

Representations from Transformers (BERT) developed by J. Devlin et al. [27] is a hybrid architecture that combines CNN, RNN, and other transformer models, which have not been considered. It is an excellent approach to trained models from unlabelled text for Natural Language Processing (NLP) applications. However, the dataset used in this paper is labelled and does not contain natural language. That is why it has not been finalized for analysis. This section presents the logical argument with supportive evidence for finalizing these three models.

Convolutional Neural Networks (CNNs)

IoMT devices communicate textual, numerical, and image data [28]. Stegomalware has become an emerging challenge in Cybersecurity because of its characteristics. As IoMT involves image data, it is essential to implement an intrusion detection model optimized for image data. According to F. Österlind et al. [24], CNNs are specially designed for images. They are efficient in image feature extraction and classification. Image data are represented as grid data structures in the network traffic. CNNs are effective in classifying grid data [29]. A typical CNN follows the mathematical model defined in Equation (1).

$$Y_{nm} = (X * K)_{nm} = \sum_{p} \sum_{q} X_{(n+p)(m+q)} K_{pq} \tag{1}$$

According to Equation (1), the hidden layers perform convolution operations, denoted by X, that extract the image features. It is followed by a pooling layer that reduces the image size. The K is the kernel of $q \times p$, and Y_{nm} is the output feature map at position (n, m). Finally, the fully connected layer learns to classify images from the features extracted through the convolutional operation. The mathematical structure of CNNs shows that they learn hierarchical image features automatically. The network intrusion patterns frequently change, and retraining the model is essential to protect the IoMT nodes. This is where the CNNs become effective, as they excel at extracting features and learning from them. This is the reason for selecting CNN as one of the Deep Learning models for intrusion detection.

Recurrent Neural Networks (RNNs)

Cybercriminals send frequent small data sequences as network packets to exploit the vulnerability of IoMT devices. A review on RNN conducted by Y. Yu et al. suggests that RNN suffers from performance issues for longer sequences. However, its performance for shorter sequences is impressive [30]. This characteristic implies that the RNN effectively classifies small intrusion data sequences. However, malicious code embedded into long data sequences may not be accurately detected by RNN. Mathematically, the RNN is defined as Equation (2).

$$h_t = \sigma(W_{hh} h_{t-1} + W_{xh} x_t + b_h) \tag{2}$$

$$y_t = W_{hy} h_t + b_y \tag{3}$$

In Equation (2), x_t is the input time step at t. The hidden state is denoted by h_t. The output is y_t. The weight matrices are defined by W_{hh}, W_{xh}, and W_{hy}. The final output from the RNN is processed by a non-linear hyperbolic tangent denoted by $\sigma(\cdot)$, which is defined by Equation (4), where (4), $x = W_{hh} h_{t-1} + W_{xh} x_t + b_h$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4}$$

The typical RNN models effectively identify malicious short sequences from the network traffic. That is why this experiment has selected them as one of the DL models.

Long Short-Term Memory (LSTM) Networks

The malicious network traffic comes in both short and long sequences. RNN is effective in identifying short sequences. Moreover, LSTM networks perform well in classifying long

data sequences [31]. It is a subset of RNN with other memory cells and three gating mechanisms. The first gate of the LSTM network is the input gate defined by (6). The second gate is the forget gate expressed in Equation (5), and the last gate is the output gate modelled as Equation (7).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{5}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{6}$$

$$o_t = sigma(W_o[h_{t-1}, x_t] + b_o) \tag{7}$$

In Equations (5)–(7), the output gate is denoted by o_t, the forget gate is f_t, and the input gate is i_t. These gates control the information flow. Relevant information and dependencies are stored in the memory cell longer using these gates. If necessary, data are erased or replaced [32]. Because of having control over the memory cell, LSTM demonstrates an outstanding performance in classifying long sequences. The memory cells are updated according to Equation (8).

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{8}$$

After updating, it is necessary to check the status of the cells. This is performed by Equation (8).

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{9}$$

The comprehensive design of LSTM combining memory cells and control gates, defined by Equation (10), enables it to handle long data sequences while maintaining an performance. This is the reason behind choosing LSTM as one of the Deep Learning models for IoMT intrusion detection.

$$h_t = o_t \odot \tanh(C_t) \tag{10}$$

These deep learning techniques can be adapted and fine-tuned for intrusion detection tasks, taking advantage of their unique capabilities to process and learn from complex network data.

3.1.2. Model Selection

IoMT data refers to the health-related data of the patients. According to B. Bhushan et al. [3], the average cost of IoMT data is precious, which is considered to be more than 50 times higher than data from other sectors. This is one of the fundamental reasons behind the recent rise in IoMT intrusion attempts [33]. It has been observed that the IoMT suffers from similar attacks, which are attempted on IoT devices. The application domain of IoMT is exclusive. However, the fundamental development structure and communication protocols are identical [34]. That is why the intrusion common to IoT devices also threatens IoMT devices. These intrusions appear in short sequences, long sequences and sometimes are embedded in image data. Figure 2 illustrates a Venn diagram that shows the overlapping characteristics of the DL models we explored.

CNN is exclusive to malicious image data. The RNN covers most short sequences and a marginal portion of the long sequences. However, the LSTM network is effective for both short and long sequences. IoMT devices' response time should be fast because they need to process signals in real-time [35]. Using too many deep learning models will create multilevel filtering, which introduces an additional processing delay. That is why CNN and LSTM have been selected as DL models to detect intrusion in IoMT. The Equation (11) shows that CNN's role is exclusive in this experiment.

$$\begin{aligned}(CNN(\text{image}) \cap RNN(\text{short})) \cup \\ (CNN(\text{image}) \cap RNN(\text{long})) \cup \\ (CNN(\text{image}) \cap LSTM(\text{long})) \equiv \varnothing\end{aligned} \tag{11}$$

The equivalency (12) shows that RNN and LSTM are mutually inclusive. That is why their union is logically equivalent to either RNN or LSTM. However, the scenario is different for long sequences. According to Equation (13), the intersection between RNN and LSTM is logically equivalent to LSTM. That is why LSTM is enough to detect intrusion from both short and long sequences.

$$\begin{aligned}RNN(short) \cup LSTM(short) &\equiv RNN(short) \\ &\equiv LSTM(short)\end{aligned} \tag{12}$$

$$RNN(long) \cap LSTM(long) \equiv LSTM(short) \tag{13}$$

After exploring the characteristics of the experimenting DL models, the CNN and the LSTM network have been selected to implement the proposed IoMT intrusion detection system.

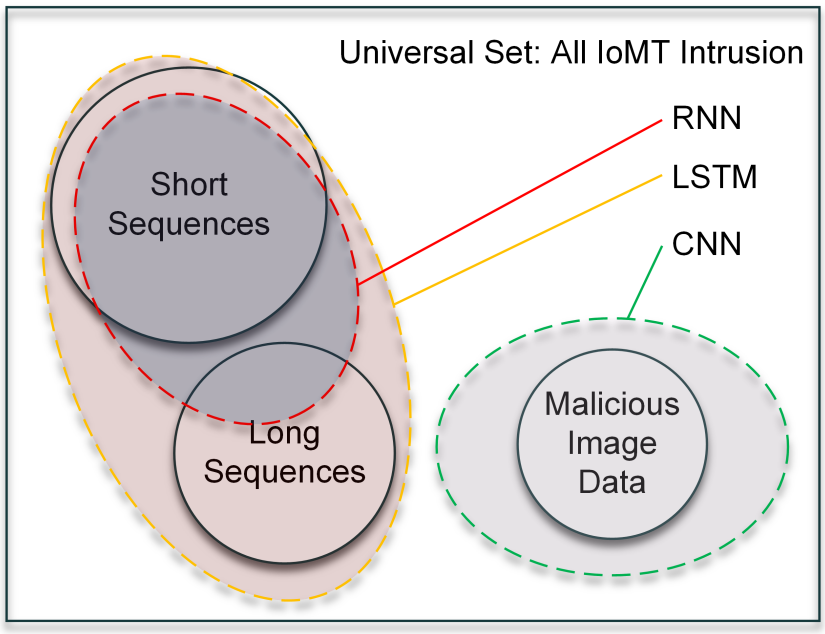

Figure 2. Model selection criteria using Venn Diagram.

3.2. Dataset Description and Processing

Deep Learning is a data-driven technology [36]. Without an adequate amount of feature-rich datasets, the models suffer from performance issues. Usually, the raw dataset contains duplicate data, missing values, outliers, and other quality-limiting issues [37]. That is why data pre-processing is an essential step in training DL models. This section presents the dataset description and pre-processing methods.

3.2.1. Textual Dataset Description and Cleaning

The Canadian Institute for Cybersecurity has collected different types of attacks on the Internet of Things (IoT) and published the CIC-IDS2017 dataset in 2017 [38]. Since then, this dataset has been frequently updated. That is why it is considered one of the most convenient datasets to train an intrusion detection system [39]. The types of attacks on IoT and IoMT are the same. During the time of conducting this experiment, there were 2.8 million instances in this dataset. There are multiple duplicate values, outliers, and missing values in the original dataset. It requires cleaning before using it.

Dataset Cleaning

The cleaning process of the CIC-IDS2017 dataset involves handling missing values, removing duplicates, and removing outliers. In the beginning, the dataset D has been defined by Equation (14), where N is the total number of instances.

$$D = \{d_1, d_2, \ldots, d_N\} \tag{14}$$

In the dataset, D, multiple missing values M_i for ith features exist. The missing values have been calculated using non-missing values using Equation (15), where μ_i is the replacement for the missing value of the ith feature.

$$\mu_i = \frac{1}{N - |M_i|} \sum_{d_j \in D \setminus M_i} d_{j,i} \tag{15}$$

Missing values after being updated after calculating them. The updating process follows the mathematical model of Equation (16).

$$d_{j,i} = \begin{cases} \mu_i, & \text{if } d_j \in M_i \\ d_{j,i}, & \text{otherwise} \end{cases} \tag{16}$$

After handling the missing values, the duplicate values were identified and removed. Equation (17) demonstrates the mathematical approach to managing duplicate values. Here D^* is the dataset after handling duplicate values. D' is the set of unique instances in D^*.

$$D' = \{d \in D^* \mid \nexists d' \in D^* \setminus \{d\}, d = d'\} \tag{17}$$

Numerous outliers in the dataset have been removed through visual inspection using different functionalities of Microsoft Excel.

Dataset Splitting

There are 56,660 instances in the dataset after cleaning it. This dataset has been split into training, testing, and validation by maintaining 70:15:15. After splitting, there are 39,662 instances for training. Both testing and validation datasets have 8499 instances. The training dataset has been used to train the network. The validation dataset was used to validate the training data performance during the training. The testing dataset is untouched during training and validation. It has been used to test the proposed IoMT intrusion detection performance after training the model.

Variables and Sample

The CIC-IDS2017 is one of the most prominent datasets for IoMT intrusion detection research. One of the reasons behind it is its 78 features. These 78 features are mapped to fifteen classes [38]. Out of these fifteen classes, one is benign, and the remaining fourteen are different types of attacks. One snippet of this dataset is listed in Table 1. It is beyond the scope of tabulating every sample variable because of space constraints. That is why five prominent features have been presented in the sample. These samples are Destination Port (Dst Port), Flow Duration (Flow Duration), Total Fwd Packets (Tot Fwd Pkts), Total Backward Packets (Tot Bwd Pkts), and Fwd Packet Length Mean (Fwd Pkt Len Mean). Not every target variable of the CIC-IDS2017 dataset applies to IoMT. DoS slowloris, DoS Slowhttptest, DoS Hulk, DoS GoldenEye, and DDoS are the frequently occurring attacks on the availability of IoMT [40–42]. The web-based interface of the IoMT services is usually under Web Attack-Brute Force, Web Attack-XSS, and Web Attack-SQL Injection. The most frequently occurring attack to gain unauthorized access to IoMT devices is the Infiltration attack. Many attackers use Bot attacks to automate their primary attacks. Although portscan is not a direct attack, it is also considered a threat to the IoMT device. This is because the attackers find the vulnerability to gain unauthorized access through open ports using PortScan.

Table 1. A simplified sample of the CIC-IDS2017 Dataset.

Dst Port	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	Fwd Pkt Len Mean	Label
80	1,545,689	2	1	274.5	BENIGN
443	4,790,712	15	11	128.667	FTP-Patator
80	2,315,410	11	7	174.364	SSH-Patator
443	1,341,299	19	15	98.947	DoS slowloris
443	123,122	3	0	266.667	DoS Slowhttptest
443	5,323,637	7	6	65.714	DoS Hulk
80	451,990	9	11	38.444	DoS GoldenEye
80	87,828	11	9	107.091	Heartbleed
443	1,341,300	2	1	0	Web Attack—Brute Force
80	879,012	3	3	0	Web Attack—XSS
443	673,129	9	9	0	Web Attack—Sql Injection
80	1,238,901	1	1	0	Infiltration
443	498,312	2	2	0	Bot
443	2,919,210	5	5	0	PortScan
80	213,012	5	5	0	DDoS

3.2.2. Textual Feature Extraction and Processing

The Deep Learning models learn from the features of the dataset. That is why feature extraction and processing are essential to training the DL models. The dataset features are not always aligned with the DL models' characteristics. That is why we need to process the features. This section presents the textual features and corresponding processing techniques used in this paper.

Normalization

In this experiment, the Z-normalization has been used, which aims to transform the features of a dataset to minimize the mean towards zero and modify the variance to unity. First of all, the dataset has been denoted as $D = \{d_1, d_2, \ldots, d_N\}$, where N is the total number of instances and $d_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ represents the feature vector of instance j. After that, the mean (μ_i) and standard (σ_i) deviation are calculated using Equations (18) and (19).

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} x_{ji} \tag{18}$$

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(x_{ji} - \mu_i)^2} \tag{19}$$

The mean and standard deviation computer is used to normalize each feature value represented using x_{ji} and governed by Equation (20).

$$z_{ji} = \frac{x_{ji} - \mu_i}{\sigma_i} \tag{20}$$

After the transformation, the transformed dataset is expressed as $Z = \{z_1, z_2, \ldots, z_N\}$. Here $z_j = (z_{j1}, z_{j2}, \ldots, z_{jn})$ is the feature vector of instance j after z-normalization.

Handling Categorical Features

The CIC-IDS2017 dataset contains both numerical and categorical data. The numerical data are scaled using the Z-normalization discussed in Section Normalization. However, the categorical data are not suitable for Z-normalization. In this experiment, the categorical data have been processed using the One-Hot encoding technique defined in Equation (21) [43].

$$OH(P)_{encoded} = [p_1, p_2, \ldots, p_n] \tag{21}$$

In Equation (21), $OH(P)_{encoded}$ is the one-hot encoded vector. Each type of categorical data has its separate vectors. Here, P stands for protocol, a categorical feature in the feature vector of the dataset. There are other categorical features as well. Each categorical feature maintains exclusive categorical vectors.

Feature Selection

A dataset for deep learning usually has multiple features. A large dataset with diversified features may not generate the expected results if the features are unrelated to the target variable. This is why selecting features correlating with the target variable is essential [44]. This experiment uses the Mutual Information (MI) technique to discover correlations. It ranks the features according to their relevance to the target variable. The MI technique is governed by mathematical expression (22).

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{22}$$

In Equation (22), the features are X, and the label of the target variables is Y. The joint and marginal probabilities are $p(x, y)$, $p(x)$, and $p(y)$, respectively. The ranking of features is conducted according to the relevance of the feature with the target variable. Features with higher ranks are more relevant to the target variable.

Sequence Generation

The LSTM network requires sequential data. When the proposed IDS for the IoMT is operational, it will receive a network packet sequence with a time step. However, the current dataset is not in sequential form. As a result, it is not suitable to train the LSTM network. It must be converted into sequential data with a time step to train the LSTM network [45]. In this experiment, the sliding window approach has been considered to convert the dataset into sequences. The process is defined in Equation (23).

$$(X_{t-w+1}, X_{t-w+2}, \ldots, X_t) \rightarrow Y_{t+1} \tag{23}$$

In Equation (23), w is the window size. This window is slid over the dataset. The input at t time step is X_t, and the output at the same time step is Y_{t+1}. After completing a certain iteration, the entire dataset is converted to sequential data suitable to train the LSTM network.

3.2.3. Image Dataset Description and Processing

A paper published by L. Nataraj et al. [46] introduced the Malimg Dataset. This dataset contains 9339 images infected with seven different types of malware. Dialer, Backdoor, Worm, Trojan, Trojan-Downloader, Rogue, and Password (PWS) are the malware. The Malimg dataset contains images with varying resolutions. That means there is no common specific size for these images. These are binary images stored in Portable Network Graphics (PNG) format.

Directory Distribution and Splitting

The Malimg dataset has been prepared to classify the family or class of different types of attacks. However, this paper aims at identifying the attacks, regardless of their family. There are 25 classes in the dataset and seven types of attacks. Seven different directories with the names of attacks have been created. The corresponding images have been moved to these directories. Each directory is further divided into three sub-directories. They are training, testing, and validation. The directory structure has been illustrated in Figure 3. Each attack dataset has been split into a training, testing, and validation dataset by maintaining a ratio of 70:15:15, respectively.

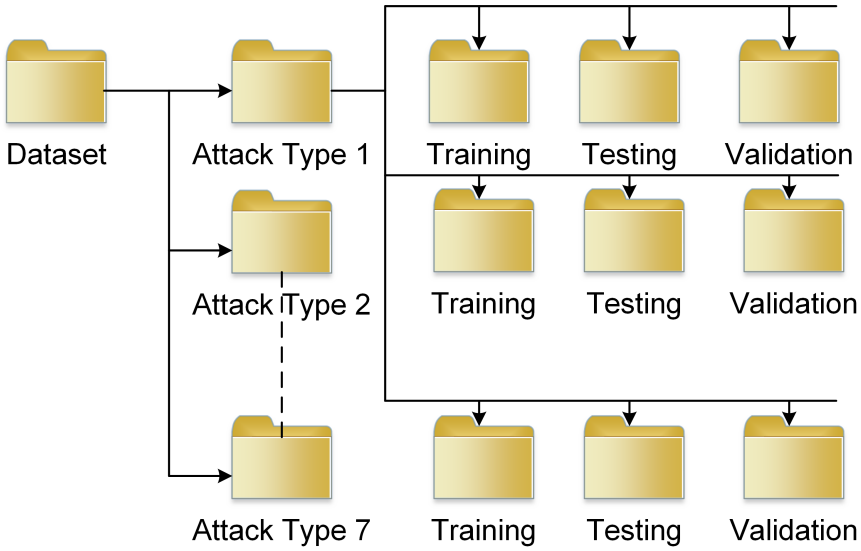

Figure 3. The directory structure of the dataset.

Image Resizing

The Malimg dataset contains images in different resolutions. That is why it is not ready to use to train CNNs because convolutional neural networks have specific input layers with particular sizes. If the input image size does not match the input layer size, the network cannot process the image. That is why resizing the images to a specific resolution is mandatory. It imposes another challenge, which is the information loss from the images. These images contain malicious codes. Resizing the image increases the chances of losing information that represents the malware. We used bicubic interpolation to ensure minimal information loss. The original image is denoted by I, and the size is $M \times N$ pixels. This image is resized to 64×64 pixels and denoted by R. In the Bicubic Interpolation (BI) method [47], the value of a cubic polynomial function at a new point is estimated as the previous value and its derivative at the neighbouring points. The original image I is defined as a continuous function $f(x, y)$ representing the intensity values of pixels at the coordinate (x, y). The entire process is governed by a mathematical model expressed in Equation (24).

$$p(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j \tag{24}$$

In Equation (24), the a_{ij} is the coefficient. The value of this coefficient is calculated from the values of the four nearest neighbours at (x, y) and the derivative of the function $f(x, y)$. The interpolation values are calculated at the desired point after getting the a_{ij} values. Once the interpolation values are available, the scaling factors are computed using (25).

$$s_x = \frac{M}{64}$$
$$s_y = \frac{N}{64} \tag{25}$$

In Equation (25), the coordinates of the resized image R are expressed as (x', y'). These pixel coordinates map them back to the corresponding pixel coordinates of the original image I.

$$x = s_x x' \tag{26}$$
$$y = s_y y' \tag{27}$$

Once the scaling factors are obtained, the Bicubic interpolation estimates the pixel values considering the continuous coordinates (x, y) of the original image I. At the same time, the a_{ij} is calculated, and the cubic polynomial function $p(x, y)$ is evaluated. Finally, the interpolated values are replaced by the pixel values (x, y) of the resized image R. This is how the images are resized to 64×64 pixels with minimal information loss.

3.3. IoMT Node Architecture and Communication Protocols

There is no specific architecture of IoMT [48]. These devices are usually task-specific. For example, the IoMT architecture for an Alzheimer's patient [49] differs from that of a lung cancer patient [50]. This difference is at the sensor and protocol levels. The core IoT architecture illustrated in Figure 4 remains the same.

3.3.1. Basic IoMT Node Architecture

The basic architecture of IoT and IoMT are similar. Both systems have a Communication unit, Processing Unit, Sensor array, and Power Supply. The connected sensors make the fundamental differences between IoT and IoMT. When the set of medical and healthcare-related sensors is connected to the sensor array, it becomes IoMT devices. On the other hand, non-healthcare-related sensors connected to the sensor array are IoT devices. Considering IoT devices with sensors $S_1, S_2, \ldots, S_N$ and IoMT devices as a collection of $MS_1, MS_2, \ldots, MS_N$, the relation between IoT and IoMT is expressed using Equation (28).

$$IoT = \{S | S \in S_N, S \in MS_N\} \tag{28}$$

The IoMT devices, such as IoT devices, are connected to an edge server. The edge server is connected to the router. The router is connected to the Internet Service Provider (ISP), which is further connected to the broader network or Internet gateway. The proposed IoMT intrusion detection system runs at the IoMT edge server. The IoMT sensors are simple input–output (IO) devices. Energy consumption (E), active time (T_{active}), idle time (T_{idle}), power consumption during active state (P_{active}), and energy consumption during idle states (P_{idle}) are a few of the performance evaluation indicators of IoMT devices. The performance indication is characterized by the mathematical Equation (29).

$$E = P_{active} \cdot T_{active} + P_{idle} \cdot T_{idle} \tag{29}$$

From a communication point of view, latency and throughput are the performance evaluation factors. The anomalous nature of the latency and throughput indicates an intrusion at the IoMT node. The throughput, data, and transmission time are characterized by their mathematical relation expressed in Equation (30), where Th_p is the throughput measured in bps, and D is the amount of data processed within T seconds.

$$Th_p = \frac{D}{T} \tag{30}$$

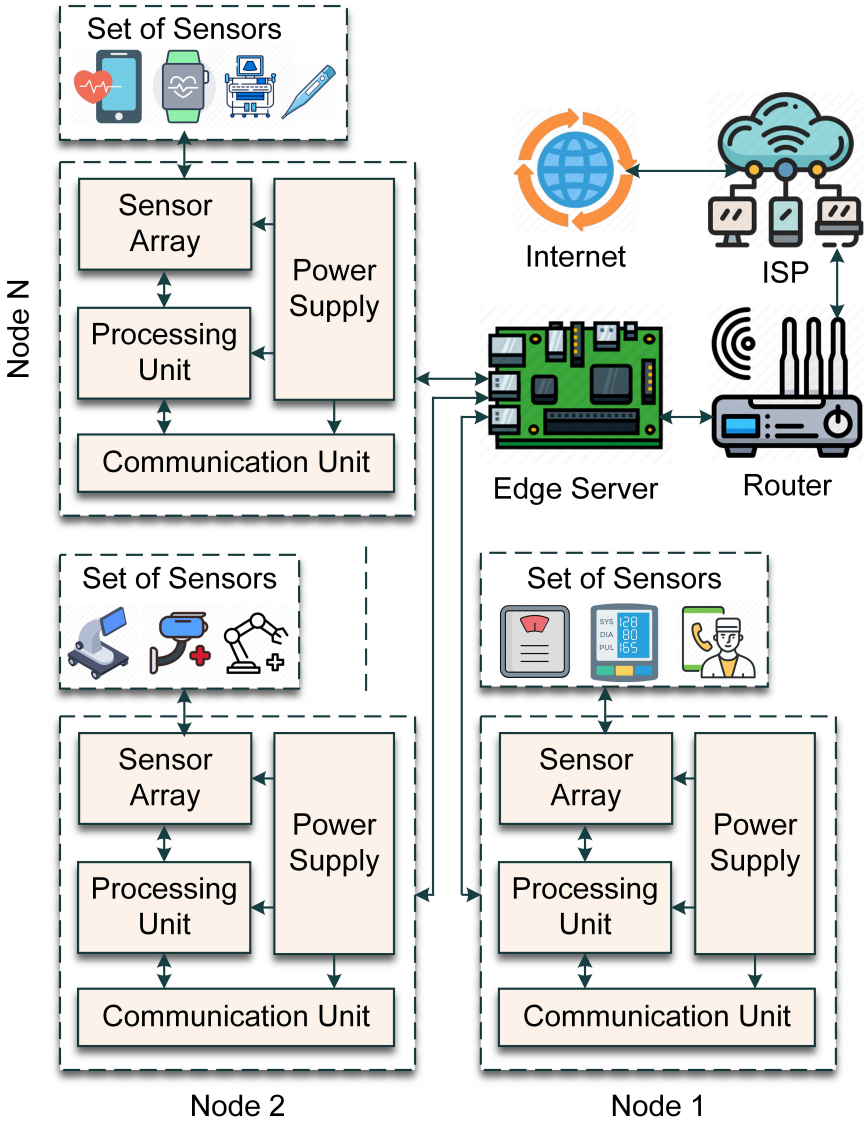

Figure 4. IoT Node Architecture.

3.3.2. Communication Protocols

Protocols govern the communication of the internet. The Internet of Medical Things (IoMT) is no different. A set of protocols also governs IoMT communication. Cybercriminals exploit the existing protocols' vulnerabilities to gain unauthorized access or disrupt the smooth service. Developing an effective intrusion detector for IoMT requires deep knowledge of the protocols [51]. IEEE 802.15.4, Bluetooth Low Energy (BLE), Low-Power Wide Area Network (LoRaWAN), and Cellular IoT are the standard protocols associated with the IoMT [52]. Observing the IoMT performance from the protocol level refers to characterizing the performance in terms of transmission range, data rate, and energy consumption, which is expressed by Equation (31).

$$E_c = \frac{P_{tx} \cdot D}{R} \tag{31}$$

In Equation (31), the transmission power is denoted by P_{tx}, the data rate as R, the amount of data transmitted as D, and the energy consumption as E_c. The received power of a signal from the internet degrades in proportion to the transmission range if it is not amplified. It indicates that receiving low power is not always an indicator of an intrusion at the IoMT nodes. However, the probability is not entirely zero. Using the Friis transmission

equation, the received power over a specific transmission range is calculated using the following Equation (32) [53].

$$P_{rx} = \frac{P_{tx} \cdot G_{tx} \cdot G_{rx} \cdot \lambda^2}{(4\pi)^2 \cdot d^2} \tag{32}$$

The difference between the calculated and the received power, P_{rx}, indicates the intrusion. In Equation (32), G_{tx} and G_{rx} are the transmitter and receiver gains, respectively. The signal's wavelength is λ when the distance between the transmitter and receiver is d.

3.3.3. Common IoMT Intrusions

The CIC-IDS2017 dataset contains features for fourteen different attacks, and the Malimg dataset has seven intrusions. These two datasets are not exclusive to IoMT. The CIC-IDS2017 is a standard dataset for all types of IoT devices. We have already established that $IoT \in IoMT$ but $IoMT \notin IoT$. As a result, not every attack available in these two datasets applies to IoMT nodes. The potential intrusions for IoMT have been listed in Table 2.

Table 2. The potential attacks for IoMT.

CIC-IDS2017 Dataset	Threat Type	Malimg Dataset	Threat Type
Brute Force (BF) Attack	Unauthorized access	Backdoor	Unauthorized access and control
DoS	Reduce the availability	Worm	Exploiting vulnerabilities
DDoS	Reduce the availability	Trojan	Unauthorized access and data integrity
Infiltration	Stealing sensitive data	Trojan Download	Unauthorized access
Portscan	Exploiting vulnerabilities	Rogue	False alarm
Botnet	Unauthorized access and control	Password (PWS)	Data disclosure

There is no guarantee that a certain intrusion will happen at a particular time. The list provided in Table 2 refers to the frequently occurring attacks on IoMT. The probability of a successful attack from these intrusions is defined in Equation (33), which is a conditional probabilistic model, where the probability of a successful attack $P(A)$ is dependent on the probability of the capability of the attackers to exploit the security features and find vulnerabilities. It also depends on the probability of breaching the system's defence mechanism. The probability of breaching the system's defence mechanism is denoted by $P(D)$, and $P(C)$ means the probability of attackers' capability to exploit the security protocols. The $P(S)$ stands for the probability of sustainability of the security features.

$$P(A|C,S,D) = \frac{P(A,C,S,D)}{P(C,S,D)} \tag{33}$$

The list of attacks presented in Table 2 have been shortlisted based on the probabilistic model defined in (33). In this joint probability, the probability of a successful attack depends on $P(C,S,D)$. The twelve attacks have been finalized by analyzing the different probabilistic characteristics of the twenty-one attacks on IoT.

3.4. SafetyMed Architecture

The proposed SafetyMed is a hybrid network, illustrated in Figure 5, consisting of CNN and LSTM. The Internet of Medical Things (IoMT) communicates in textual, numeric, and image data. Intrusion detection from textual and numerical data ignoring the malicious image data leave a severe security vulnerability in IoMT. The LSTM networks are excellent

at detecting intrusions from short and long sequences representing textual or numerical data. The features available on the CIC-IDS2017 dataset have been used to train the LSTM network. On the other hand, CNNs are specially designed Deep Neural Networks (DNNs) to handle image data properly. They are effective in identifying embedded malware in images. Combining the LSTM and CNN results in a complete intrusion detection system that detects intrusion from all textual, numerical, and image data types.

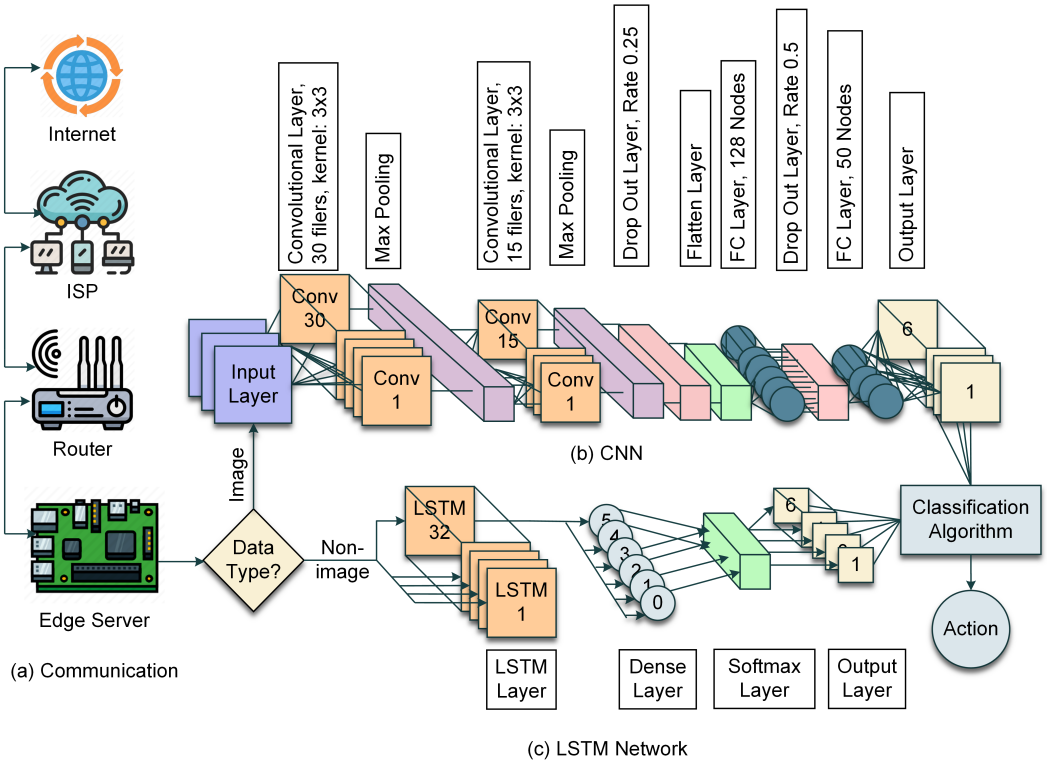

Figure 5. The overview architecture of the SafetyMed.

The proposed SafetyMed architecture uses a parallel combination of CNN and LSTM networks. A data-type differentiating module identifies the input data. It passes the image data to CNN and textual data to the LSTM network. After the differentiating modules' processing, the CNN and LSTM network work separately. The SafetyMed Classification Algorithm (SCA), explained in Section 3.6, combines the predictions from both networks, performing the hybridization. The LSTM network uses numerical features which represent the intrusion from network packet sequences. This means it handles text from a non-NLP point of view.

3.4.1. CNN Design and Implementation

A Convolutional Neural Network (CNN), illustrated in Figure 5b, has been designed, optimized, and implemented for the intrusion detection of malicious image data. It consists of convolutional layers, max-pooling layers, and some additional layers for performance optimization. The fully connected layer architecture has been used in this experiment for feature learning. The network output has seven nodes to detect seven types of intrusions.

Input and Convolutional Layers

The size of the input layer of the proposed CNN is $64 \times 64 \times 3$. That means it requires three-channel images with a resolution of 64×64. Usually, images downloaded from the web or received online have three channels. That is why the input layer accepts three-channel images. The input layer passes the signals to the first convolutional layer. There are 30 filters of kernel size 3×3 in this layer. The working principle of this layer is defined

in Equation (34), where F_i is the convolution filter of each layer, and the output feature map is O_i.

$$O_i(x,y) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} I(x+m, y+n) F_i(m,n) \tag{34}$$

In Equation (34), the kernel size is expressed by k. The i in F_i represents the i-th filter. The Rectified Linear Uni (ReLU), defined in Equation (35), has been used as the activation function of the convolutional layers. The ReLU has been used element-wise in the proposed network architecture to maintain non-linear characteristics.

$$\text{ReLU}(x) = \max(0, x) \tag{35}$$

Pooling Layer Design

The proposed CNN uses a max pooling layer after convolutional layers. It has been used to downsample the operation. Reducing the spatial dimension of the input feature map is an important step of feature extraction. It has been conducted using the max pooling layer, which is expressed in Equation (36).

$$P(x,y) = \max_{m=0}^{k-1} \max_{n=0}^{k-1} I(x \times k + m, y \times k + n) \tag{36}$$

We divided the feature maps into multiple non-overlapping regions. After that, the maximum values of each region are taken using Equation (36). In this equation, $P(x,y)$ is the output from the pooling layer, and the pool size is k. The input feature map, in other words, the input image, is I.

Additional Layers

During the experiment, it was observed that it requires more convolutional layers. Consequently, more ReLU activation functions and pooling layers need to be added. These layers have been considered as an additional layer for the performance improvement of the proposed network. This additional layer consists of a convolutional layer with 15 filters. Each filter's kernel size is 3×3. After processing the signals of this convolutional layer using the ReLU activation function, the max pooling layer with a 2×2 pool size has been used. This architecture tends to overfit. It has been fixed by a dropout layer with a dropout rate of 0.25. The dropout is governed by Equation (37).

The model continues with another convolutional layers (15 filters, 3×3 kernel size, ReLU activation) and a max pooling layer (2×2 pool size), followed by a dropout layer with a dropout rate of 0.25. Dropout is a regularization technique that helps prevent overfitting by randomly setting a fraction of input units to 0 during training:

$$D(x) = \begin{cases} x, & \text{with probability } 1-p \\ 0, & \text{with probability } p \end{cases} \tag{37}$$

According to Equation (37), it randomly chooses weights and turns them into zero. As a result, that particular weight does not play any role in the training and eventually the network becomes more regularized which prevents overfitting. In this equation, $D(x)$ is the dropout layer. The input is defined by x, and the dropout rate is p.

Fully Connected Layer Design

The fully connected layer starts after the dropout layer, which prevents the overfitting problem. However, there is a flattened layer that converts every signal into a one-dimensional array. This one-dimensional data are passed to the fully connected layers, which are also known as dense layers. The first fully connected layer has 128 hidden nodes. Each node is activated using the ReLU activation function. We used a 0.5% dropout rate to prevent the proposed CNN from overfitting. After that, another fully connected layer with 50 hidden

nodes has been introduced. These nodes use the same activation function. The entire process is expressed in Equation (38).

$$y_i = \text{Activation}(\sum_{j=1}^{n} w_{ij}x_j + b_i) \tag{38}$$

In Equation (38), the output of the i-th unit of the fully connected layer is expressed by y_i. The input and weight matrix are denoted by x_j and w_{ij}, respectively. The bias factor is expressed using b_i. In every case, each term is indexed at i except for x because it is the input. The summation defined by Equation (38), iterates according to the number of pixels in the input, which is indexed at j. The entire expression is under the activation function. In this experiment, the ReLU has been used as the activation function.

Output Layer Design

The original Malimg dataset has seven types of intrusions. However, the proposed SafetyMed intrusion detection system focuses on IoMT only. Six intrusions are threats to IoMT according to the probabilistic model of Equation (33). That is why the output layer of the proposed CNN has six nodes. We used the ReLU activation function for the hidden nodes. However, the Softmax activation function has been used for output nodes. The primary reason behind using the Softmax activation function is to obtain the probabilistic responses from the CNN, scaled between 0 to 1. The output layer has been developed using a mathematical model defined in (39).

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \tag{39}$$

In Equation (39), the input from the dense layer to this layer is x_i. These inputs are processed by the Softmax function. The n in the equation is the number of output nodes. In this experiment $n = 6$ because the number of output classes is 6.

Model Compilation

The final step, model compilation, is conducted using the categorical cross-entropy loss function. The proposed CNN is a multi-class classifier. That means it classifies multiple classes simultaneously. For these types of classifiers, the cross-entropy loss function generates better results. This function is defined in Equation (40).

$$L(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \tag{40}$$

In Equation (40), the output class label is y. It is a categorical output. That is why we used the One-Hot encoding scheme to express different classes using binary sequences. The prediction from the model is expressed using $\hat{y}$. The prediction is made in terms of the probability distribution. The maximum n number of classes are predicted using the proposed CNN.

The Adaptive Moment Estimation (ADAM) optimizer has been used to train the model. This optimizer combines Adaptive Gradient (AdaGrad) and Root-Mean-Square Propagation (RMSPorp). That is why this optimizer has the advantages of both of these optimizers [54].

3.4.2. LSTM Network Architecture

An LSTM network, illustrated in Figure 5c, has been designed, implemented, and optimized for this experiment to detect intrusions from sequential network data. It has 32 LSTM layers followed by a dense layer. The dense layer converts the output from the LSTM layer into a one-dimensional array. There are six nodes in the dense layer. These nodes use the Softmax activation function. Finally, a multi-class classification layer has been added to the Softmax layer. It has six classes.

Training the Network

The LSTM network has been trained using the training dataset, which is a subset of the CIC-IDS2017 dataset. There are 39,662 instances in this dataset. The LSTM network requires sequential data at different timesteps. The original CIC-IDS2017 dataset has been processed in Section 3.2, where it has been converted into a sequence with a specific timestep. The proposed LSTM network takes only 2 min 14 s to reach 97.78% validation accuracy with ten epochs. In each epoch, there are 325 iterations. The learning progress of the proposed LSTM network is illustrated in Figure 6.

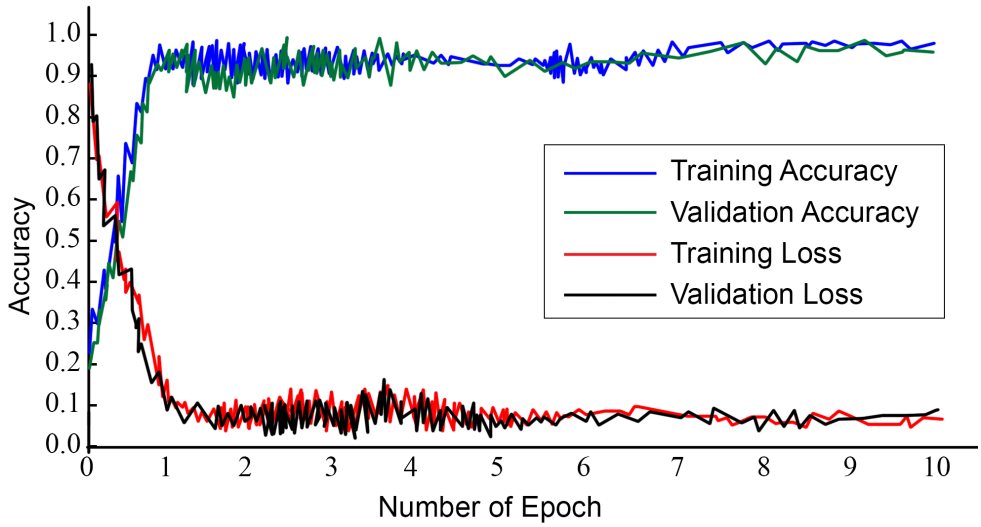

Figure 6. The learning progress with respect to epoch.

The learning progress curve of the LSTM network is illustrated in Figure 6. The learning curve shows the relation among training accuracy, training loss, validation accuracy, and validation loss with respect to the number of epochs and accuracy. The training and validation accuracy curves overlap each other. That means the network does not overfit. Similar characteristics are visible for the training and validation loss as well. The training and validation accuracy increases rapidly until the first epoch. After that, there are repeated small ripples in the learning curve. However, the smoothed version of these curves exhibits near-linear characteristics. A similar statement applies to the training and validation loss as well.

Weight Initialization

The proposed LSTM network has been designed to learn from short and long sequential data. The dataset is large, and numerous significant variations exist among the instances. That is why appropriately initializing the weights is crucial for the network to converge quickly by gaining good accuracy. A proper weight initialization method also handles the vanishing or exploding gradient problems. This experiment uses the Orthogonal Initializing (ORI) method to initialize the weight of the LSTM network [55]. Here, the orthogonal matrix is a square matrix $Q \in \mathbb{R}^{n \times n}$ with the property defined by Equation (41).

$$QQ^T = Q^TQ = I \tag{41}$$

In Equation (41), Q^T represents the transpose of Q. The I is the identity matrix of the same size as Q. These matrices have orthonormal columns and rows. That means they are mutually orthogonal. It implies they have a norm in equation 1 according to expression (42), where q_i and q_j represent the columns of the orthogonal matrix Q. If they are considered orthogonal rows, the overall response from the system remains unchanged.

$$\forall i, j \in \{1, \ldots, n\},\ q_i^T q_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{42}$$

The weight W of the LSTM network is initialized as orthogonal matrices according to the mathematical Equation (43). The weight comes in a $W \in \mathbb{R}^{n \times m}$ form, where $n \neq m$. It starts with a random orthogonal matrix $Q \in \mathbb{R}^{k \times k}$, where $k = \max(n, m)$. After that, the matrix size is batched with the size of the LSTM layer size. This is how the weights have been initialized in the proposed LSTM network.

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial W^T} = \frac{\partial L}{\partial I} \tag{43}$$

Optimization Algorithm

The proposed LSTM network uses an Adaptive Moment Estimation (ADAM) optimization algorithm. It is a combination of the Adaptive Gradient Algorithm (AdaGrad) [56] and Root Mean Square Propagation (RMSProp) [54]. That is why it has the advantages of both of them. It combines these two algorithms by the first and second momentum defined by Equations (44) and (45). Here, the g_t represents the gradient at time step t. The exponential decays for these two equations are β_1 and β_2.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{44}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{45}$$

After calculating the moments, the bias-corrected first and second moments are calculated and denoted by $\hat{m}_t$ and $\hat{v}_t$, which are defined by Equations (46) and (47), respectively.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{46}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{47}$$

The $\hat{m}_t$ and $\hat{v}_t$ are used in Equation (48), where θ_t is the weight update at time t. In this equation, α is the learning rate whose value is between 0 to 1. It is used to control the pace of weight updates. A small constant ϵ is added with a denominator to prevent division by zero.

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{48}$$

3.5. Architectural Comparison with BERT

The architecture of SafetyMed and BERT have similarities and differences. The first similarity is the data diversion according to their types. Both architectures pass the image to the CNN and the non-image data to the recurrent branch. However, the CNN architecture of SafetyMed has been designed to learn features from the Malimg dataset only optimally. On the other hand, the BERT employs a generalized CNN architecture suitable for a wide range of images. A significant difference between SafetyMed and BERT is the preprocessing scheme for recurrent units. The input to these units is preprocessed by following the NLP principle so that they can be vectorized using Word2Vec. Conversely, the SafetyMed preprocess data which carry no semantic meaning. These sequential data are converted into specific sequences segmented at specific timestamps.

The BERT architecture uses a concatenation layer that concatenates the features from CNN and RNN models. A combination of features from both models participates in classification. As a result, the uniqueness of the features is not preserved. SafetyMed maintains a parallel processing scheme where the CNN and LSTM networks perform

concurrently. The image features and textual features participate in the classification process independently. After the model-wise classification, an additional classification algorithm performs the final classification.

3.6. SafetyMed Classification Algorithm

The proposed intrusion detection system for IoMT combines CNN and LSTM networks. They work simultaneously when both image and non-image data are received. If no image data are received, the CNN remains inactive. If there is no non-image data, the LSTM network remains inactive. As illustrated in Figure 5, a classifier combines the prediction from the CNN and LSTM network using the SafeMed Classification Algorithm (SCA). The algorithm is presented as Algorithm 1.

Algorithm 1 The SafetyMed Classification Algorithm

Start
while *packet* **do**
Receive Packet $p \leftarrow$ Type(Packet);
  if $p == image$ **then**
    $i \leftarrow$ ReadImage(p)
    $i \leftarrow$ ResizeImage(p, $64 \times 64 \times 3$)
    $C[t_s] \leftarrow$CNN(i, t_s)
    append.log(t_s)
    if $C[t_s] > 0.7$ **then**
      Intrusion($C[t_s]$, True)
      BlockTraffic($p.source$)
      Reject(p)
    if $p! = image$ **then**
      $d \leftarrow$ ReadData(p)
      $d \leftarrow$ GenerateSequence(d, t)
      D[t_s] $\leftarrow$ LSTM(d, t_s)
      append.log(t_s)
      if $D[t_s] > 0.7$ **then**
        Intrusion($D[t_s]$, True)
        BlockTraffic($p.source$)
        Reject(p) None
Terminate

The SCA is activated when it receives network packets. After receiving packets, it checks the data type of the packets. Image and non-image data are handled differently by separate networks. The algorithm resizes the images to $64 \times 64 \times 3$ and then passes them to the CNN. It maintains a log according to timestamp t_s. The processing is different for non-image data. Initially, the non-image data are converted into length sequences d with t time differences. These sequences are passed to the LSTM network. Algorithm 1 uses the trained CNN and LSTM network to classify the intrusions from both image and non-image data. If the probability of a certain class is more than 70%, then the algorithm considers it to be a considerable intrusion. It immediately blocks traffic from the source and also rejects the current packet.

Complexity Analysis

The time complexity analysis of a system consisting of CNN and LSTM networks is not straightforward. It depends on the number of layers, the size of the input data, the weights of nodes, the combination of biases, etc. The complexity of an image with n is $O(n + CNN)$. The LSTM network handles the non-image data. For data with m

size, the complexity is $O(m + LSTM)$. Considering that the system loops over every packet where the number of packets is p, the overall time complexity of the algorithm is $O(p \times (n + m + CNN + LSTM))$ for the worst case. However, it is $O(p)$ for the best case.

4. Implementation and Response Mechanism

The proposed SafetyMed has been implemented and experimented with in the testbed. Although the experimental environment has been artificially created, the real-world intrusion characteristics have been maintained by randomizing the intrusions. The malicious packets have been randomly injected with regular packets, as if coming from real attackers.

4.1. Experimental Setup

The experimental setup has been designed using Cooja Network Simulator. This network simulator generates a realistic replication of a busy network. It runs on the Contiki Operating System (COS) [57]. In this experiment, the COS has been installed in a virtual machine. A Raspberry Pi 4 Model B with 8GB primary memory has been used as the edge server. The experimental setup has been illustrated in Figure 7. The physical hardware houses the virtual hardware through Virtualization Technology (VT). The Contiki OS is installed on the virtual hardware. It simulates a realistic network as a dynamic network generating and carrying thousands of packets in different parts of the world. A separate intrusion server has been created to insert malicious codes among the network traffic randomly. A random mixture module retrieves different types of image and non-image intrusions with the simulated network traffic. A wireless router receives the traffic and converts non-image data into sequential data at different timesteps. It converts the image data into a grid data structure. Finally, these data are transmitted to the edge server where the proposed SafetyMed runs. This edge server has implemented the SafetyMed classification (SC) algorithm. If no intrusion is detected, the network traffic is allowed to communicate with the IoMT devices. Otherwise, the SC algorithm blocks the source that generates intrusion and rejects the corresponding packets.

4.2. Threshold Selection

The proposed Deep-IDS runs from the edge server. The sensors receive data from the internet through the server. The same thing applies to data transmission as well. The edge server sits in between the sensors and the router. Whenever the edge server receives a packer, it initiates the Deep-IDS and passes the packets to it. If the LSTM network detects the intrusion, it checks the probability of certain intrusion. If the probability is more than 70%, then the Deep-IDS rejects the data and sends the traffic block signal to the router. The router blocks the source of the intrusion. The Deep-IDS passes the data to the sensors if no intrusion is detected.

Figure 8 shows an inverse relationship between false alarms and detection rates as the threshold value increases. The detection rate is high with lower threshold values, but so is the false alarm rate. The false alarm rate decreases as the threshold value increases, but the detection rate also drops, indicating a trade-off between the two metrics. If there are too many false alarms, the system will cause too much interruption in regular operations. If the detection rate falls, the system will fail to secure the IoMT. Choosing an optimal threshold value that balances minimizing false alarms and maximizing detection rates is challenging. It is evident in Figure 8 that at a threshold value of 70, the false alarm rate is significantly reduced to 1.46% while maintaining a relatively high detection rate of 98.01%, which further increases the threshold results in a steep decline in the detection rate; this is unacceptable depending on the specific use case. From this observation, the response mechanism of the system has been at a threshold of 70%. At this threshold value, the proposed intrusion detection system maintains the optimum false alarm rate and detection rate.

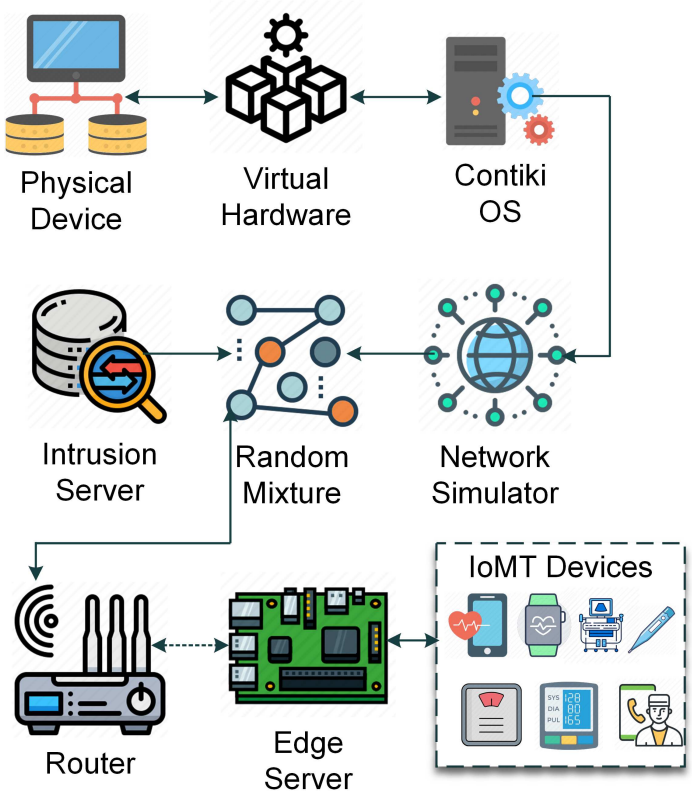

Figure 7. Overview of the implementation of the proposed IoMT IDS.

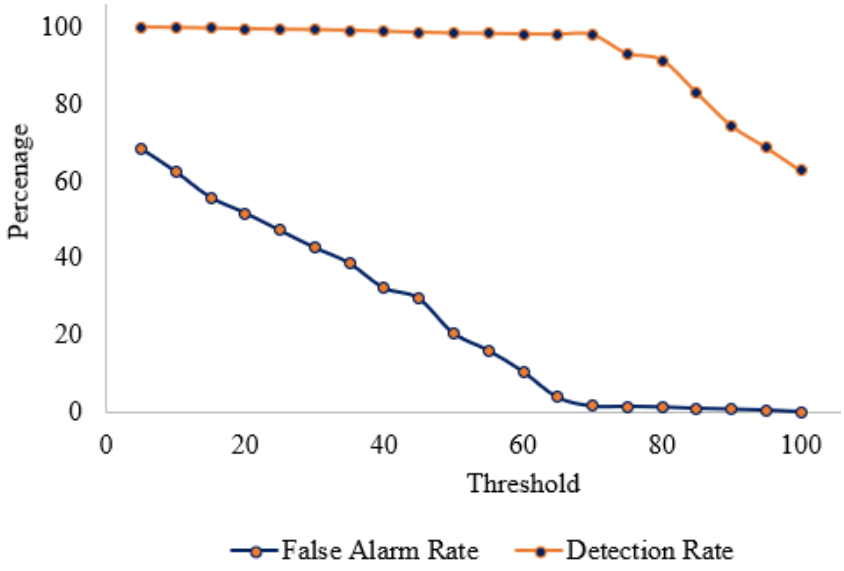

Figure 8. Threshold selection.

5. Performance Evaluation

SafetyMed is the first IoMT intrusion detection system that simultaneously detects the intrusion from image and non-image data. The network architecture is a combination of CNN and LSTM networks. That is why the performance of the proposed IDS has been evaluated from image intrusion, non-image intrusion, and combined intrusion classification. This experiment uses the CICIDS2017 dataset. The overall performance of this dataset is satisfactory. However, it leaves a question on the performance of the SafetyMed on other datasets. It has been evaluated using CICIDS2018 and CICIDS2019 datasets to leave no stone unturned. The SafetyMed is the first IDS capable of detecting and defending twelve

intrusions. No other IDS has this capability. As a result, very little common ground exists between SafetyMed and other IDSs. Because of being exclusive, its performance could not be compared with other similar IDSs.

5.1. Evaluation Metrics

The same evaluation metrics have been used for both CNN and LSTM. Both networks perform classification. That is why using the same evaluation metrics has become possible. According to the literature review, accuracy, precision, recall (sensitivity), and F1 Score are the state-of-the-art evaluation metrics for classification problems. These metrics are dependent on Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These values are obtained from the confusion matrix. Mathematically, the evaluation metrics are defined by Equations (49)–(52), respectively [58].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{49}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{50}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{51}$$

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{52}$$

Accuracy is the measurement of the number of correctly detected intrusions. There are both positive and negative classes in the dataset. Precision is the measurement of the quality of positive prediction by SafetyMed. The recall represents the percentage of correctly identified positive classes out of the total classes. The F1 score combines precision and recall to determine the number of times the model makes correct predictions from the entire dataset.

5.2. Performance Analysis of LSTM

The confusion matrix, illustrated in Figure 9, demonstrates an excellent performance by the proposed intrusion detection system. It is the performance of the system on sequential data. It detects intrusions with 98.23% accuracy. The average precision and recall are above 95%, indicating the prediction's reliability from the LSTM network.

The performance of the LSTM network has been summarized in Table 3. This comprehensive evaluation shows that the precision value ranges from 0.958 for DoS to 0.9898 for Botnet intrusion. The precision represents the ability of SafetyMed to classify the true positives correctly. That means the system is very effective in minimizing false alarms. The lowest recall is 0.9631 for Portscan, and the highest is 0.98 for the DoS attack. This indicates the proportion of the true positive predicted out of all actual positive cases.

The F1 score in Table 3 indicates the balanced performance of SafetyMed. It is a combination of precision and recall. It ranges from 0.9662 for DDoS to 0.9768 for Botnet, which demonstrates the ability of the proposed intrusion detection system to maintain a high precision and recall. Figure 10 illustrates the performance of the LSTM network in detecting intrusions.

In Figure 10, the precision, recall, and F1-score values are on the left vertical axis. The FPR and FNR are scaled on the right vertical axis. The low False Positive Rates (FPR) and False Negative Rates (FNR) reveal the system's aptitude for minimizing both classification errors. The FPRs range from a mere 0.0011 for Botnet to 0.0084 for DoS, indicating that the system rarely misclassifies benign activities as threats. Similarly, the FNRs, which measure the proportion of false negatives to actual positives, are relatively low, with values between 0.02 for DoS and 0.0369 for Portscan. From the confusion matrix analysis of Figure 9, data in Table 3, and performance visualization in Figure 10, it is evident that the experimenting LSTM network is an excellent non-image intrusion detector and ensures reliable security, protecting the IoMT network from intrusions.

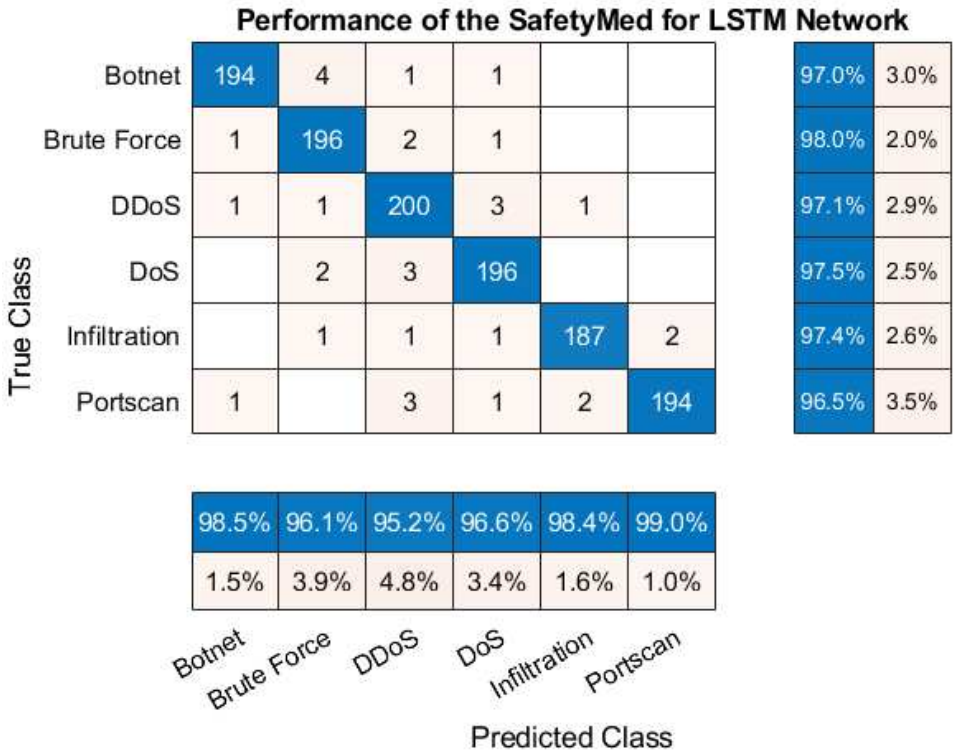

Figure 9. Confusion Matrix Analysis for LSTM Network.

Table 3. Performance summary of the LSTM network.

Intrusion	Precision	Recall	F1 Score	FPR	FNR
Brute Force	0.9848	0.9798	0.9823	0.003	0.0202
DoS	0.958	0.98	0.9689	0.0084	0.02
DDoS	0.9615	0.9709	0.9662	0.008	0.0291
Infiltration	0.9647	0.975	0.9698	0.0076	0.025
Portscan	0.9833	0.9631	0.9731	0.0026	0.0369
Botnet	0.9898	0.9641	0.9768	0.0011	0.0359

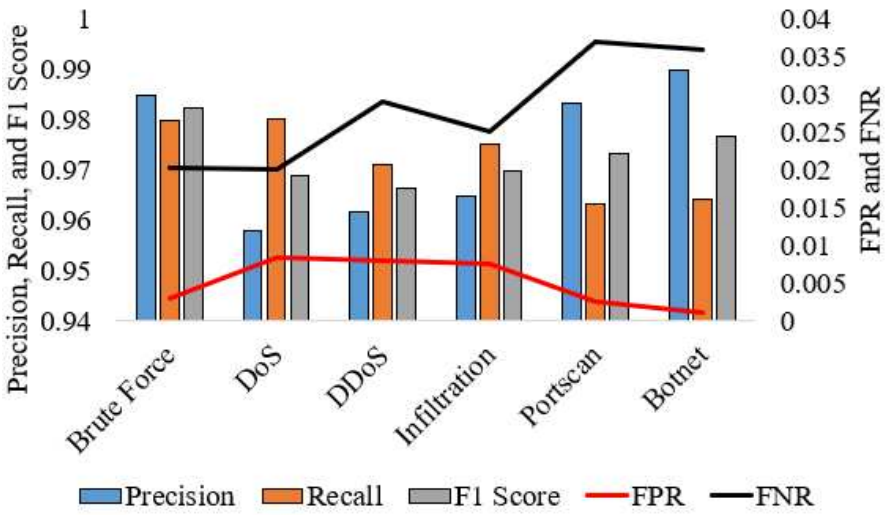

Figure 10. The performance visualization of the LSTM network.

5.3. Performance Analysis of CNN

The confusion matrix illustrated in Figure 11 demonstrates the performance of the CNN of the SafetyMed intrusion detection system. It is the performance of the system on image data. It classifies the intrusions with 96.92% accuracy. The average precision of the intrusion detector is 0.9692. This indicates that the model accurately predicts true positive cases. The characteristics of the recall are quite similar. The average recall is 0.9644, which suggests that the system is sensitive to intrusion, and it can detect a significant number of intrusions, minimizing the risk of undetected threats.

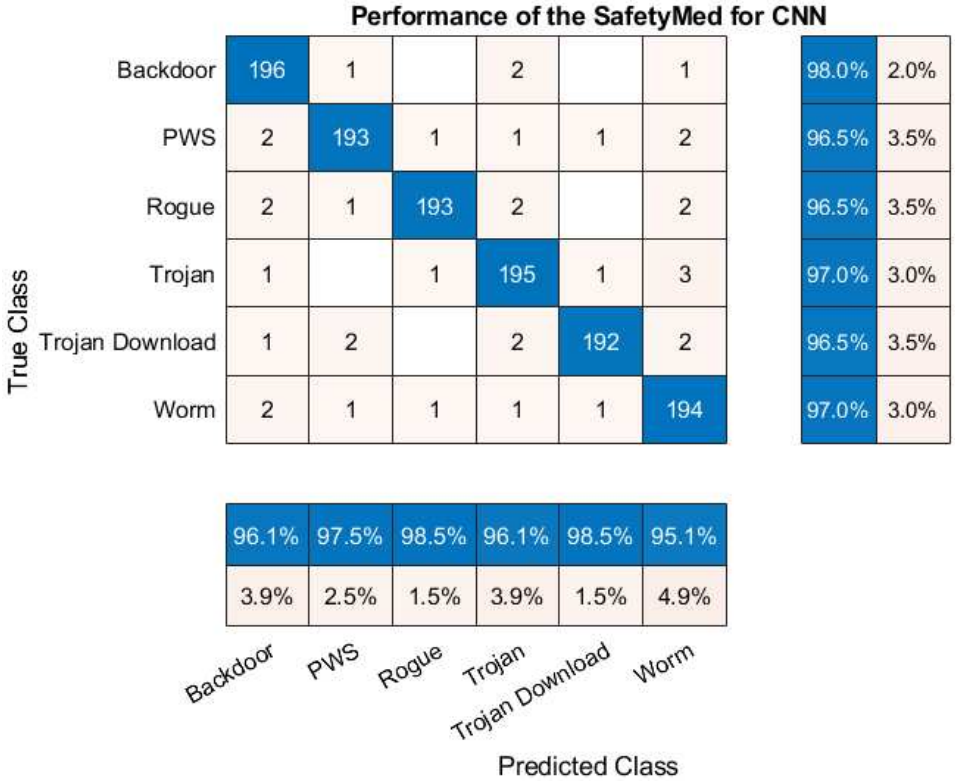

Figure 11. Confusion Matrix Analysis for CNN.

The overall performance of the trained CNN to detect an intrusion is listed in Table 4. The highest precision of the system is recorded at 0.9797 for the Trojan, and the lowest is 0.9608 for the Backdoor attack. The recall ranges from 0.9509 for PWS to 0.98 for the Backdoor attack. The lowest F1-score is 0.9604 for the PWS, and the highest F1-score is 0.9723 for the Trojan attack. It has been observed that the precision, recall, and F1-score do not fall below 0.96, which is evidence that the trained CNN is excellent at detecting an intrusion through images.

Table 4. Performance summary of the CNN.

Intrusion	Precision	Recall	F1 Score	FPR	FNR
Backdoor	0.9608	0.98	0.9704	0.0072	0.02
Worm	0.965	0.965	0.965	0.0063	0.035
Trojan	0.9797	0.965	0.9723	0.0036	0.035
TRD	0.9653	0.9653	0.9653	0.0063	0.0347
Rogue	0.9746	0.96	0.9673	0.0045	0.04
PWS	0.97	0.9509	0.9604	0.0054	0.049

The performance of the proposed intrusion detection system has been visually presented in Figure 12. The left x-axis represents the precision, recall, and F1-score performance scale. The right x-axis represents the FPR and FNR scales. The average FPR and FNR of the SafetyMed are 0.0054 and 0.049, respectively. It is an indication that the IDS rarely classifies non-intrusion events as intrusive.

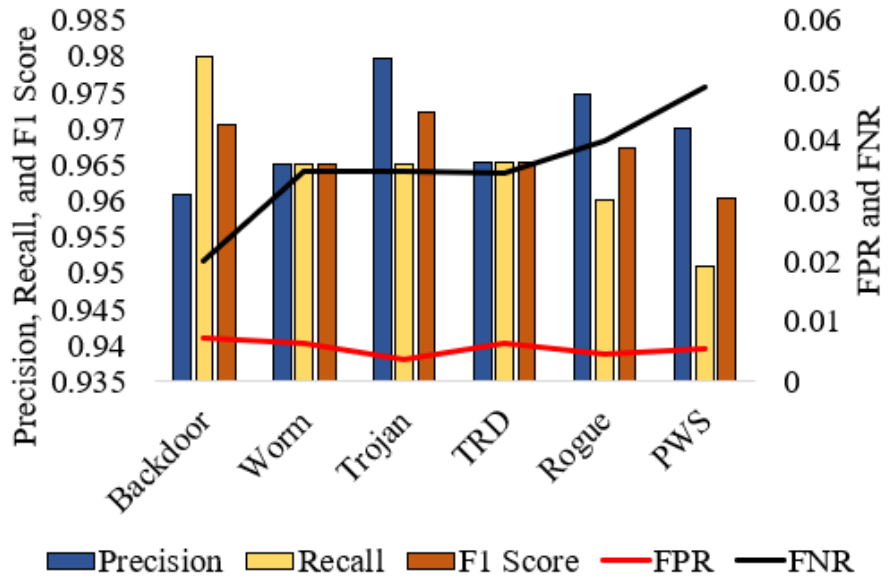

Figure 12. The performance visualization of the CNN.

5.4. Overall Performance

The overall performance of the proposed intrusion detection system, SafetyMed, is represented by the confusion matrix illustrated in Figure 13. The average accuracy of SafetyMed is 97.63%. The average precision, recall, and F1-score are 0.9847, 0.97, and 0.9773, respectively. The average FPR and FNR are 0.0071 and 0.0267, respectively.

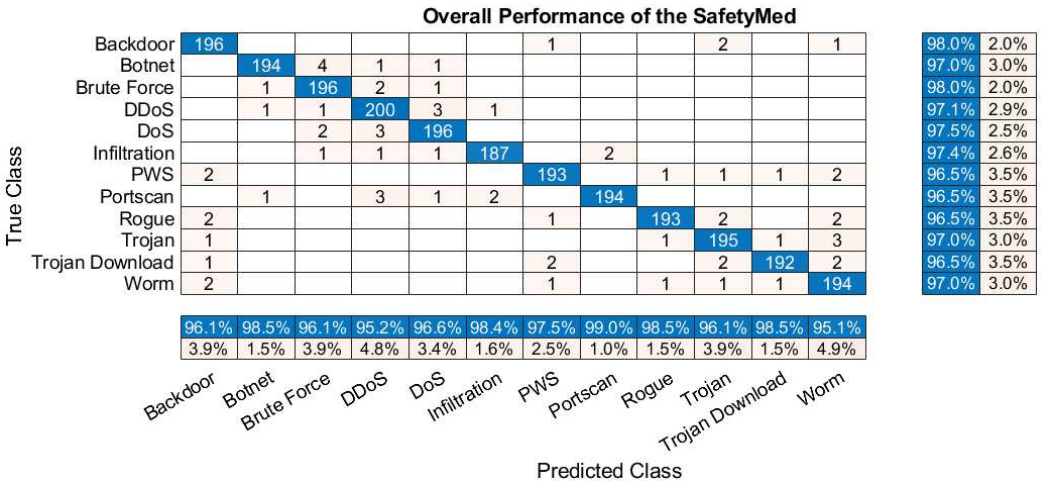

Figure 13. The overall performance of the proposed intrusion detection system.

The variations among the precision, recall, and F1-score are illustrated in Figure 14. According to this figure, the range of these values varies from 0.95 to 0.99. Intrusion detection with performance in between this range is undoubtedly an outstanding IDS. Moreover, the FPR is 0.71% and FNR is 2.67%. These values indicate that the proposed SafetyMed is reliable and rarely generates false positive or negative results.

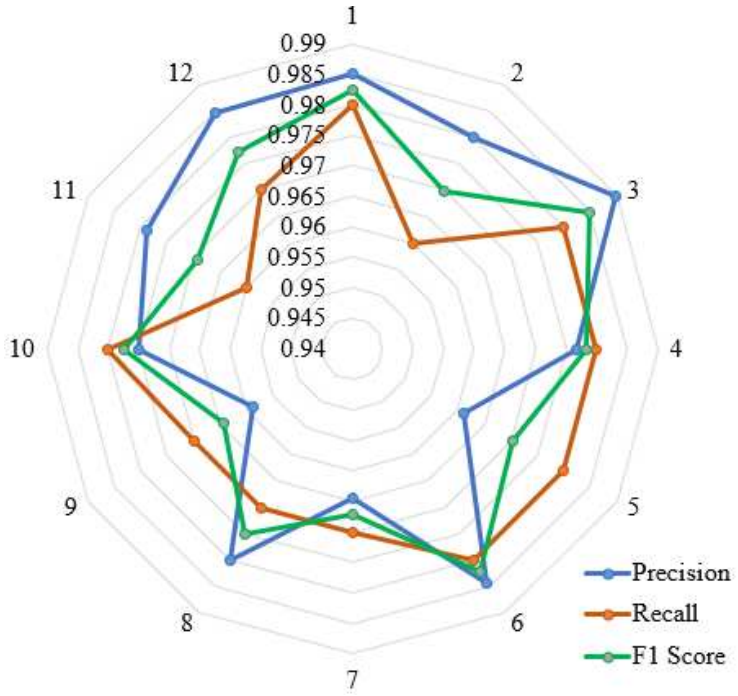

Figure 14. The variations amongthe precision, recall, and F1-score.

5.5. Performance on Different Datasets

SafetyMed demonstrates an outstanding performance on the CICIDS2017 dataset. The dataset has been split into three segments—(i) Training, (ii) Testing, and (iii) Validation. The network's performance has been tested using the Testing dataset, which was not used during the training phase. Although it is enough to validate the capability of SafetyMed, it has been further tested with CICIDS2018 and CICIDS2019 datasets. The performance comparison of SatefyMed for different datasets is listed in Table 5. The performance variations of SafetyMed for different datasets are insignificant.

Table 5. Performance of the proposed system on different datasets.

	Precision			**Recall**			**F1 Score**		
Intrusion	**CICDIS 2017**	**CICDIS 2018**	**CICDIS 2019**	**CICDIS 2017**	**CICDIS 2018**	**CICDIS 2019**	**CICDIS 2017**	**CICDIS 2018**	**CICDIS 2019**
Backdoor	0.9608	0.966	0.96527	0.98	0.98279	0.98168	0.9704	0.97040	0.96984
Worm	0.965	0.966	0.96779	0.965	0.96835	0.96332	0.965	0.96612	0.96444
Trojan	0.9797	0.980	0.97970	0.965	0.96556	0.96444	0.9723	0.97398	0.97118
TRD	0.9653	0.965	0.96753	0.9653	0.96977	0.96865	0.9653	0.96753	0.96642
Rogue	0.9746	0.973	0.97963	0.96	0.96168	0.96447	0.9673	0.96618	0.96562
PWS	0.97	0.969	0.96832	0.9509	0.95369	0.94922	0.9604	0.96319	0.96096
Brute Force	0.9848	0.983	0.98815	0.9798	0.97924	0.98315	0.9823	0.98286	0.98565
DoS	0.958	0.957	0.95688	0.98	0.98000	0.97832	0.9689	0.97225	0.96946
DDoS	0.9615	0.964	0.96485	0.9709	0.96922	0.97537	0.9662	0.96843	0.97123
Infiltration	0.9647	0.966	0.96861	0.975	0.97947	0.97500	0.9698	0.96868	0.97315
Portscan	0.9833	0.986	0.98274	0.9631	0.96366	0.96813	0.9731	0.97589	0.97310
Botnet	0.9898	0.994	0.99371	0.9641	0.96466	0.96689	0.9768	0.97680	0.98127

The graphical illustration of the comparison is presented in Figure 15. It also shows insignificant performance variations for different datasets. The performances of SafetyMed have been compared in terms of precision, recall, and F1-score. Figure 15 shows the

performance of each type of attack for every dataset experimented in this paper. The ignoble variations listed in Table 5 and demonstrated in Figure 15 prove that SafetyMed is well-trained, not biased to any particular dataset, and maintains a generalized performance when the dataset varies.

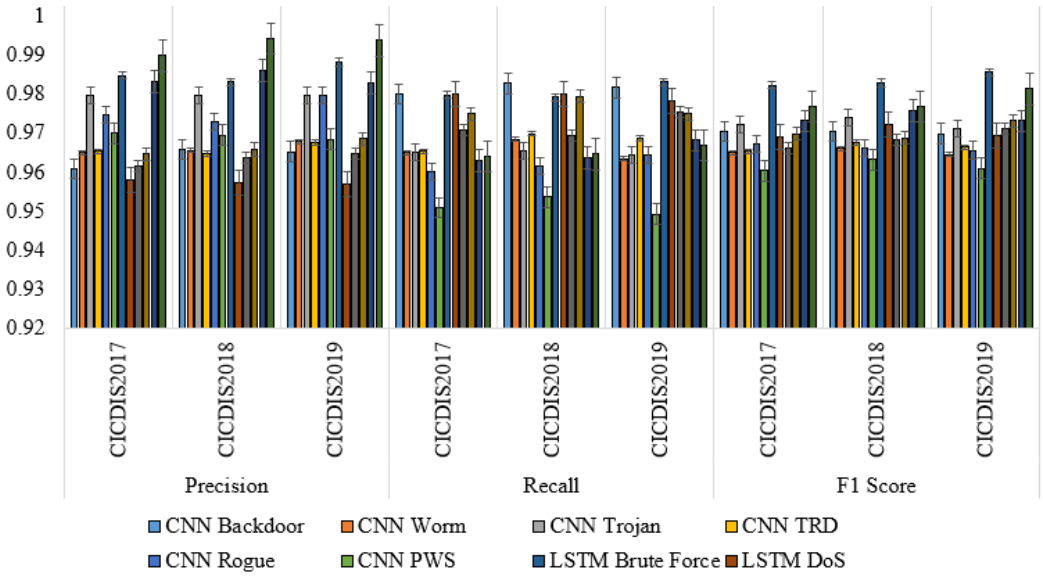

Figure 15. Performance comparison among different datasets.

5.6. Performance Comparison

The performance of SafetyMed has been compared with five concurrent papers, which are listed in Table 6. The comparison shows that it significantly outperforms the state-of-the-art (SOTA) approaches in handling various intrusions and incorporating a response mechanism. While the papers by V. Ravi et al. [59], M. Alalhareth et al. [60], S. A. Wagan [9], W. Lu et al. [61], and S. Saif et al. [62] can only detect up to 2 types of intrusions, the proposed method can successfully detect 12 different types. Regarding accuracy, S. Saif et al. [62] achieved the highest accuracy of 99% among the SOTA approaches. However, the proposed method closely follows this with an impressive accuracy of 97.63%, which is commendable given its ability to detect a far greater number of intrusion types. The proposed approach also stands out with its unique response mechanism, a feature absent in the SOTA methods. Additionally, it includes a detection rate of 98.01%, a metric not provided by the other studies. Therefore, the proposed approach significantly advances the existing methodologies, boasting a broader scope and an integrated response mechanism.

Table 6. Performance comparison with state-of-the-art (SOTA) methodologies.

Paper	Number of Intrusions	Detection Rate	Accuracy	Response Mechanism
V. Ravi et al. [59]	2	NA	95%	NA
M. Alalhareth et al. [60]	2	NA	88.90%	NA
S. A. Wagan [9]	2	NA	92%	NA
W. Lu et al. [61]	2	NA	81.96	NA
S. Saif et al. [62]	1	NA	**99%**	NA
Proposed	**12**	**98.01%**	**97.63%**	**Yes**

6. Limitations and Future Direction

Despite the outstanding performance, the proposed SafetyMed suffers from several limitations. These limitations are not intractable. However, it is beyond the scope of this paper to overcome these limitations at this phase. As a result, it leaves an opportunity to conduct more research in the future to strengthen SafetyMed and includes additional features.

6.1. Architectural Complexity

The proposed IoMT intrusion detection system combines CNN and LSTM networks. As a result, the overall architecture of SafetyMed has become a complex one. Attaining the same result with simpler architecture would be desired. However, this opportunity has not been explored yet. That means there is still scope to optimize the network architecture further to make it simpler, which is the future scope of this research.

6.2. Expensive

SafetyMed uses a Raspberry Pi 4 Model B with 8GB primary memory as the edge server, which is an expensive device for many researchers. It requires an additional cooling unit and power supply unit as well. That is why the upfront cost of developing SafetyMed is high. However, the Raspberry Pi is a multipurpose headless computer, but the cost can be reduced if an embedded system is designed for SafetyMed only; however, this can be considered as a future scope of the paper.

6.3. Adversarial Machine Learning (AML) Attack

The proposed intrusion detection system has been trained with the public dataset. That means it is possible to discover the features through reverse engineering. It makes the system vulnerable to Adversarial Machine Learning (AML) attacks [63]. Although the intruder would not have direct access to add irrelevant data to the training dataset, there are still risks of embedding malicious code which does not fall within the learned features' characteristics of the trained models. This vulnerability will be studied in the subsequent versions of SafetyMed.

6.4. Testbed Experiment

The experimental environment replicates the real-world environment. Although the network simulator generates traffic as if it is an original network, the insertion of intrusion is conducted randomly. It has been taken for granted that there will be an intrusion. This assumption deviates the experimental setup from 100% resemblance to the real-world environment. This weakness can be overcome with permission to install the proposed IDS in a functioning IoMT network.

6.5. Cyber-Physical System Security

The proposed intrusion detection system focuses on cybersecurity only. Cyber-physical system security is also essential to maintain the integrity of the IoMT network [64]. Anyone with access to the edge server of the SafetyMed can temper the trained model, retrain the models with the adversarial dataset, or bypass the connection with the edge server and expose the IoMT devices to direct network traffic. Implementing a cyber-physical security module can overcome this limitation, which we plan to implement in the future.

7. Conclusions

The healthcare sector is on the verge of a paradigm shift, and the availability and popularity of the Internet of Medical Things (IoMT) have accelerated this transition from a traditional to a personalized one. However, the security concerns of these devices have become a major barrier to the massive adoption of this technology. The resource-constrained architecture of the IoMTs is insufficient to employ sophisticated security algorithms to defend against ever-changing malicious signal patterns intended to intrude on healthcare services. SafetyMed, presented in this paper, ensures maximum protection of the IoMT by

defending against twelve types of attacks. Augmenting this technology with the existing IoMT-enabled services is a potential solution to this sector's security concern. The novelty of SafetyMed lies in the innovative architecture, detection rate optimization, false positive rate reduction, and effective classification algorithm development. These novelties fill up the gaps discovered in existing solutions. It detects an intrusion with an average accuracy of 97.63%. The average False Positive Rate (FPR) of SafetyMed is only 0.71%, representing its reliability. The precision of the SafetyMed is 98.47%, which indicates that 98.47% of the positively identified intrusions are correct. A similar performance is observed for the recall and F1-score, which are 97% and 97.73%, respectively.

Despite the outstanding performance and remarkable potential, SafetyMed is not immune to limitations, including architectural complexities and production costs. Moreover, it has only been applied in an experimental setup, which may not reflect the real-world scenarios in many cases just yet. Furthermore, SafetyMed has no defence mechanism against the AML attack, which is a major limitation of this system. The cyber-physical system security is another weakness of it. However, these limitations pave the path to conducting further experiments, facilitating more research scopes for advancing this technology.

Author Contributions: Conceptualization, N.F. and M.A.Y.; methodology, N.F. and M.W.; software, S.A.A. and M.A.K.; validation, P.L. and M.A.M.; formal analysis, M.A.M.; investigation, P.L. and S.A.A.; resources, M.A.M.; data curation, N.F.; writing—original draft preparation, N.F.; writing—review and editing, A.A.; visualization, M.A.Y.; supervision, M.A.Y. and M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-RP23004).

Data Availability Statement: Intrusion Detection Evaluation Dataset (CIC-IDS2017) dataset has been used in this research which is publicly available at https://www.unb.ca/cic/datasets/ids-2017.html.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Natarajan, R.; Lokesh, G.H.; Flammini, F.; Premkumar, A.; Venkatesan, V.K.; Gupta, S.K. A Novel Framework on Security and Energy Enhancement Based on Internet of Medical Things for Healthcare 5.0. *Infrastructures* **2023**, *8*, 22. [CrossRef]
2. Shakeel, T.; Habib, S.; Boulila, W.; Koubaa, A.; Javed, A.R.; Rizwan, M.; Gadekallu, T.R.; Sufiyan, M. A survey on COVID-19 impact in the healthcare domain: Worldwide market implementation, applications, security and privacy issues, challenges and future prospects. *Complex Intell. Syst.* **2023**, *9*, 1027–1058. [CrossRef] [PubMed]
3. Bhushan, B.; Kumar, A.; Agarwal, A.K.; Kumar, A.; Bhattacharya, P.; Kumar, A. Towards a Secure and Sustainable Internet of Medical Things (IoMT): Requirements, Design Challenges, Security Techniques, and Future Trends. *Sustainability* **2023**, *15*, 6177. [CrossRef]
4. Goscinski, A.; Delicato, F.C.; Fortino, G.; Kobusińska, A.; Srivastava, G. Special issue on Distributed Intelligence at the Edge for the Future Internet of Thing. *J. Parallel Distrib. Comput.* **2023**, *171*, 157–162. [CrossRef]
5. Khan, I.A.; Moustafa, N.; Razzak, I.; Tanveer, M.; Pi, D.; Pan, Y.; Ali, B.S. XSRU-IoMT: Explainable simple recurrent units for threat detection in Internet of Medical Things networks. *Future Gener. Comput. Syst.* **2022**, *127*, 181–193. [CrossRef]
6. Si-Ahmed, A.; Al-Garadi, M.A.; Boustia, N. Survey of machine learning based intrusion detection methods for internet of medical things. *Appl. Soft Comput.* **2023**, *140*, 110227. [CrossRef]
7. Idrissi, I.; Boukabous, M.; Grari, M.; Azizi, M.; Moussaoui, O. An Intrusion Detection System Using Machine Learning for Internet of Medical Things. In Proceedings of the 3rd International Conference on Electronic Engineering and Renewable Energy Systems: ICEERE 2022, Saidia, Morocco, 20–22 May 2022; Springer: Berlin/Heidelberg, Germany, 2023; pp. 641–649.
8. Khan, F.; Jan, M.A.; Alturki, R.; Alshehri, M.D.; Shah, S.T.; ur Rehman, A. A Secure Ensemble Learning-Based Fog-Cloud Approach for Cyberattack Detection in IoMT. *IEEE Trans. Ind. Inform.* **2023**, *19*, 10125–10132. [CrossRef]
9. Wagan, S.A.; Koo, J.; Siddiqui, I.F.; Qureshi, N.M.F.; Attique, M.; Shin, D.R. A fuzzy-based duo-secure multi-modal framework for IoMT anomaly detection. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 131–144. [CrossRef]
10. Kumar, A.K.; Vadivukkarasi, K.; Dayana, R. A Novel Hybrid Deep Learning Model for Botnet Attacks Detection in a Secure IoMT Environment. In Proceedings of the 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 9–11 February 2023; pp. 44–49.
11. Srinarayani, K.; Padmavathi, B.; Kavitha, D. Detection of Botnet Traffic using Deep Learning Approach. In Proceedings of the 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 23–25 March 2023; pp. 201–206.

12. Fan, X.; Yang, R. A Network Intrusion Detection Method Based on Improved Bi-LSTM in Internet of Things Environment. *Int. J. Inf. Technol. Syst. Approach (IJITSA)* **2023**, *16*, 1–14. [CrossRef]
13. Eilertsen, G.; Kronander, J.; Denes, G.; Mantiuk, R.K.; Unger, J. HDR image reconstruction from a single exposure using deep CNNs. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1–15. [CrossRef]
14. Chitra, R. A Novel Autoencoder Based Feature Independent Ga Optimised Xgboost Classifier for Iomt Malware Detection. *SSRN* **2023**, 1–29. [CrossRef]
15. Karagiannis, S.; Ribeiro, L.L.; Ntantogian, C.; Magkos, E.; Campos, L.M. Chidroid: A Mobile Android Application for Log Collection and Security Analysis in Healthcare and IoMT. *Appl. Sci.* **2023**, *13*, 3061. [CrossRef]
16. Tahir, B.; Jolfaei, A.; Tariq, M. A Novel Experience-Driven and Federated Intelligent Threat-Defense Framework in IoMT. *IEEE J. Biomed. Health Inform.* **2023**, 1–8. [CrossRef]
17. Lu, W. Applied Machine Learning for Securing the Internet of Medical Things in Healthcare. In Proceedings of the Advanced Information Networking and Applications: Proceedings of the 37th International Conference on Advanced Information Networking and Applications (AINA-2023), Juiz de Fora, Brazil, 29–31 March 2023; Springer: Berlin/Heidelberg, Germany, 2023; Volume 2, pp. 404–416.
18. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. *Sensors* **2017**, *17*, 1967. [CrossRef] [PubMed]
19. Khan, I.A.; Moustafa, N.; Pi, D.; Sallam, K.M.; Zomaya, A.Y.; Li, B. A new explainable deep learning framework for cyber threat discovery in industrial IoT networks. *IEEE Internet Things J.* **2021**, *9*, 11604–11613. [CrossRef]
20. Awotunde, J.B.; Abiodun, K.M.; Adeniyi, E.A.; Folorunso, S.O.; Jimoh, R.G. A deep learning-based intrusion detection technique for a secured IoMT system. In Proceedings of the Informatics and Intelligent Applications: First International Conference, ICIIA 2021, Ota, Nigeria, 25–27 November 2021; Revised Selected Papers; Springer: Berlin/Heidelberg, Germany, 2022; pp. 50–62.
21. Garg, N.; Wazid, M.; Singh, J.; Singh, D.; Das, A.K. Security in IoMT-driven smart healthcare: A comprehensive review and open challenges. *Secur. Priv.* **2022**, *5*, e235. [CrossRef]
22. Rbah, Y.; Mahfoudi, M.; Balboul, Y.; Fattah, M.; Mazer, S.; Elbekkali, M.; Bernoussi, B. Machine learning and deep learning methods for intrusion detection systems in iomt: A survey. In Proceedings of the 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 3–4 March 2022; pp. 1–9.
23. Saxe, A.; Nelli, S.; Summerfield, C. If deep learning is the answer, what is the question? *Nat. Rev. Neurosci.* **2021**, *22*, 55–67. [CrossRef]
24. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [CrossRef]
25. Tyagi, A.K.; Abraham, A. *Recurrent Neural Networks: Concepts and Applications*; The Mathworks, Inc.: Natick, MA, USA, 2022.
26. Egan, S.; Fedorko, W.; Lister, A.; Pearkes, J.; Gay, C. Long Short-Term Memory (LSTM) networks with jet constituents for boosted top tagging at the LHC. *arXiv* **2017**, arXiv:1711.09059.
27. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
28. Ghubaish, A.; Salman, T.; Zolanvari, M.; Unal, D.; Al-Ali, A.; Jain, R. Recent advances in the internet-of-medical-things (IoMT) systems security. *IEEE Internet Things J.* **2020**, *8*, 8707–8718. [CrossRef]
29. Trivedi, S.; Patel, N.; Faruqui, N. Bacterial Strain Classification using Convolutional Neural Network for Automatic Bacterial Disease Diagnosis. In Proceedings of the 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 19–20 January 2023; pp. 325–332.
30. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
31. Jadhav, S.; Zhao, J.; Fan, Y.; Li, J.; Lin, H.; Yan, C.; Chen, M. Time-Varying Sequence Model. *Mathematics* **2023**, *11*, 336. [CrossRef]
32. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A tutorial into long short-term memory recurrent neural networks. *arXiv* **2019**, arXiv:1909.09586.
33. Nandy, S.; Adhikari, M.; Khan, M.A.; Menon, V.G.; Verma, S. An intrusion detection mechanism for secured IoMT framework based on swarm-neural network. *IEEE J. Biomed. Health Inform.* **2021**, *26*, 1969–1976. [CrossRef]
34. Li, N.; Xu, M.; Li, Q.; Liu, J.; Bao, S.; Li, Y.; Li, J.; Zheng, H. A review of security issues and solutions for precision health in Internet-of-Medical-Things systems. *Secur. Saf.* **2023**, *2*, 2022010. [CrossRef]
35. Ziya, T.; Karakose, M. Comparative study for deep reinforcement learning with CNN, RNN, and LSTM in autonomous navigation. In Proceedings of the 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 26–27 October 2020; pp. 1–5.
36. Finlay, S. *Artificial Intelligence and Machine Learning for Business: A No-Nonsense Guide to Data-Driven Technologies*, 4th ed.; Relativistic: Bailrigg, UK; Lancaster, UK, 2021.
37. Qiu, J.; Wu, Q.; Ding, G.; Xu, Y.; Feng, S. A survey of machine learning for big data processing. *EURASIP J. Adv. Signal Process.* **2016**, *2016*, 1–16.
38. Rosay, A.; Cheval, E.; Carlier, F.; Leroux, P. Network intrusion detection: A comprehensive analysis of CIC-IDS2017. In Proceedings of the 8th International Conference on Information Systems Security and Privacy, Online, 9–11 February 2022; SCITEPRESS-Science and Technology Publications: Setúbal, Portugal, 2022; pp. 25–36.

39. Aktar, S.; Nur, A.Y. Towards DDoS Attack Detection using Deep Learning Approach. *Comput. Secur.* **2023**, *129*, 103251. [CrossRef]
40. Shanmugam, B.; Azam, S. Risk Assessment of Heterogeneous IoMT Devices: A Review. *Technologies* **2023**, *11*, 31.
41. Tarikere, S.; Donner, I.; Woods, D. Diagnosing a healthcare cybersecurity crisis: The impact of IoMT advancements and 5G. *Bus. Horiz.* **2021**, *64*, 799–807. [CrossRef]
42. Okey, O.D.; Maidin, S.S.; Adasme, P.; Lopes Rosa, R.; Saadi, M.; Carrillo Melgarejo, D.; Zegarra Rodríguez, D. BoostedEnML: Efficient technique for detecting cyberattacks in IoT systems using boosted ensemble machine learning. *Sensors* **2022**, *22*, 7409. [CrossRef] [PubMed]
43. Yu, L.; Zhou, R.; Chen, R.; Lai, K.K. Missing data preprocessing in credit classification: One-hot encoding or imputation? *Emerg. Mark. Financ. Trade* **2022**, *58*, 472–482. [CrossRef]
44. Thakkar, A.; Lohiya, R. Attack classification using feature selection techniques: A comparative study. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 1249–1266. [CrossRef]
45. Lippi, M.; Montemurro, M.A.; Degli Esposti, M.; Cristadoro, G. Natural language statistical features of LSTM-generated texts. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3326–3337. [CrossRef] [PubMed]
46. Nataraj, L.; Karthikeyan, S.; Jacob, G.; Manjunath, B.S. Malware images: Visualization and automatic classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security, Pittsburgh, PA, USA, 20 July 2011; pp. 1–7.
47. Hwang, J.W.; Lee, H.S. Adaptive image interpolation based on local gradient features. *IEEE Signal Process. Lett.* **2004**, *11*, 359–362. [CrossRef]
48. Silvestre-Blanes, J.; Sempere-Payá, V.; Albero-Albero, T. Smart sensor architectures for multimedia sensing in iomt. *Sensors* **2020**, *20*, 1400. [CrossRef] [PubMed]
49. Karimi, M.; Harouni, M.; Jazi, E.I.; Nasr, A.; Azizi, N. Improving Monitoring and Controlling Parameters for Alzheimer's Patients Based on IoMT. In *Prognostic Models in Healthcare: AI and Statistical Approaches*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 213–237.
50. Faruqui, N.; Yousuf, M.A.; Whaiduzzaman, M.; Azad, A.; Barros, A.; Moni, M.A. LungNet: A hybrid deep-CNN model for lung cancer diagnosis using CT and wearable sensor-based medical IoT data. *Comput. Biol. Med.* **2021**, *139*, 104961. [CrossRef]
51. Koutras, D.; Stergiopoulos, G.; Dasaklis, T.; Kotzanikolaou, P.; Glynos, D.; Douligeris, C. Security in IoMT communications: A survey. *Sensors* **2020**, *20*, 4828. [CrossRef]
52. Gerodimos, A.; Maglaras, L.; Ferrag, M.A.; Ayres, N.; Kantzavelou, I. IOT: Communication protocols and security threats. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 1–13. [CrossRef]
53. Sharma, S.; Ganguly, C.; De, S. Effect of Polarization on RF Signal Transmission over Two-Ray Channel. In Proceedings of the 2023 National Conference on Communications (NCC), IEEE, Guwahati, India, 23–26 February 2023; pp. 1–6.
54. Singarimbun, R.N.; Nababan, E.B.; Sitompul, O.S. Adaptive moment estimation to minimize square error in backpropagation algorithm. In Proceedings of the 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM), IEEE, Medan, Indonesia, 28–29 November 2019; pp. 1–7.
55. Li, S.; Jia, K.; Wen, Y.; Liu, T.; Tao, D. Orthogonal deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1352–1368. [CrossRef]
56. Lydia, A.; Francis, S. Adagrad—An optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci.* **2019**, *6*, 566–568.
57. Österlind, F.; Eriksson, J.; Dunkels, A. Cooja TimeLine: A power visualizer for sensor network simulation. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, Zürich, Switzerland, 3–5 November 2010; pp. 385–386.
58. Trivedi, S.; Patel, N.; Faruqui, N. NDNN based U-Net: An Innovative 3D Brain Tumor Segmentation Method. In Proceedings of the 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 26–29 October 2022; pp. 0538–0546.
59. Ravi, V.; Pham, T.D.; Alazab, M. Deep Learning-Based Network Intrusion Detection System for Internet of Medical Things. *IEEE Internet Things Mag.* **2023**, *6*, 50–54. [CrossRef]
60. Alalhareth, M.; Hong, S.C. An Improved Mutual Information Feature Selection Technique for Intrusion Detection Systems in the Internet of Medical Things. *Sensors* **2023**, *23*, 4971. [CrossRef] [PubMed]
61. Lu, W.; Burnett, B.; Phipps, R. Feature Selections for Detecting Intrusions on the Internet of Medical Things. In *Proceedings of the International Conference on Wireless Intelligent and Distributed Environment for Communication*; Springer: Cham, Switzerland, 2023; pp. 89–105.
62. Saif, S.; Yasmin, N.; Biswas, S. Feature engineering based performance analysis of ML and DL algorithms for Botnet attack detection in IoMT. *Int. J. Syst. Assur. Eng. Manag.* **2023**, *14*, 512–522. [CrossRef]
63. Finlayson, S.G.; Bowers, J.D.; Ito, J.; Zittrain, J.L.; Beam, A.L.; Kohane, I.S. Adversarial attacks on medical machine learning. *Science* **2019**, *363*, 1287–1289. [CrossRef]
64. Achar, S.; Faruqui, N.; Whaiduzzaman, M.; Awajan, A.; Alazab, M. Cyber-Physical System Security Based on Human Activity Recognition through IoT Cloud Computing. *Electronics* **2023**, *12*, 1892. [CrossRef]