ISSN 2007-9737

A Machine and Deep Learning Approach for Intrusion Detection and Attack Classification in Medical IoT Networks

Maryam Shabbir[1,2], Abdullah Abdullah[1,3,*] Fatima Shabbir[2]

[1] Bahria University,
Department of Computer Science,
Pakistan

[2] Punjab University College of Information Technology,
Computer Science Department,
Pakistan

[3] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

maryamshabbir.bulc@bahria.edu.pk, abdullah2025@cic.ipn.mx, mscsf24m008@pucit.edu.pk

Abstract. In the domain of the Internet of Medical Things (IoMT), upholding the confidentiality, integrity, and availability of sensitive medical data is of utmost importance. However, the intricate network of interconnected IoMT devices presents formidable challenges in effectively identifying intrusions and categorizing attacks. This research project focuses on harnessing the capabilities of both machine learning and deep learning techniques to develop robust systems for intrusion detection and attack classification, specifically tailored to IoMT environments. Through the implementation of cutting-edge algorithms and methodologies, our objective is to strengthen the security framework of IoMT systems, thus ensuring the protection of patient data against unauthorized access, tampering, and disruptions in service. By conducting thorough experimentation and analysis, we aim to assess the performance of various models and methodologies, with the ultimate aim of achieving high levels of detection accuracy while minimizing false positives and false negatives. Ultimately, our research endeavors to drive forward progress in IoMT security, contributing to the creation of safer and more dependable healthcare delivery systems.

Keywords. Intrusion detection, confidentiality, integrity, availability (CIA), machine learning.

1 Introduction

The emergence of the Internet of Medical Things (IoMT) represents a pivotal shift in the healthcare landscape, offering unprecedented opportunities for enhanced patient care, treatment efficacy, and operational efficiency [37]. By interconnecting medical devices, sensors, and systems, IoMT enables real-time data monitoring, remote patient management, and personalized healthcare delivery [26].

However, alongside these advancements come significant security and privacy challenges, necessitating robust measures to protect sensitive medical data and ensure the integrity and availability of healthcare services. In recent years, there has been a notable increase in the adoption of IoMT technologies, driven by advancements in wearable devices, implantable sensors, and remote monitoring systems.

These innovations empower healthcare providers with unparalleled access to patient data, facilitating proactive interventions, timely diagnoses, and customized treatment plans [35]. Additionally, IoMT has the potential to enhance patient engagement, enabling individuals to

actively participate in their healthcare journey while alleviating strain on traditional healthcare infrastructure [6].

Despite its promising benefits, the widespread implementation of IoMT is hindered by persistent security concerns. The interconnected nature of IoMT devices, coupled with the growing sophistication of cyber threats targeting healthcare networks, poses significant risks to patient data, including unauthorized access, data breaches, and malicious attacks [36].

Furthermore, the dynamic and diverse nature of IoMT ecosystems complicates security management, necessitating adaptive and scalable approaches to effectively mitigate emerging threats [25].

This paper aims to address the overarching challenge of enhancing the security posture of IoMT environments, with a specific focus on intrusion detection and attack classification. Traditional security measures, such as firewalls and encryption protocols, offer limited protection against advanced cyber threats targeting IoMT networks [11].

Consequently, there is a critical need for advanced security solutions capable of identifying and mitigating intrusions in real-time, thereby safeguarding the confidentiality, integrity, and availability of patient data and healthcare services.

The research is guided by the following ressearch questions:

a) How can machine learning and deep learning techniques be effectively leveraged to strengthen intrusion detection and attack classification in IoMT environments?

b) What are the primary challenges and considerations associated with implementing robust security measures in IoMT ecosystems?

c) How do different models and methodologies compare in terms of their effectiveness in detecting and mitigating intrusions in IoMT networks?

d) What are the broader implications of our findings for the future development and deployment of secure IoMT systems?

This paper is structured as follows such as the first section provides an overview of the background and context of IoMT technologies, emphasizing their transformative potential and the associated security challenges. The subsequent section delves into the specific problem statement, highlighting the critical need for enhanced security measures in IoMT environments, particularly focusing on intrusion detection and attack classification.

Following that, the research questions guiding our investigation are delineated, providing a framework for our analysis and discussion. Subsequent sections detail our methodology, experimental setup, and findings, culminating in a comprehensive evaluation of various intrusion detection models and methodologies. Finally, we discuss the implications of our research findings, identify avenues for future research, and conclude with recommendations for effectively securing IoMT environments.

2 Related Work

To work on Intrusion detection, the implementation of a thorough strategy incorporating short-term and long-term solutions is crucial. The coordination of diverse measures is crucial for the successful accomplishment of the required results. Furthermore, the establishment of monitoring and evaluation mechanisms is necessary to check the effectiveness of the strategy and make the necessary adjustments.

Artificial Intelligence (AI) has become deeply integrated into daily life, enhancing both personal and professional domains through advanced analytical and predictive capabilities. In the realm of communication and well-being, machine learning techniques are employed to foster safer digital environments by detecting cyberbullying on social media [7].

This analytical power extends to personalized services, where AI optimizes business-consumer interactions by tailoring email marketing campaigns for higher engagement [24]. The impact of AI is particularly profound in healthcare, where innovative frameworks enable critical tasks such as heartbeat classification for cardiac monitoring [4]

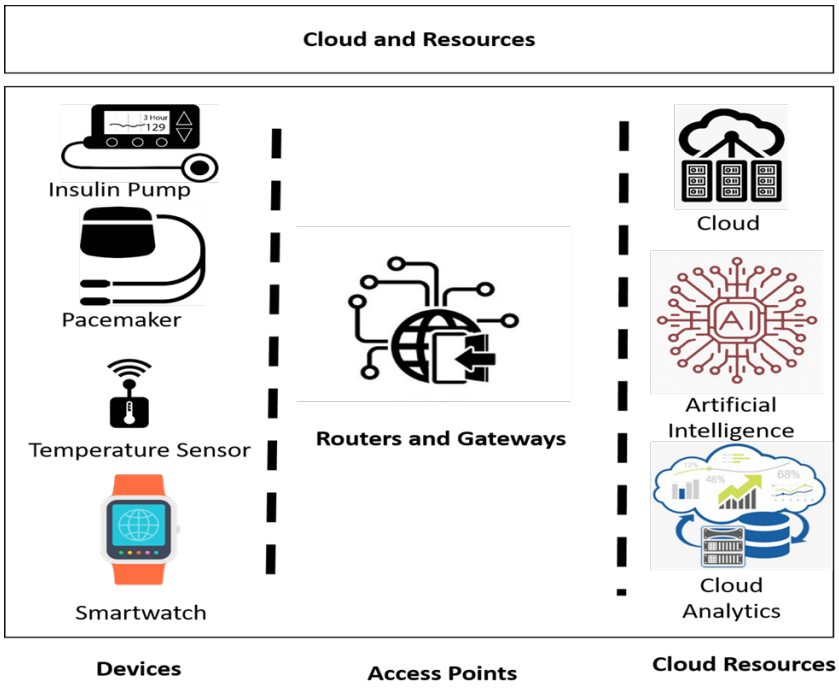

Fig. 1. IoMT architecture flow

and the prediction of breast cancer from clinical data to support early diagnosis [12].

Furthermore, AI models address complex systemic challenges that affect daily life, from forecasting nutrition supply chains to ensure food security [2] and predicting protein interactions to accelerate drug discovery [3], to optimizing enterprise project selection for more efficient delivery of goods and services [5].

The architecture of IoMT (Internet of Medical Things) is structured into three layers, including the Device layer, the Fog layer, and the Cloud layer, as outlined by [30] refer to Figure 1. Within the Device layer, physical devices like wearables and medical sensors collect and transmit medical data. Acting as an intermediary, the Fog Layer connects the Device layer to the Cloud layer, which manages data reception, transmission, pre-processing, as well as security and data privacy.

The Cloud Layer serves as the central repository for storing and analyzing medical data, providing access to authorized users such as healthcare providers and researchers, as highlighted by [31].

The Fog Layer is commonly identified as a significant vulnerability point. To mitigate these vulnerabilities, the implementation of an intelligent Intrusion Detection System (IDS) becomes imperative. IDSs are categorized based on three criteria, as presented by [32], see Figure 2. These criteria include the data source, leading to classifications such as Host-based IDS (HIDS) and Network-based IDS (NIDS), with NIDS focusing on monitoring network activities to enhance overall security.

Examining traffic and recognizing intrusions involves analyzing traffic patterns, where Network-based IDS (NIDS) focuses on overall traffic analysis, while HIDS centers on individual hosts or endpoints to identify anomalies or attacks. Based on behavior, IDS is divided into Passive IDS and Active IDS. Passive IDS records attacks and may trigger alerts, whereas Active IDS modifies the environment to counteract threats.

In terms of detection methodology, IDS is categorized into Signature-based IDS (SIDS) and Anomaly-based IDS (AIDS). SIDS uses

Table 1. Machine learning and deep learning based IDS

Articles	Contribution
[27]	In the research conducted by Khan et al. (2023), they devised an Intrusion Detection System (IDS) using Recurrent Neural Network and Gated Recurrent Units (RNN-GRU), incorporating two optimization techniques: Adam and Adamax.
[41]	The study employed the Improved Squirrel Search Algorithm (ISSA) to model an IDS, coupling it with a Modified Deep Belief Network (MDBN) on the UNSW-NB15 dataset.
[29]	The research formulated Neural Network-based IDS, integrating the Extended Kalman Filter as a backpropagation strategy.
[9]	In the realm of IoT networks, authors focused on detecting DDoS and DoS attacks, utilizing the Conditional Tabular Generative Adversarial Network (CTGAN) within their IDS model.
[1]	The study embraced ensemble learning, employing Logistic Regression (LR), Naïve Bayes (NB), and Decision Tree (DT) for a comprehensive voting classifier.
[22]	The author ventured into developing a multiclass classification model using Convolutional Neural Network (CNN).
[34]	For optimal feature selection and probe categorization, the author implemented RF recursive feature elimination and leveraged SVM and Adaptive Neuro-Fuzzy System.
[8]	The author explored Deep Neural Network (DNN) implementation using the Whale Optimization Algorithm for effective feature selection.
[40]	The author conducted a comparative study involving k-Nearest Neighbor (KNN), DNN, Naïve Bayes (NB), Random Forest (RF), and SVM, incorporating feature engineering through Principal Component Analysis (PCA) and Grey Wolf.
[28]	The author designed the first five layers with static rule bases, and the sixth layer utilized a one-class Support Vector Machine (SVM).
[10]	The author introduced the Fog-Based Attack Detection (FBAD) framework, employing an Ensemble of Online Sequential Extreme Learning Machine (EOS-ELM).

known attack patterns or signatures, while AIDS employs machine learning to understand typical network behavior and detect deviations suggesting an attack.

The effective assessment of an IDS involves various metrics, such as the detection rate, the false discovery rate, the response time, and the scalability. Detection rate measures the IDS capability to identify intrusions, while the false alarm rate quantifies false positives generated.

Response time assesses how quickly the IDS detects and responds to an intrusion, and scalability evaluates its ability to handle substantial volumes of network traffic. The subsequent section offers insights into existing Machine Learning-based IDS.

Intrusion Detection Systems (IDS) utilize Machine Learning (ML) employing supervised, unsupervised, or semi-supervised learning techniques for scrutinizing network traffic data to detect potential intrusions. In the supervised paradigm, the IDS undergoes training on labeled data, classifying each data point as normal or indicative of an attack.

Conversely, unsupervised learning enables the IDS to discern the inherent structure of the data, pinpointing anomalies or outliers that may signal an attack. Semi-supervised learning integrates both labeled and unlabeled data for training, empowering the IDS to adapt to novel attack patterns and identify unknown attacks, surpassing the capabilities of traditional signature-based IDS.

Deep Learning (DL)-based IDS, utilizing neural networks, finds the insightful patterns and correlations within network traffic data. DL facilitates the development of an advanced IDS capable of discerning complex attacks, including zero-day attacks, polymorphic attacks, and stealthy attacks. Furthermore, DL-based IDS exhibits proficiency in handling high-dimensional data and identifying subtle changes in network traffic patterns that may elude traditional IDS detection. However, DL-based IDS demands a substantial amount of training data and may pose challenges of computational intensity.

Both ML and DL-driven IDSs exhibit promise in enhancing the efficiency of intrusion detection against cyber-attacks. The overview in Table 1 briefly outlines past attempts at intrusion detection systems, presenting various approaches utilizing ML and DL methods to address a common problem. Nevertheless, challenges exist for ML and DL-driven IDSs, including the need for properly labeled data and substantial computational resources.

DL-based IDSs are susceptible to adversarial attacks, where deviation from normal network traffic can lead to detection. To address these challenges, the ensemble learning technique balances bias and variance by combining models with diverse strengths and weaknesses, as shown in Table 1. This approach not only enhances model interpretability but also integrates different feature selection or engineering approaches, as elucidated in the subsequent subsection.

3 Materials and Methods

The methodology employed by the ML-based IoMT system is outlined in Figure 3, encompassing three key stages: Data Pre-Processing, Training and Testing, and Performance Evaluation. The Network intrusion dataset is utilized, and the initial step involves addressing missing values.

Important features are then selected through a tree-based classifier, determining the relative importance score for each feature. Categorical features are subsequently transformed into numerical ones using label encoding and one-hot encoding.

Following data refinement, it is divided into training and testing subsets in a 7:3 proportion. A fivefold cross-validation technique is applied to the training dataset to optimize hyperparameters. Once optimal hyperparameter values are determined, the model is trained using the complete training dataset.

The trained model undergoes testing using the designated dataset, and its performance is evaluated using metrics like Accuracy, Precision, Recall, F1-Score, FDR, and FPR. This entire process is repeated for each classification model considered by the ML-based IoMT system. Moving forward, the dataset employed by the proposed ML-based IDS for IoMT is discussed below.

3.1 Dataset

We use a comprehensive Kaggle IoT and IIoT network traffic dataset from Kaggle, specifically the *Edge-IIoTset*, to train and evaluate our intrusion detection model. The data set captures diverse network traffic generated by multiple devices, sensors, and imaging equipment in a controlled environment that simulates real-world IoT scenarios, as detailed Tables 2 to Table 6 show all components of the data set, as shown in Figure 3.

It is suitable for both centralized and federated learning-based intrusion detection systems. The data set is structured into seven layers: Cloud Computing, Virtualization of Network Functions, Blockchain Network, Fog Computing, Software-Defined Networking, Edge Computing, and IoT/IIoT Perception [42].

The dataset includes data from more than ten types of devices, such as temperature/humidity sensors, ultrasonic sensors, water level detectors, pH meters, soil moisture sensors, heart rate sensors, and flame sensors. The collected traffic encompasses normal activity as well as fourteen

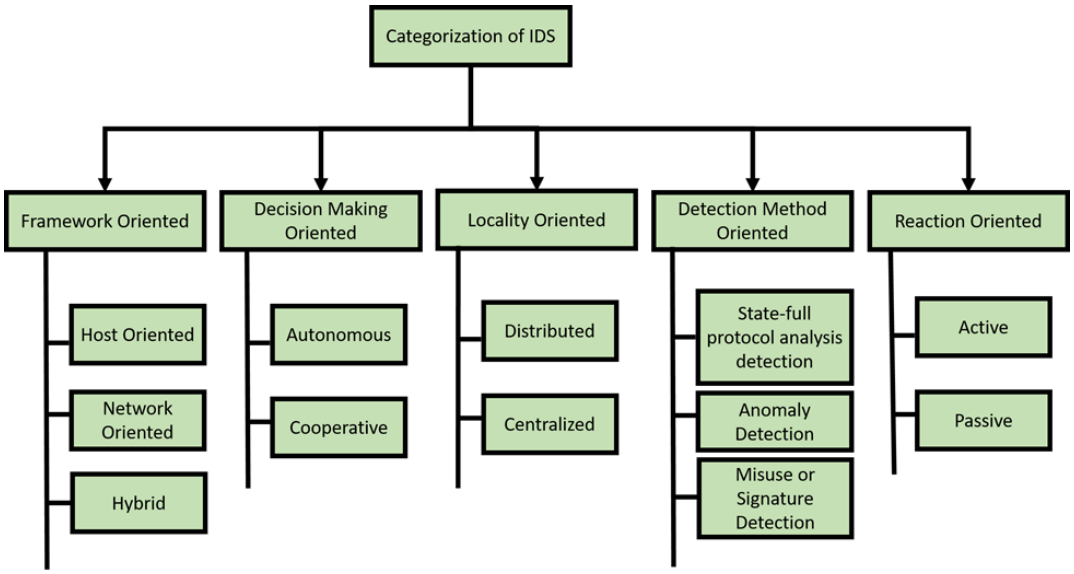

Fig. 2. IDS categorization hierarchy

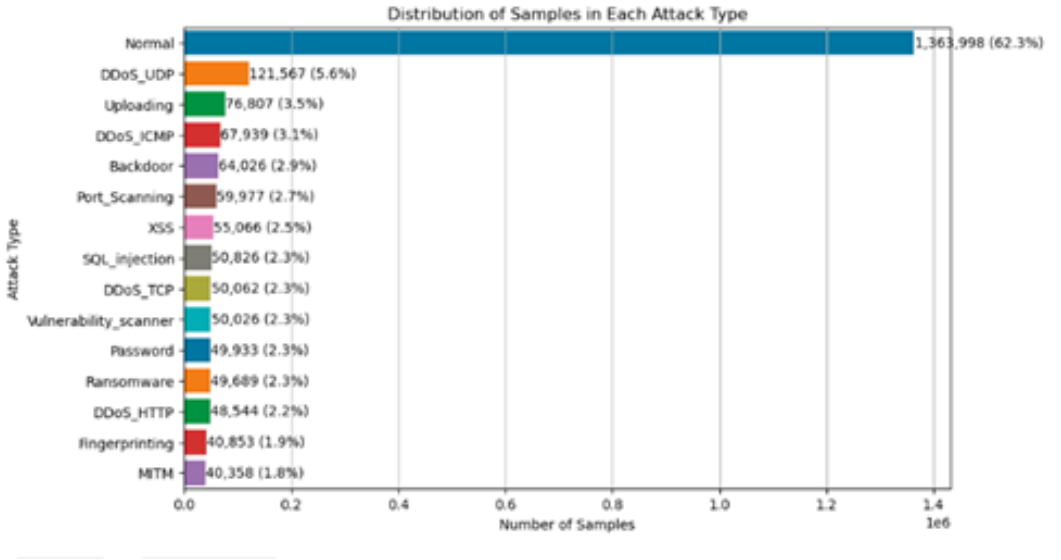

Fig. 3. Dataset detailed description

types of attacks, categorized into five main threat categories:

- DoS/DDoS attacks,
- Information gathering,
- Man-in-the-middle attacks,
- Injection attacks,
- Malware attacks.

The dataset contains the following features, suitable for training machine learning-based IDS models, presented in Table 3.

Table 2. Number of instances per attack type

Attack Type	Instances
Normal	1,615,643
DDoS_UDP	121,568
DDoS_ICMP	116,436
SQL_Injection	51,203
Password	50,153
Vulnerability_Scanner	50,110
DDoS_TCP	50,062
DDoS_HTTP	49,911
Uploading	37,634
Backdoor	24,862
Port_Scanning	22,564
XSS	15,915
Ransomware	10,925
MITM	1,214
Fingerprinting	1,001

Numeric features are summarized in Table 4.

The dataset can be utilized for:

— Training and testing supervised machine learning models for intrusion detection.

— Anomaly detection using unsupervised learning.

— Feature importance and network behavior analysis.

Being synthetically generated, the dataset simulates real-world network traffic patterns, but may not capture all environmental complexities. IP addresses should be anonymized, categorical features encoded, and numeric features scaled during preprocessing.

3.2 Preprocessing and Model Training

The ML-based Intrusion Detection System (IDS) designed for the Internet of Medical Things (IoMT) undergoes training using the *ToN_IoT Network dataset*. Various classification-based ML techniques, including Decision Tree, Ensemble Voting Classifier, Bagging, and Random Forest, are applied for the training of the dataset. A concise overview of these techniques is provided below, drawing insights from Geron as shown in Tables 7 and Table 8.

The ML-based Intrusion Detection System (IDS) crafted for the Internet of Medical Things (IoMT) undergoes training with the *Network intrusion dataset*. Various classification-oriented ML techniques, such as Multinomial Naive Bayes [33], Logistic Regression [18], Logistic Regression with Stochastic Gradient Descent [13], Linear Support Vector Classification [16], Logistic Regression [17], Decision Tree [38], Random Forest [14], and Adaptive Boosting [20], Gradient Boosting [21, 15], and Extreme Gradient Boosting [19] are utilized to train the dataset. A concise overview of these techniques is provided below [23, 39].

Multinomial Naive Bayes operates as a probabilistic classification algorithm widely employed in natural language processing tasks, including text classification and sentiment analysis. Its foundation in Bayes theorem allows it to calculate the probability of a given class based on a set of features. In the context of Multinomial Naive Bayes, these features are discrete, typically representing word counts or frequencies.

Conversely, Logistic Regression is a statistical technique used to analyze datasets featuring one or more independent variables influencing an outcome. Particularly suited for binary or categorical dependent variables, Logistic Regression utilizes a logistic function to transform a linear combination of input variables, producing a score that signifies the probability of a specific outcome.

Utilizing Stochastic Gradient Descent for parameter updates, Logistic Regression with Stochastic Gradient Descent offers a modified approach to traditional Logistic Regression. This method is particularly advantageous for extensive datasets, as it enables incremental updates to the model parameters instead of computing the gradient for the entire dataset in a single iteration. This makes it a practical and efficient choice for handling large volumes of data.

Logistic Regression with Stochastic Gradient Descent represents an altered iteration of the standard Logistic Regression, employing Stochastic Gradient Descent optimization for parameter updates. This modification proves especially advantageous when managing extensive datasets,

Table 3. Network traffic features

Feature	Type	Description
Duration	Numeric	Duration of the network flow (seconds)
Protocol	Categorical	Protocol used (TCP, UDP, ICMP)
SourceIP	String	Source IP address
DestinationIP	String	Destination IP address
SourcePort	Numeric	Source port number
DestinationPort	Numeric	Destination port number
PacketCount	Numeric	Number of packets in the flow
ByteCount	Numeric	Number of bytes transferred
Label	Categorical	Normal or Attack

Table 4. Summary statistics for numeric features

Feature	Min	Max	Mean	Std. Dev.
Duration (s)	0.01	100.0	50.3	28.8
SourcePort	1,098	65,461	32,900	18,500
DestinationPort	1	1,022	520	293
PacketCount	1	999	490	290
ByteCount	1,304	1,499,009	723,000	434,000

Table 5. Distribution of network protocols

Protocol	Percentage
TCP	35%
ICMP	33%
Other	32%

Table 6. Distribution of traffic classes

Label	Count (%)
Attack	1,020 (51%)
Normal	980 (49%)

facilitating incremental updates to the model parameters rather than computing the gradient for the entire dataset in a single pass. This approach enhances efficiency, making it a pragmatic choice for handling substantial volumes of data.

Linear Support Vector Classification is a linear classification algorithm designed to identify the hyperplane that best separates data points of distinct classes. The hyperplane, characterized by parameters learned during the training process, plays a crucial role in the algorithm functionality. The primary objective is to maximize the margin between the hyperplane and the nearest data points of each class, contributing to the algorithm efficiency in accurately classifying instances.

The Decision Tree stands as a straightforward yet potent classification algorithm, crafting a tree-like model that illustrates decisions and their potential consequences. Its functioning involves recursively dividing the data into subsets based on input variable values, continuing this process until the subsets achieve homogeneity regarding the target variable. This iterative approach enables Decision Trees to adeptly organize and depict decision-making processes in a structured and intelligible fashion.

The Ensemble Voting Classifier is an ensemble learning algorithm designed to merge predictions from multiple models using a voting mechanism. In the case of the Hard Vote Ensemble, also known as majority voting, the prediction involves selecting the class that receives the highest number of votes from the base models. On the other hand, the Soft Vote Ensemble computes probabilities for each class predicted by individual base models and then averages these probabilities across all models. The ultimate classification result identifies the class with the highest average probability.

Bagging is an ensemble learning technique that involves the utilization of multiple models trained on different subsets of the training data. Through bootstrapping, which consists of the random selection and training of models with replacement, Bagging introduces diversity among the models. The ultimate prediction is determined by averaging the predictions generated by each model. This approach enhances overall robustness by amalgamating diverse insights, resulting in a consolidated and reliable prediction.

Random Forest functions as an ensemble learning algorithm that merges multiple decision trees to create a strong classifier. This involves the random sampling of input variables and training data for each tree, followed by averaging the predictions of all the trees. This approach effectively mitigates overfitting, ultimately boosting the model accuracy.

Both Bagged Decision Trees (canonical bagging) and Random Forest generate multiple decision trees using bootstrap samples of the training data. The crucial distinction lies in how they select features for node splitting. Bagged decision trees consider all available features, whereas Random Forest uses a random subset, reducing correlation among trees and enhancing diversity to counter overfitting.

The Gradient Boosting Classifier, another potent ensemble learning algorithm, combines several weak learners (decision trees) to construct a robust classifier. It does so iteratively by adding decision trees aimed at correcting errors from previous ones, and the final prediction is derived by aggregating predictions from all the trees.

In the case of Extreme Gradient Boosting, it represents a specialized variant of Gradient Boosting that employs a more regularized model to prevent overfitting. It also introduces features like handling missing values and parallel processing.m The Gradient Boosting Classifier stands as a potent ensemble learning technique, amalgamating various weak learners, typically decision trees, to forge a robust classifier.

Its approach revolves around iteratively incorporating decision trees to rectify the errors of prior iterations, ultimately yielding a final prediction derived from the collective contributions of all trees involved. Extreme Gradient Boosting, a variant of Gradient Boosting, enhances this approach by implementing a more regularized model to mitigate overfitting. It also introduces additional functionalities such as handling missing values and facilitating parallel processing.

Adaptive Boosting, another ensemble learning method, combines multiple weak learners to create a resilient classifier. It operates by assigning weights to training data points and dynamically adjusting these weights based on the performance of preceding models. The final prediction emerges from a weighted fusion of predictions from all participating models.

The role of hyperparameters in shaping model performance is paramount. Optimal hyperparameter selection, established before model training, is critical for achieving optimal results.

These parameters encompass crucial aspects such as learning rate, regularization strength, and architectural specifications like the number of hidden layers in a neural network. In the context of the proposed Machine Learning-based Intrusion Detection System for the Internet of Medical Things (IoMT), hyperparameter tuning is a crucial step in selecting the most appropriate parameters for the machine learning algorithm.

Hyperparameter tuning involves subjecting each classification model to a rigorous fivefold cross-validation process on the training dataset. Through this process, adjustments are made and validated across various values, as specified in Table 3. Following validation, the classification models are tested using the optimal hyperparameter values. Subsequent evaluation of model performance provides valuable insights into their effectiveness, a discussion we delve into next.

3.3 Deep Learning Models

This section explores the utilization of deep learning architectures for intrusion detection and attack classification within the Internet of Medical Things (IoMT) environment. We explore a range of models, including XGBoost, Convolutional Neural Network (CNN), CNN coupled with Autoencoder, Gated Recurrent Unit (GRU), Long Short-Term

Table 7. Hyper-parameters in machine learning models

Model	Hyper-parameters
AdaBoosting	L_rate, No_of_estimators
Extreme_Gradient_Boosting (XGB)	Mx_depth, Gamma, Mn_child_wght, No_of_estimators, L_rate
Gradient_Boosting_Classifier (GBC)	Criteria, Mx_depth, L_rate, Subsample, No_of_estimators, Mn_sample_leaf, Loss_rate
Random_Forest (RF)	No_of_estimators, Mn_split, Mn_samples_Leaf, Mx_depth
Bagging_Algorithm	Mx_features, Mx_samples, No_of_estimators
Ensemble_Learning_Algorithm	Voting_param
Decision_Tree	Mn_sample_leaf, Mx_depth, Criteria, Mn_split
Linear_Support_Vector_Machine	Regularization_param, Loss_rate, Penalty
Logistic_Regression	Iterations, Regularization_param, Penalty
Logistic_Regression_with_Stochastic_Gradient_Descent	Class_wght, Eta, L_rate, Alpha_val, Penalty, Loss_rate
Naïve_Bayes	Alpha_val

Memory (LSTM), GRU combined with CNN, and Autoencoder integrated with either GRU or LSTM.

3.3.1 XGBoost

Although not classified as a deep learning model, XGBoost is widely recognized for its outstanding performance in classification tasks, rendering it a credible baseline for comparison. In our research, we employ XGBoost to proficiently classify IoMT network traffic data, thereby enabling the discernment of various intrusion categories.

3.3.2 Convolutional Neural Network (CNN)

Renowned for its prowess in image recognition, CNNs can be extended to analyze sequential data, making them apt for processing time-series signals like network traffic data. Our study uses CNNs to automatically extract relevant features from IoMT network traffic, thereby enhancing the effectiveness of intrusion detection and attack classification.

We design and train a CNN model to analyze the preprocessed network traffic data and classify it into different categories, such as normal traffic or various types of attacks. To implement the CNN, we first divide the preprocessed network traffic data into training and testing sets.

Model Training and Evaluation: We train the CNN model using the training set of preprocessed network traffic data. We use state-of-the-art large-scale CNN architectures, such as VGG16 and InceptionV3, which have been adopted and fine-tuned for our specific task of intrusion detection and attack classification on the Internet of Medical Things.

We utilize transfer learning techniques to leverage the pre-trained models on large-scale datasets for our specific medical IoT environment. Model training and evaluation involve feeding the preprocessed network traffic data into the CNN model and optimizing its parameters using techniques such as gradient descent and backpropagation.

To evaluate the performance of the trained CNN model, we use the testing set of preprocessed network traffic data. We evaluate the performance of the trained CNN model by measuring various metrics such as accuracy, precision, recall, F-measure, and FAR.

3.3.3 Convolutional Neural Network + Autoencoder

We explore the synergistic combination of CNN with Autoencoder to extract features and identify anomalies in IoMT network traffic. While CNN captures high-level features, the Autoencoder reconstructs input data, facilitating the identification of anomalous patterns deviating from typical IoMT network behavior.

3.3.4 Gated Recurrent Unit (GRU)

As a variant of recurrent neural networks (RNNs), GRU excels in capturing temporal dependencies within sequential data efficiently. Our research leverages GRU networks to analyze the temporal dynamics of IoMT network traffic, aiding in the detection of anomalies and classification of intrusion attempts based on sequential events.

3.3.5 Long Short-Term Memory (LSTM)

Similar to GRU, LSTM is proficient in learning long-term dependencies in sequential data. We employ LSTM networks to model temporal relationships in IoMT network traffic, facilitating the detection of abnormal patterns indicative of intrusion attempts. LSTM networks are a type of recurrent neural network that are well-suited for capturing temporal dependencies in sequential data, such as network traffic. To implement the LSTM, we preprocess the network traffic data by converting it into sequential input sequences.

Model Training and Evaluation: Similar to the CNN approach, we train the LSTM model using the training set of preprocessed network traffic data. We utilize LSTM architectures specifically designed for time series analysis, such as stacked LSTM or bidirectional LSTM. Model training and evaluation involve feeding the preprocessed sequential input sequences into the LSTM model and optimizing its parameters using techniques such as gradient descent and backpropagation.

To evaluate the performance of the trained LSTM model, we use the testing set of preprocessed sequential input sequences. We evaluate the performance of the trained LSTM model by measuring various metrics such as accuracy, precision, recall, and F-measure.

3.3.6 GRU + CNN

This hybrid architecture amalgamates the strengths of GRU and CNN models for intrusion detection in IoMT environments. While GRU captures temporal dependencies, CNN extracts spatial features from network traffic data, offering a holistic approach to intrusion detection and attack classification.

3.3.7 Autoencoder + GRU/LSTM

Autoencoders, utilized for dimensionality reduction and feature learning, are integrated with either GRU or LSTM networks to perform anomaly detection in IoMT network traffic. The Autoencoder learns to reconstruct normal patterns, thereby aiding in the detection of deviations indicative of intrusions or attacks.

4 Experiments and Results

In our experimental evaluations, we assess the performance of these deep learning models across various metrics such as accuracy, precision, recall, and F1-score, as shown in Tables 8 to Table 10.

Through comprehensive analyses, we evaluate the effectiveness of each model in accurately detecting and classifying intrusion attempts in IoMT environments, considering factors such as model complexity, training time, and computational resources required as shown in Table 9 and Table 10.

The experimental evaluation of the proposed deep learning models demonstrates their effectiveness in detecting and classifying intrusion attempts within IoMT environments. The key evaluation metrics employed include accuracy, precision,

Table 8. Detailed analysis of base machine learning models

Model	Loss	Accuracy	Recall	Precision	F1-Score	FAR
AdaBoosting	0.32	0.86	0.84	0.88	0.86	0.14
Extreme Gradient Boosting (XGB)	0.25	0.88	0.89	0.87	0.88	0.12
Gradient Boosting Classifier (GBC)	0.28	0.87	0.86	0.88	0.87	0.13
Random Forest (RF)	0.30	0.85	0.86	0.84	0.85	0.15
Bagging Algorithm	0.31	0.84	0.83	0.85	0.84	0.16
Ensemble Learning Algorithm	0.27	0.87	0.88	0.86	0.87	0.13
Decision Tree	0.33	0.81	0.80	0.82	0.81	0.18
Linear Support Vector Machine	0.38	0.79	0.81	0.77	0.79	0.21
Logistic Regression	0.34	0.83	0.85	0.81	0.83	0.17
Logistic Regression with Stochastic Gradient Descent	0.35	0.82	0.84	0.80	0.82	0.18
Naïve Bayes	0.42	0.75	0.76	0.74	0.75	0.25

Table 9. Detailed analysis of fine tuned models

	CNN	CNN+Autoencoder	GRU	LSTM	XGBoost	GRU+CNN	Autoencoder+GRU	Autoencoder+LSTM
Loss	0.0995	0.111	0.124	0.1260	0.067	0.0099	1.2600	0.233
Accuracy	0.949	0.9469	0.9398	0.9393	0.964	0.9494	0.7142	0.9150
Recall	0.923	0.9219	0.9176	0.9128	0.965	0.9228	0.7142	0.8795
Precision	0.983	0.9810	0.9702	0.9745	0.985	0.9849	0.7142	0.9705
F1-Score	0.951	0.9500	0.9428	0.9421	0.960	0.9523	0.7142	0.9220
FAR	0.001	0.0012	0.0020	0.0017	0.0023	0.001	0.0204	0.001

Table 10. Overall model's evaluation

Model	Results
XGBoost	96%
Convolutional Neural Network	94.9%
Convolutional Neural Network + Autoencoder	90%
GRU	93.9%
LSTM	93.9%
GRU + CNN	94.9%
Autoencoder + GRU	71.4%
Autoencoder + LSTM	91.5%

recall, F1-score, false positive rate (FAR), and false discovery rate (FDR).

Overall model performance, as presented in the Table, indicates that XGBoost achieved the highest classification accuracy of 96%, followed closely by CNN and GRU+CNN models at 94.9%. The hybrid architectures, such as CNN+Autoencoder and Autoencoder+LSTM, exhibit slightly lower performance, with Autoencoder+GRU showing the lowest accuracy of 71.4%.

Detailed analysis of each model across all metrics, highlighting that CNN and GRU-based models maintain a favorable balance between accuracy, precision, recall, and F1-score, while the FAR remains consistently low for most models.

These results demonstrate that deep learning architectures, particularly CNN and GRU hybrids, offer robust and reliable solutions for intrusion detection in IoMT networks, balancing detection capability with computational efficiency.

The performance of the models was evaluated using several standard metrics. These metrics and their corresponding mathematical definitions are given below as shown in Equations 1 to Equation 6:

— **Accuracy (Maximize):**

$$\text{Accuracy} = \frac{NTP + NTN}{NTP + NTN + NFP + NFN}, \tag{1}$$

— **F1 Score (Maximize):**

$$F_1 = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}}, \tag{2}$$

— **Recall (Maximize):**

$$\text{Recall} = \frac{NTP}{NTP + NFN}, \tag{3}$$

— **Precision (Maximize):**

$$\text{Precision} = \frac{NTP}{NTP + NFP}, \tag{4}$$

— **False Positive Rate (Minimize):**

$$\text{FPR} = \frac{NFP}{NTP + NFP}, \tag{5}$$

— **False Discovery Rate (Minimize):**

$$\text{FDR} = \frac{NFP}{NTN + NFP}. \tag{6}$$

5 Discussion and Conclusion

The increasing threat of cyber attacks targeting the healthcare sector emphasizes the urgent need for robust Intrusion Detection Systems (IDS). This research explores the IDS model based on machine learning techniques, which was meticulously trained and evaluated using the Edge-IIoT dataset, specifically designed to emulate cyber attacks.

To ensure the reliability of the data, extensive preprocessing steps were undertaken, including the removal of columns with significant null values. Furthermore, essential features were identified using a tree-based classifier, assigning importance scores to each feature, while categorical data underwent conversion into numerical values through one-hot encoding.

Subsequently, hyperparameter tuning was conducted with fivefold cross-validation across a range of classification models, including Multinomial Naive Bayes, Logistic Regression, Logistic Regression with Stochastic Gradient Descent, Linear Support Vector Classification, Decision Tree, Ensemble Voting Classifier, Bagging, Random Forest, Adaptive Boosting, Gradient Boosting, and Extreme Gradient Boosting.

The primary aim was to identify the optimal hyperparameter values for each model. Following this, the performance of the models was thoroughly assessed using various metrics such as accuracy, precision, recall, F1-score, False Discovery Rate (FDR), and False Positive Rate (FPR).

The experimental findings underscored the efficacy of the Adaptive Boosting ensemble learning technique, which consistently outperformed other classification methods across all performance metrics. Moreover, when compared to existing models based on the ToN_IoT dataset, the Adaptive Boosting-based IDS for IoMT demonstrated superior performance in terms of accuracy, precision, FPR, and FDR.

This underscores the potential of the Adaptive Boosting-based intrusion detection model for deployment within the IoMT architecture, facilitating enhanced network traffic analysis to effectively serve as an IDS. The proposed IDS for IoMT holds significant promise for improving access to high-quality healthcare, particularly benefiting individuals in remote or underserved areas.

The societal impact of IDS for IoMT extends to bolstering the reliability, privacy, and security of medical systems and devices. By safeguarding patient information, ensuring uninterrupted healthcare operations, and proactively mitigating cyber threats, the proposed IDS for IoMT stands as a vital tool in advancing healthcare delivery and

outcomes, ultimately enhancing patient well-being in our interconnected world.

References

1. **Abbas, A., Khan, M. A., Latif, S., Ajaz, M., Shah, A. A., Ahmad, J. (2022).** A new ensemble-based intrusion detection system for Internet of Things. Arabian Journal for Science and Engineering, Vol. 47, No. 2, pp. 1805–1819.
2. **Abdullah, Ather, M. A., Rodriguez, J. L. O., Sánchez-Mejorada, C. G., Ruiz, M. J. T., Tellez, R. Q. (2026).** A leakage-aware multimodal machine learning framework for nutrition supply–demand forecasting using temporal and spatial data fusion. Computers, Vol. 15, No. 3, pp. 156.
3. **Abdullah, Fatima, Z., Ather, M. A., Chanona-Hernandez, L., Rodríguez, J. L. O. (2026).** Hybrid dual-context prompted cross-attention framework with language model guidance for multi-label prediction of human off-target ligand–protein interactions. International Journal of Molecular Sciences, Vol. 27, No. 2, pp. 1126.
4. **Abdullah, Fatima, Z., Safder, H. A., Manzoor, M., Sánchez-Mejorada, C. G., Torres Ruiz, M. J., Quintero Téllez, R. (2026).** RhythmX™: An interpretable self-supervised contrastive learning framework for heartbeat classification. Technologies, Vol. 14, No. 3, pp. 148.
5. **Abdullah, Hafeez, N., Sánchez-Mejorada, C. G., Torres Ruiz, M. J., Quintero Téllez, R., Alex, E. A., Sidorov, G., Gelbukh, A. (2026).** Enhancing decision intelligence using hybrid machine learning framework with linear programming for enterprise project selection and portfolio optimization. AI, Vol. 7, No. 2, pp. 52.
6. **Abdullah, Hafeez, N., Ullah, F., Ather, M., Hasan, A., Gelbukh, A., Oropeza-Rodríguez, J., Sidorov, G., Kolesnikova, O. (2025).** Performance tradeoffs in adaptive hybrid encryption and decryption techniques security analysis for optimized protection in IoT-environmental data systems. Contemporary Mathematics, Vol. 6, No. 5, pp. 5368–7533.
7. **Abdullah, A., Ullah, F., Hafeez, N., Latif, I., Sidorov, G., Riveron, E. F., Gelbukh, A. (2025).** Cyberbullying detection on social media using machine learning techniques. Computación y Sistemas, Vol. 29, No. 3.
8. **Agarwal, A., Khari, M., Singh, R. (2022).** Detection of DDOS attack using deep learning model in cloud storage application. Wireless Personal Communications, Vol. 127, No. 1, pp. 419–439.
9. **Alabsi, B. A., Anbar, M., Rihan, S. D. A. (2023).** Conditional tabular generative adversarial based intrusion detection system for detecting DDoS and DoS attacks on the Internet of Things networks. Sensors, Vol. 23, No. 12, pp. 5644.
10. **Alrashdi, I., Alqazzaz, A., Alharthi, R., Aloufi, E., Zohdy, M. A., Ming, H. (2019).** FBAD: Fog-based attack detection for IoT healthcare in smart cities. 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, USA.
11. **Asif, M., Abbas, S., Khan, M. A., Fatima, M., Ali, A., et al. (2024).** Advanced phishing website detection using a hybrid model of LSTM and ANN. , pp. 222–226.
12. **Ather, M. A., Abdullah, Fatima, Z., Rodríguez, J. L. O., Sidorov, G. (2026).** An interpretable multi-dataset learning framework for breast cancer prediction using clinical and biomedical tabular data. Computers, Vol. 15, No. 2, pp. 97.
13. **Bottou, L. (2010).** Large-scale machine learning with Stochastic Gradient Descent. Proceedings of COMPSTAT'2010, pp. 177–186. DOI: 10.1007/978-3-7908-2604-3_16.
14. **Breiman, L. (2001).** Random Forests. Machine Learning, Vol. 45, No. 1, pp. 5–32. DOI: 10.1023/a:1010933404324.

15. **Chen, T., Guestrin, C. (2016).** XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. DOI: 10.1145/2939672.2939785.
16. **Cortes, C., Vapnik, V. (1995).** Support-Vector networks. Machine Learning, Vol. 20, No. 3, pp. 273–297. DOI: 10.1007/bf00994018.
17. **Cox, D. R. (1958).** The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), Vol. 20, No. 2, pp. 215–232. DOI: 10.1111/j.2517-6161.1958.tb00292.x.
18. **Cox, D. R. (1959).** The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), Vol. 21, No. 1, pp. 238–238. DOI: 10.1111/j.2517-6161.1959.tb00334.x.
19. **Elshaarawy, M. K. (2025).** Stacked-based hybrid gradient boosting models for estimating seepage from lined canals. Journal of Water Process Engineering, Vol. 70, pp. 106913.
20. **Freund, Y., Schapire, R. E. (1995).** A decision-theoretic generalization of On-Line Learning and an application to Boosting. Lecture Notes in Computer Science, pp. 23–37.
21. **Friedman, J. H. (2001).** Greedy function approximation: A Gradient Boosting Machine. The Annals of Statistics, Vol. 29, No. 5, pp. 1189–1232.
22. **Gao, W., Wang, M., Pei, Y., Li, F., Wang, C. (2026).** A lightweight multi-classification intrusion detection model for edge IoT networks. Electronics, Vol. 15, No. 5, pp. 938.
23. **Géron, A. (2022).** Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.
24. **Hafeez, N., Nasir, M. U., Shabbir, M., Mehmood, S., et al. (2024).** Personalized email marketing: A machine learning approach for higher engagement and conversion rates. 2024 Horizons of Information Technology and Engineering (HITE), IEEE, pp. 1–6.
25. **Haque, H. M. U., Ahmad, N., Aini, Q. U. A., Saeed, A., et al. (2025).** AGRITECH: A smart system for sustainable farming. VAWKUM Transactions on Computer Sciences, Vol. 13, No. 1, pp. 290–306.
26. **Haque, H. M. U., Hafeez, N., et al. (2024).** Formal modelling and verification of autonomous reasoning based flight simulation system. Lahore Garrison University Research Journal of Computer Science and Information Technology, Vol. 8, No. 1.
27. **Hnamte, V., Hussain, J. (2023).** DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. Telematics and Informatics Reports, pp. 100053. DOI: 10.1016/j.teler.2023.100053. Accessed 14 Mar. 2023.
28. **Kintzlinger, M., Cohen, A., Nissim, N., Rav-Acha, M., Khalameizer, V., Elovici, Y., Shahar, Y., Katz, A. (2020).** CardiWall: A trusted firewall for the detection of malicious clinical programming of Cardiac Implantable Electronic Devices. IEEE Access, Vol. 8, pp. 48123–48140.
29. **Kulkarni, D. D., Jaiswal, R. K. (2023).** An intrusion detection system using Extended Kalman Filter and neural networks for IoT networks. Journal of Network and Systems Management, Vol. 31, No. 3. DOI: 10.1007/s10922-023-09748-x.
30. **Kulshrestha, P., Vijay Kumar, T., Khari, M. (2023).** Intrusion detection system for Internet of Medical Things. In International Conference on Advances in IoT and Security with AI. pp. 293–303.
31. **Lee, J. D., Cha, H. S., Park, J. H., et al. (2021).** M-IDM: A multi-classification based intrusion detection model in healthcare IoT. Computers, Materials & Continua, Vol. 67, No. 2.
32. **Liu, M., Xue, Z., Xu, X., Zhong, C., Chen, J. (2018).** Host-based intrusion detection system with system calls: Review and future trends. ACM Computing Surveys, Vol. 51, No. 5, pp. 1–36.

33. **Mccallum, A., Nigam, K. (1998).** A comparison of event models for Naive Bayes text classification. pp. 41–48.

34. **Mehmood, M., Javed, T., Nebhen, J., Abbas, S., Abid, R., Bojja, G. R., Rizwan, M. (2022).** A hybrid approach for network intrusion detection. Computers, Materials, & Continua, Vol. 70, No. 1, pp. 91.

35. **Nabawy, R. M., Hassan-Ibrahim, M., Rabee-Kaseb, M. (2024).** Survey of mobile cloud computing security and privacy issues in healthcare. Computación y Sistemas, Vol. 28, No. 3, pp. 1017–1029.

36. **Olivares-Rojas, J. C., Reyes-Archundia, E., Gutiérrez-Gnecchi, J. A., Molina-Moreno, I., Méndez-Patiño, A., Cerda-Jacobo, J. (2023).** Cyber hygiene in smart metering systems. Computación y Sistemas, Vol. 27, No. 2, pp. 459–475.

37. **Padilla-Gomez, M., Gamboa-Cruzado, J., Távara-Aponte, S., Núñez-Meza, , Amayo-Gamboa, F., Arauco-Esquivel, S. (2024).** Systematic literature review on cybersecurity and its influence on cyber attacks targeting IoT devices. Computación y Sistemas, Vol. 28, No. 4, pp. 1847–1864.

38. **Quinlan, J. R. (1986).** Induction of decision trees. Machine Learning, Vol. 1, No. 1, pp. 81–106. DOI: 10.1007/bf00116251.

39. **Qureshi, R., Koo, I. (2026).** A comprehensive survey of cybersecurity threats and data privacy issues in healthcare systems. Applied Sciences, Vol. 16, No. 3, pp. 1511.

40. **RM, S. P., Maddikunta, P. K. R., Koppu, S., Gadekallu, T. R., Chowdhary, C. L., Alazab, M., et al. (2020).** An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. Computer Communications, Vol. 160, pp. 139–149.

41. **Sarkar, N., Keserwani, P. K., Govil, M. C. (2024).** A better and fast cloud intrusion detection system using improved Squirrel Search Algorithm and modified Deep Belief Network. Cluster Computing, Vol. 27, No. 2, pp. 1699–1718.

42. **Zia, M. (2024).** Network traffic data for intrusion detection. DOI: 10.34740/KAGGLE/DSV/8661173.

Article received on 19/09/2025; accepted on 12/02/2026.
**Corresponding author is Abdullah Abdullah.*