Computer Modeling in Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.062841

ARTICLE

SNN-IoMT: A Novel AI-Driven Model for Intrusion Detection in Internet of Medical Things

Mourad Benmalek[1,*,#], **Abdessamed Seddiki**[2,#] **and Kamel-Dine Haouam**[1]

[1]Computer Engineering Department, College of Engineering, Al Yamamah University, Riyadh, 11512, Saudi Arabia
[2]Ecole Nationale Supérieure d'Informatique, BP 68M, Oued-Smar, Algiers, 16309, Algeria

*Corresponding Author: Mourad Benmalek. Email: m_benmalek@yu.edu.sa

[#]These authors contributed equally to this work

Received: 29 December 2024; Accepted: 07 March 2025; Published: 11 April 2025

ABSTRACT: The Internet of Medical Things (IoMT) connects healthcare devices and sensors to the Internet, driving transformative advancements in healthcare delivery. However, expanding IoMT infrastructures face growing security threats, necessitating robust Intrusion Detection Systems (IDS). Maintaining the confidentiality of patient data is critical in AI-driven healthcare systems, especially when securing interconnected medical devices. This paper introduces SNN-IoMT (**S**tacked **N**eural **N**etwork Ensemble for **IoMT** Security), an AI-driven IDS framework designed to secure dynamic IoMT environments. Leveraging a stacked deep learning architecture combining Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM), the model optimizes data management and integration while ensuring system scalability and interoperability. Trained on the WUSTL-EHMS-2020 and IoT-Healthcare-Security datasets, SNN-IoMT surpasses existing IDS frameworks in accuracy, precision, and detecting novel threats. By addressing the primary challenges in AI-driven healthcare systems, including privacy, reliability, and ethical data management, our approach exemplifies the importance of AI to enhance security and trust in IoMT-enabled healthcare.

KEYWORDS: Healthcare; Internet of Medical Things; artificial intelligence; deep learning; intrusion detection system

1 Introduction

Internet of Things (IoT) encompasses a collection of networked physical objects and devices that incorporate sensors, software, and connectivity to support the sharing of information via the Internet [1]. The definition and scope of IoT continue to expand, but it generally includes smart consumer devices [2], homes [3], enterprises [4], utilities [5], vehicles [6], wearables [7], cities [8] and healthcare [9]. IoT allows devices to remotely monitor, collect, analyze, and share data, transforming traditional objects into intelligent interconnected endpoints [10].

The deployment of IoT technologies in healthcare, specifically through the integration of medical devices and equipment, is often termed the Internet of Medical Things (IoMT) [11]. Other terms are used in the literature to describe this concept, such as Internet of Healthcare Things (IoHT), Smart Healthcare, and Healthcare 4.0 [12].

In IoMT systems, sensors, wearables, and smart medical devices are integrated to allow continuous tracking, data collection, and exchange of data between patients and healthcare providers [13]. IoMT has experienced significant growth, particularly in the aftermath of COVID-19, as the demand for remote

healthcare supervision became increasingly critical [14]. Healthcare providers needed to track patient vitals and manage diseases from a distance to minimize infection risks and provide opportunities for instant interventions.

However, despite their benefits, the increasing number of medical devices introduces pressing security and ethical issues. The network attack surface is increasing, enabling hackers to exploit vulnerabilities within these networks. Such breaches allow the unauthorized retrieval of sensitive patient data and disruption of medical device functionality, raising significant privacy and data management concerns. In 2017, a White House report indicated that cyber-criminals had begun targeting the IoMT devices [15]. During the COVID-19 era, the healthcare sector became a prime target for cyber-attacks [16]. Accordingly, several security measures have been implemented to safeguard IoMT networks. These measures include key management, authentication, encryption, and intrusion detection systems [17].

Protecting patient privacy is a critical aspect of deploying AI-driven Intrusion Detection Systems (IDSs) in healthcare environments, ensuring that patients' sensitive information remains confidential while maintaining the system integrity. In this work, we focus on detecting intrusions in IoMT environments through the use of IDS. These systems are vital for the early detection of threats and the mitigation of potential harm. Traditional methods of intrusion detection often struggle with the complex data generated by IoMT. Additionally, they lack support for a wide range of advanced IoT protocols, such as Message Queuing Telemetry Transport (MQTT) [17]. Hence, more advanced approaches are required that not only enhance security but also ensure effective data integration and uphold ethical standards in data handling.

In this context, IDSs leveraging Machine Learning (ML) and Deep Learning (DL) techniques have surfaced as a potential solution. These methods can mitigate cyber-threats by analyzing network traffic data to differentiate between normal and abnormal activities.

Ensemble methods are a widely used approach in ML/DL, based on combining multiple models to address complex problems. By aggregating the predictions of several models, this paradigm outperforms individual models across various performance metrics, especially in terms of accuracy [18]. The most commonly used ensemble techniques involve Bagging, Boosting, and Stacking. Each method improves performance by reducing variance, reducing bias, or improving generalization. Bagging is a technique that combines predictions from multiple models by averaging them, assigning the same weight to each model's output [19]. Boosting combines models iteratively, where each new model assigns higher weights to poorly handled cases to address the errors made by previous ones [19]. Stacking is a technique that aggregates predictions from several base models and uses them as inputs to train a meta-model, to improve the overall accuracy [17].

Several previous studies have proposed innovative approaches to enhance IDSs capabilities in IoMT environments. Despite these advances, significant challenges and gaps persist in this domain. Firstly, there is a notable lack of datasets that capture network traffic generated by interconnected healthcare devices, limiting the development and evaluation of effective IDSs tailored for IoMT. Secondly, many of the proposed methods exhibit weaknesses in performance metrics, particularly in accuracy, which is critically important in the healthcare sector, a field where even minor inaccuracies can lead to serious consequences. Lastly, a considerable number of previous works have relied on datasets collected from standard networks or generic IoT infrastructures, instead of being specifically derived from IoMT environments. These challenges undermine the applicability and effectiveness of their solutions in real-world healthcare settings.

To address these challenges, we propose **SNN-IoMT** (**S**tacked **N**eural **N**etwork Ensemble for **IoMT** Security), a novel DL-based framework for identifying intrusions in IoMT systems. Our framework employs a stacked ensemble architecture that combines the distinctive strengths of multiple DL approaches:

Multi-Layer Perceptron (MLP) for pattern recognition, Convolutional Neural Networks (CNN) for feature extraction, Long Short-Term Memory (LSTM) for temporal pattern analysis, and Artificial Neural Networks (ANN) for complex data processing. This comprehensive integration aims to achieve highly accurate intrusion detection in medical device networks. Fig. 1 presents the key steps of our proposed framework.

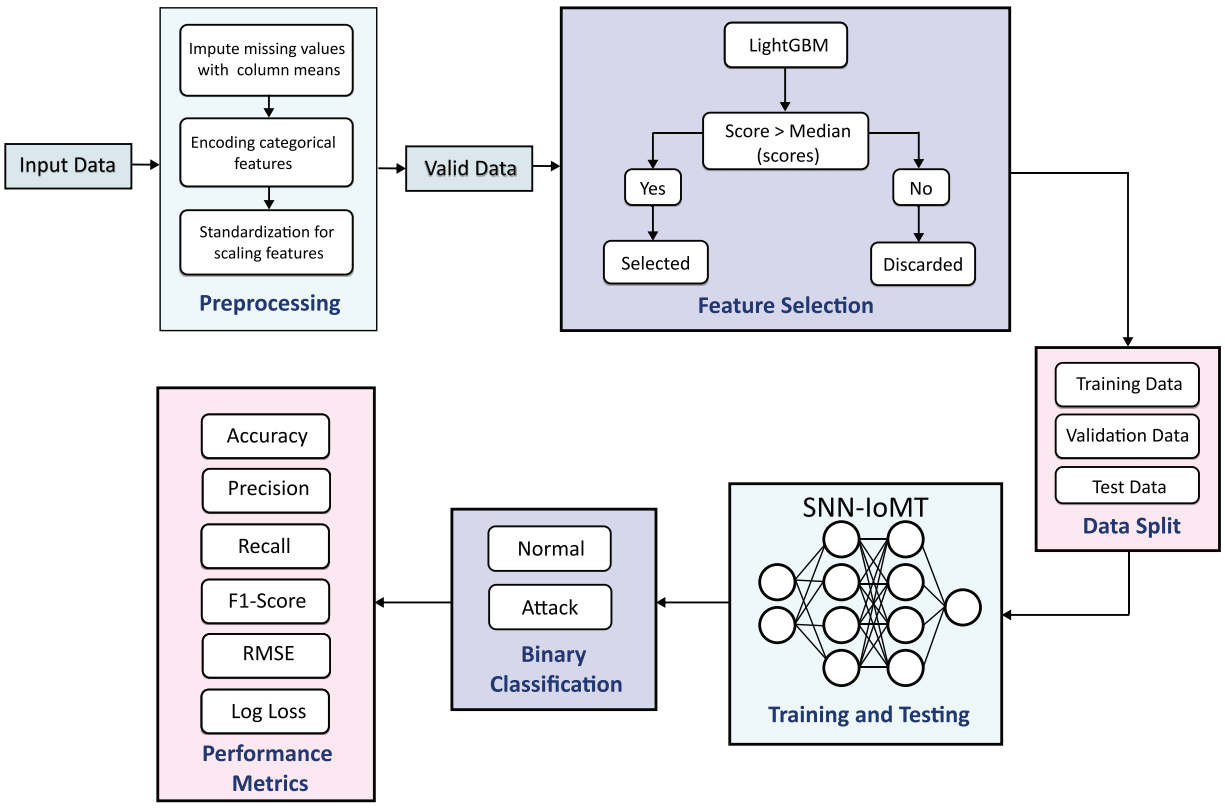


Figure 1: Flowchart of the proposed framework

Our work makes several key contributions:

- We introduce **SNN-IoMT**, a novel DL stacking model designed to enhance intrusion detection capabilities in interconnected medical networks.
- We implement a **LightGBM-based feature selection technique** to optimize our model's efficiency. This method employs gradient boosting algorithms to rank and filter the most important features depending on their contribution.
- We train the meta-learner using **k-fold cross-validation**, which improves the stacking ensemble's resilience and generalization. This technique enables extensive assessment and reduces overfitting, resulting in better model performance.
- We validate our model using two domain-specific datasets: **WUSTL-EHMS-2020** and **IoT-Healthcare-Security-Dataset**. Both datasets are generated from medical environments and have been previously used in IoMT intrusion detection research. Our extensive evaluation demonstrates that SNN-IoMT achieves superior performance compared to existing solutions across multiple metrics. The evaluation on two distinct datasets confirms the model's reliability and versatility in different IoMT use cases.

The paper is organized as follows: In Section 2, we highlight existing research on IDS that explores ML and DL techniques in IoMT environments, highlighting the strengths and weaknesses of each approach presented. Section 3 provides a background about IoMT and IDSs. In Section 4, we explain our new approach, called SNN-IoMT. Datasets, data preprocessing and feature selection are presented in Section 5. Section 6 provides an in-depth analysis of the model's robustness and validation process. Details of the results are provided in Section 7. Section 8 discusses the challenges and considerations for deploying the model in real-world IoMT environments. Section 9 is a conclusion that summarizes our findings.

2 Related Work

Recently, the fast-paced development of IoMT has brought several innovations to healthcare. Numerous studies have focused on developing IDSs for securing IoMT environments, leveraging ML and/or DL approaches to improve cyber threats detection. To enhance clarity, Table 1 contains the abbreviations used in this work.

Table 1: List of abbreviations

Abbreviation	Meaning
ADB	Adaptive Boosting
ANN	Artificial Neural Network
BG	Bagging
BiLSTM	Bidirectional Long Short-Term Memory
DT	Decision Tree
GBC	Gradient Boosting Classifier
KNN	K-Nearest Neighbors
LR	Logistic Regression
LRSGD	Logistic Regression with SGD
LSVC	Linear Support Vector Classifier
MNB	Multinomial Naive Bayes
NB	Naive Bayes
RF	Random Forest
SVM	Support Vector Machine
XGB	XGBoost

Hady et al. [20] introduced a new methodology utilizing KNN, SVM, RF, and ANN applied on the WUSTL-EHMS-2020 dataset. The RF model achieved a notable accuracy of 93%. This high accuracy is achieved due to its ensemble learning approach, which combines multiple DTs. Furthermore, RF ranks feature importance, allowing it to focus on the most relevant attributes. However, this level of accuracy remains insufficient in a critical healthcare domain.

Hussain et al. [15] created a new dataset called IoT-Healthcare-Security, and proposed a framework incorporating several ML models, including RF, KNN, DT, LR and NB. Among all models, the RF attained the highest accuracy of 99.51%. While this high level of accuracy is promising, the model must be trained on other datasets to demonstrate that it can be generalized across various IoMT infrastructures.

Saheed et al. [21] presented an IDS that employs a wrapper-based Particle Swarm Optimization (PSO) combined with ML models, leveraging the NSL-KDD dataset for detect anomalies within intelligent healthcare systems. Their framework included RF, KNN, DT, and ANN. Although the RF model achieved

an accuracy of 99.76%, the drawback lies in the fact that the dataset used is outdated and not derived from a healthcare system. As a result, the findings may not adequately reflect the challenges in current IoMT environments.

The NSL-KDD dataset was also used by Subasi et al. [22] to explore several ML algorithms, including SVM, RF, and bagging techniques. Their comparative analysis revealed that RF with bagging proved to be the most robust model, with an accuracy of 97.67%. Similarly to the previous work, the dataset used is outdated, and does not represent real IoMT traffic patterns.

A novel DL approach named Swarm-Neural Network (Swarm-NN) was proposed by Nandy et al. [23]. This model generates and trains a set of neural networks to predict malicious traffic in IoMT data. To evaluate their approach, the authors leveraged the ToN-IoT dataset. The accuracy achieved was 99.5%. Nevertheless, the dataset used is not derived from actual IoMT environments, raising concerns about the capacity of the model to adapt to real-life healthcare scenarios.

Binbusayyis et al. [24] applied several ML models including NB, DT, SVM DT, and ANN on the BoT-IoT dataset, aiming to develop an anomaly-based framework for threat detection in Smart Healthcare environments. Similar to the previous work, this dataset does not represent an IoMT network.

BlueTrack is a dataset introduced by Zubair et al. in [25]. It contains data about Bluetooth Low Energy (BLE) communications. This dataset was used to develop an IDS based on both ML and DL models. NB, LR, K-means, SVM, RF, and MLP are the leveraged models in their framework. Their results demonstrated that MLP outperformed others, achieving an accuracy of 99.8%. However, the dataset used has a limited scope in terms of attack variety, as it focuses only on BLE attacks. This limitation reduces its applicability to other communication protocols.

In [26], Gupta et al. used RF, LR, and DT to develop a model based on a novel tree classifier. They performed an analysis of average, standard deviation, 25th percentile, 75th percentile, and frequency statistics for selecting features and reducing the number of dimensions. Applied to the WUSTL-EHMS-2020 dataset, the RF model attained the best accuracy of 94.23%. Nevertheless, their proposed model still demonstrates a relatively low accuracy.

Wagan et al. [27] presented a new framework utilizing a BiLSTM model and the WUSTL-EHMS-2020 dataset to make the difference between normal and abnormal traffic in IoMT infrastructure. The BiLSTM architecture reached an accuracy of 92.95%. However, this accuracy is still insufficient in the healthcare domain, given its critical nature. In another work, Zukaib et al. [28] used three datasets: IoTID20, WUSTL-IIoT-2021, and WUSTL-EHMS-2020 to develop a new approach called Meta-IDS, which is based on weak learners (RF, AdaBoost, DT, MLP, MNB), combined with the Bat algorithm, a meta-heuristic algorithm to optimize the hyperparameters of weak learners. Additionally, another meta-learner was incorporated to enhance the accuracy by learning from pre-trained weak learners. The highest accuracy achieved by Meta-IDS was 99.99% with the WUSTL-IIoT-2021 dataset. However, the authors mentioned that, despite the very high accuracy achieved, their approach requires substantial computational resources, which is a notable limitation of the methodology.

An ML-based IDS was designed by Kulshrestha et al. [29] using several classifiers including MNB, LR, LRSGD, LSVC, BG, GBC, RF, ADB, and XGB. Applying these models to the ToN-IoT dataset, ADB achieved the best accuracy of 99.18%. Nevertheless, the dataset employed in this work is not derived from an IoMT network, which increases the concern about its applicability in real-world healthcare scenarios. However, this dataset does not represent IoMT network traffic patterns.

Table 2 presents a summary of existing ML/DL-based IDSs for IoMT, the highest metrics achieved by the best classifier, along with the advantages and limitations of each approach.

Table 2: Summary of related work

Work	Dataset	Classifier	Metrics				Pros (+) and Cons (-)
			M_1	M_2	M_3	M_4	
[20]	WUSTL-EHMS-2020	ANN	90.04%	–	–	–	+ Simple effective classifier - Accuracy still insufficient
[15]	IoT healthcare security	RF	99.51%	99.70%	99.79%	99.65%	+ High accuracy with novel dataset - Lack of validation on diverse datasets
[21]	NSL-KDD	RF	99.76%	99.75%	96.45%	96.45%	+ High accuracy using wrapper-based PSO - Not validated in real IoMT dataset
[22]	NSL-KDD	Bagging	97.67%	–	–	97.7%	+ Robust comparative analysis - Not validated in real IoMT dataset
[23]	ToN-IoT	Swarm-NN	99.5%	–	–	–	+ Innovative DL approach - Not validated in real IoMT dataset
[24]	BoT-IoT	DT	100%	–	–	–	+ Perfect accuracy - Not realistic dataset for IoMT
[25]	BlueTrack	MLP	99.8%	99.7%	99.06%	99.38%	+ High accuracy with MLP model - Limited scope to BLE attacks
[26]	WUSTL-EHMS- 2020	RF	94.23%	–	–	93.8%	+ Comprehensive feature analysis - Weak overall accuracy
[27]	WUSTL-EHMS- 2020	BiLSTM	92.95%	91.61%	95.64%	95.64%	+ Advanced LSTM architecture - Insufficient accuracy for healthcare
[28]	WUSTL-EHSM- 2020	Meta-IDS	99.57%	99.57%	99.57%	99.56%	+ High accuracy - High computational resource requirement
[28]	WUSTL-IIoT- 2021	Meta-IDS	99.99%	99.99%	99.99%	99.99%	+ High accuracy - High computational resource requirement
[28]	IoTID20	Meta-IDS	99.91%	99.93%	99.91%	99.91%	+ High accuracy - High computational resource requirement
[29]	ToN-IoT	ADB	99.18%	98.68%	98.98%	98.83%	+ High accuracy - Not validated in real IoMT dataset

Note: **Metrics: M_1:** Accuracy, **M_2:** Precision, **M_3:** Recall, **M_4:** F1-Score.

3 Background

This section provides the foundational concepts and necessary background information relevant to our study, including the IoMT architecture and the fundamentals of IDSs.

3.1 IoMT Architectures

IoMT is a set of interconnected biomedical devices and tools, seamlessly integrated with various software applications. These technologies are designed to communicate with healthcare infrastructures, facilitating the efficient management and exchange of healthcare data [30]. An IoMT network is based on several layers, that contribute to the systematic collection, processing, and application of healthcare [30]. Fig. 2 illustrates the multi-level of this architecture. These layers can be subdivided as follows:

1. **Perception Layer:** This bottom physical layer works at the edge level. It plays a crucial role in collecting information from various sources such as wearable (glucose monitors, smartwatches) or other smart medical devices. At this level, Patient data is gathered through sensors and sent to other layers for additional evaluation. For example, wearable Electrocardiogram (ECG) monitors track heart activity rates, and send the collected signals in real-time to other layers. Similarly, smart insulin pumps measure glucose levels and adjust insulin delivery.
2. **Network Layer:** This layer ensures reliable data transmission, including communication protocols (short-range protocols like Bluetooth, Zigbee, Z-Wave, 6LoWPAN, and longer-range protocols like WiFi, 2G/3G/4G/5G cellular, Low Power Wide Area Network (LPWAN)), gateways, and cloud infrastructure. This layer is susceptible to many types of network threats such as Transmission Control

Protocol Synchronize (TCP SYN) Flood, ARP Spoofing, Hijacking, eavesdropping, and ransomware, necessitating robust security measures for this layer. For instance, a smart hospital may use WiFi or 5G to transmit a patient's vital signs from perception layer devices to databases. Any disruption in this layer could affect medical interventions.

3. **Application Layer:** This important layer comprises software applications that enable the analysis, visualization, and generation of insights from data. Since it directly interacts with end-users and critical services, it is vulnerable to application-level attacks such as unauthorized access, malicious software, and data breaches. For example, a telemedicine application processes and displays ECG signals. If this layer is compromised, sensitive data could be leaked or manipulated.

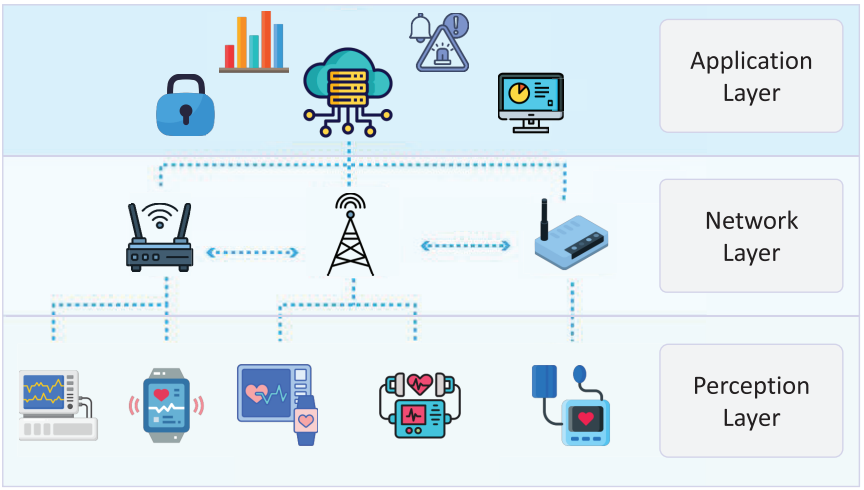


Figure 2: IoMT architecture

3.2 Intrusion Detection System (IDS)

An IDS serves as a system aimed at securing and protecting activities within networks and systems from malicious actions, unauthorized access attempts, and policy violations [31]. IDSs are used in various digital environments, including networks, applications, and host systems, to ensure the security of essential data and infrastructure. IDSs have emerged as a vital component in cybersecurity strategies for organizations to protect critical data, prevent disruption, and safeguard digital infrastructure. IDSs have two primary functions: detection and response. Once a threat is detected, the system alerts security teams, allowing rapid responses or mitigation actions to prevent damages and data loss. There are two notable types of IDSs:

- **Signature-based IDS:** Identifies attacks by examining data for predefined signatures or attack patterns. It compares the new activities with a database of signatures or known attacks such as malware types, virus signatures, or specific exploit patterns. Its strength lies in its ability to identify known threats but is poor at detecting new/unknown threats as there are no signatures to match them against.
- **Anomaly-based IDS:** Detects adversarial behavior by measuring current activity against normal activities. A baseline is usually created through ML models or statistical analysis and changes when a system is constantly monitored. It is based on heuristic techniques that efficiently identify zero-day and unknown attacks, hence authentication in dynamic environments. This solution is also more adaptive to changing network conditions or behaviors.

In this regard, IDSs provide a critical service for IoMT environments, protecting the security of patients and the integrity of medical devices. Considering the criticality of healthcare data and patient safety, anomaly-based detection is usually employed in IDSs focused on IoMT, which are a critical defense layer for IoMT infrastructure security.

4 Proposed Model

Our model, SNN-IoMT, is a stacking ensemble architecture that integrates three types of neural networks, categorized into two levels:

- Level 0: Base learners
- Level 1: Meta-learner

The classification outputs from the base learners serve as inputs for the meta-learner. This architecture allows combining the strengths of various heterogeneous model types to improve classification quality.

SNN-IoMT employs three distinct models in Level 0. These models include an MLP, a CNN, and an LSTM. The outputs from these base learners are integrated into a high-level ANN, serving as the meta-learner, which combines their predictions to produce the final classification. Fig. 3 shows a representation of the SNN-IoMT framework. The following subsections outline the architectures of each base learner and the meta-learner.

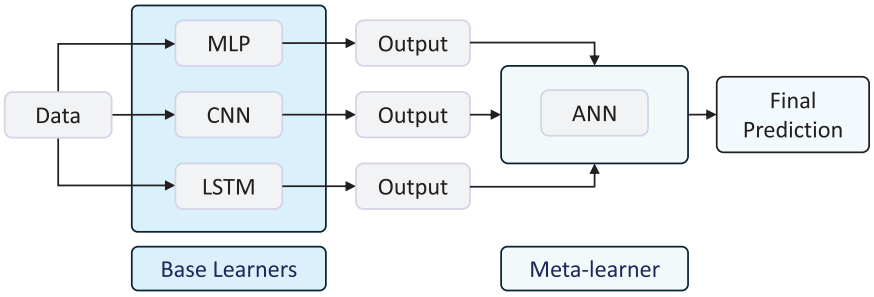


Figure 3: SNN-IoMT model

4.1 Base Learners

The stacking model employs three different base learners, each contributing its strengths to enhance the overall performance. These base learners are selected to capture diverse patterns in the data. An outline of each model is provided in the following subsections.

4.1.1 Multi Layer Perceptron (MLP)

The MLP implemented in this ensemble is a simple feedforward neural network. It is selected for its simplicity and computational efficiency. It is composed of two layers:

- A hidden layer with 8 neurons, each neuron uses the Rectified Linear Unit (ReLU) activation function (1):

$$f(x) = \max(0, x) \tag{1}$$

- A single neuron as an output, employing the sigmoid function for binary classification (2):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

Training of the model is performed using the Adam optimizer (3), a gradient-based optimization algorithm that converges faster than other optimizers and requires less tuning, making it useful in real-world applications.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$
$$\theta_t = \theta_{t-1} - \frac{\alpha m_t}{\sqrt{v_t} + \epsilon} \tag{3}$$

m_t and v_t represent the moving averages of the gradient and squared gradient, respectively, g_t denotes the gradient at time step t, and θ_t represents the updated parameters.

To enhance classification performance, two callbacks are used to monitor training:

- **ReduceLROnPlateau:** When there is no improvement in the validation loss for 5 successive epochs, the learning rate is reduced by 0.2.
- **EarlyStopping:** Training is stopped if the validation loss does not show improvement after 10 consecutive epochs, and the model with the best validation performance is recovered.

The model is optimized using the binary cross-entropy loss function (4):

$$\text{Loss}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1-y_i) \log(1-\hat{y}_i)] \tag{4}$$

where y_i represents the true label and $\hat{y}_i$ corresponds to the predicted value of class i.

4.1.2 Convolutional Neural Networks (CNN)

The CNN used within our stacking model is a one-dimensional network, adapted with binary classification. It offers a reliable technique for managing large datasets. Its architecture comprises:

- A one-dimensional layer featuring two filters, each having a kernel size of 5. The activation function used is ReLU (1).
- A MaxPooling layer with a pool size of 2. It reduces the dimensionality of the feature map, and selects the highest value in each pooling window (5):

$$p_j = \max(x_{i:j}) \tag{5}$$

  where $x_{i:j}$ represents the values within the pooling window.
- A flatten layer to extract a 1D vector from the pooled feature map. The vector is used as input for the dense layer.
- A dense layer containing one neuron and the sigmoid function (2).

The Adam optimizer (3) is applied throughout the training process to adjust the weights during the backpropagation. The model incorporates binary cross-entropy (4) to compute the loss. Similarly to the MLP, ReduceLROnPlateau and EarlyStopping are used during the training.

4.1.3 Long Short-Term Memory (LSTM)

The LSTM architecture used is simple and processes data for binary classification. It is perfectly adapted for analyzing sequential data. Given that network traffic is inherently sequential, this model is chosen. Its architecture consists of the following:

- A layer with 13 LSTM units.
- A dense layer with one neuron, using the sigmoid classification function (2).

The training process is carried out using the Adam optimizer (3), and the binary cross-entropy loss function (4). Similarly to the previous models, two callbacks are used: (1) the learning rate reduction on plateau and (2) early stopping.

4.2 Meta-Learner

The meta-learner in this stacking ensemble is an ANN, which takes as input the prediction of the base learners (MLP, CNN, and LSTM). It aggregates these predictions to produce the final classification. The model is structured as follows:

- The initial dense layer features 16 neurons and incorporates the ReLU function (1).
- The second dense layer is composed of 8 neurons, further reducing the dimensionality and refining the learned features.
- The output layer has a single neuron with a sigmoid activation function (2).

Training the ANN involves the Adam optimizer (3) and binary cross-entropy loss (4).

The hyperparameters for training the models (CNN, LSTM, MLP, ANN) are selected manually based on empirical experimentation, aiming to ensure a balance between accuracy and computational efficiency. While a single large model could achieve high accuracy, it would be useless for IoMT devices. Instead, we combine multiple lightweight architectures with carefully chosen hyperparameter values in a stacking approach, resulting in an optimized resource usage, while achieving impressive intrusion detection performance. Table 3 summarizes the hyperparametres of each model.

Table 3: Summary of hyperparameters for MLP, CNN, LSTM, and meta-learner models in the SNN-IoMT ensemble

Model	Hyperparameter	Value/Function
MLP	Hidden layer neurons	8
	Activation (output layer)	Sigmoid
	Activation (hidden layer)	ReLU
CNN	Convolutional layer filters	2
	Kernel size	5
	Activation (convolutional layer)	ReLU
	Pooling type	MaxPooling
	Pooling size	2
	Activation (output layer)	Sigmoid
LSTM	LSTM units	13
	Activation (LSTM)	Tanh (default)
	Activation (output layer)	Sigmoid
Meta-Learner (ANN)	First dense layer neurons	16
	Activation (first dense layer)	ReLU
	Second dense layer neurons	8
	Activation (second dense layer)	ReLU
	Activation (output layer)	Sigmoid
All Models	Optimizer	Adam

(Continued)

Table 3 (continued)

Model	Hyperparameter	Value/Function
	Loss function	Binary Cross-Entropy
	Training epochs	20
	Batch size	32
Callbacks	ReduceLROnPlateau factor	0.2
	EarlyStopping patience (epochs)	10

5 Datasets, Data Preprocessing and Feature Selection

The experiments were conducted on **Google Colab** using a GPU-enabled environment. The hardware setup includes 12.7 GB of system RAM and 15.0 GB of GPU memory, used with 112.6 GB of disk space. The model training and testing were performed using Python version **3.10.12** with the following libraries:

- **TensorFlow (2.17.0):** Used for building and training DL models.
- **Scikit-learn (1.5.2):** Used for model evaluation, including accuracy calculations and confusion matrices (CM).
- **Pandas (2.2.2):** Employed in the processing and evaluation of data.
- **Matplotlib (3.7.1):** Used for visualizing data and model performance.
- **NumPy (1.26.4):** Utilized for numerical computations.
- **Keras (3.4.1):** Served as an interface for TensorFlow to simplify the definition of neural networks models.

5.1 Datasets

Two datasets available to the public are used for the experiments: **WUSTL-EHMS-2020** and **IoT-Healthcare-Security**. These datasets contain data obtained from IoMT networks. Two datasets were chosen to demonstrate that the SNN-IoMT model is capable of effectively generalizing across several IoMT healthcare environments. In the rest of the paper, we denote WUSTL-EHMS-2020 as W-EHMS, and IoT-Healthcare-Security as IHS.

5.1.1 WUSTL-EHMS-2020 Dataset

This dataset was generated through an Enhanced Healthcare Monitoring System (EHMS) testbed [20]. The EHMS is a controlled environment dedicated to simulate and evaluate real-world healthcare systems. From the patient's sensors, the data flow starts transmitting bio-metric information to the gateway. Then, the server receives the data forwarded by the gateway via a switch and router to facilitate visualization. While being transmitted, attackers can intercept the data before it reaches the server. The dataset includes two types of attacks:

- **Spoofing Attack:** The adversary intercepts the communication between the server and the gateway, violating the privacy of patient data.
- **Data Injection Attack:** The adversary alters packets during transmission, violating data integrity by modifying its content before reaching its destination.

The dataset comprises 44 features, including 35 network flow metrics, 8 patient biometric characteristics, and 1 label feature. Table 4 and Fig. 4 highlight the dataset statistical information.

Table 4: Statistical characteristics of W-EHMS dataset

Label	Value
Normal instances	14,272
Malicious instances	2,046
Total instances	16,318

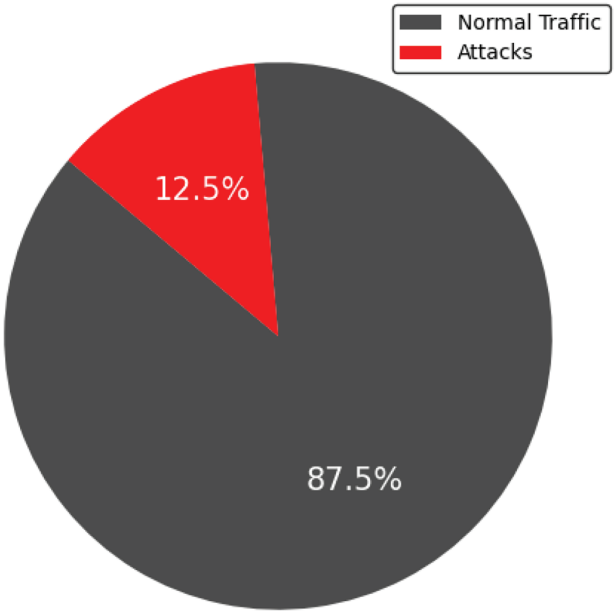


Figure 4: Breakdown of normal vs. attack traffic in W-EHMS dataset

The dataset effectively simulates real-world network traffic, with a significantly higher number of normal samples compared to attack samples. Typically, adversarial traffic in networks represents only a small fraction of the overall traffic.

5.1.2 IoT-Healthcare-Security Dataset

The IoT-Healthcare-Security dataset is collected from a traffic in healthcare systems powered by IoT. It was generated utilizing the IoT-Flock tool, which creates standard and malicious network traffic for IoT devices in various scenarios. A two-bed IoT-based Intensive Care Unit (ICU) setup was established. Every bed is furnished with a collection of nine devices for monitoring a patient and a solitary control unit. This setup provides a realistic environment to study the interactions across multiple medical devices and their associated security risks.

The dataset includes various attack types that can target IoMT infrastructure. The following attacks are represented in the dataset:

- **MQTT Publish Flood:** This threat aims to overwhelm the message broker by inundating it with a high amount of MQTT publish messages.
- **Authentication Bypass Attack:** This attack circumvents established authentication mechanisms to gain unauthorized access to IoMT networks.

- **Denial of Service (DoS) Attack:** Its purpose is to make a service unavailable by flooding the target with excessive requests.
- **Man-in-the-Middle (MitM) Attack:** The cyber-attacker intercepts the traffic between two parties without their knowledge. It involves data manipulation or eavesdropping.
- **Device Impersonation:** In this attack, an attacker pretends to be a legitimate IoT device by copying its identity, like an IP address.
- **Data Manipulation Attack:** It happens when an attacker alters the data sent via the IoT networks, such as patient data (e.g., vital signs), leading to incorrect treatments.

There are three **Comma-Separated Values (CSV)** files that make up the dataset: **Attack.csv**, **patientMonitoring.csv**, and **environmentMonitoring.csv**. These files contain a total of 52 features that represent various network flow metrics, including IP addresses, TCP data, MQTT traffic, and more. Table 5 and Fig. 5 highlight the dataset statistical information.

Table 5: Statistical characteristics of IHS dataset

Label	Value
Normal instances	108,568
Malicious instances	80,126
Total instances	188,694

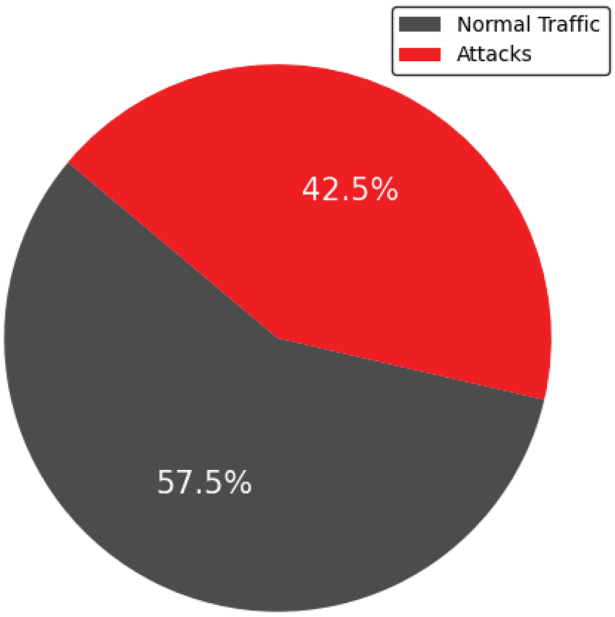


Figure 5: Breakdown of normal vs. attack traffic in IHS dataset

5.2 Data Preprocessing

The preprocessing steps applied to both datasets were designed to ensure that they are clean and suitable for model training.

5.2.1 W-EHMS Dataset

Let $X \in \mathbb{R}^{n \times m}$ represent the dataset, with n indicating the number of samples and m referring to the number of features. The first, second, and last columns were dropped, resulting in a new dataset $X' = X[:, 2 : m-1]$.

Then, we create a binary target variable $T \in \{0, 1\}$, in which 0 denotes normal traffic and 1 denotes adversarial traffic, computed as $T_i = \{0 \text{ if Attack Category}_i = \text{normal}, 1 \text{ otherwise}\}$, which defined our binary classification target $y = T$.

For the origin and target Media Access Control address (MAC) addresses, the colons are removed, and the values are label-encoded to transform them into numerical values.

Label encoding is also applied to other categorical features, including Source Address (SrcAddr) and Destination Address (DstAddr).

Missing values in the numeric columns are imputed with the mean of the respective column j (6):

$$x_{ij} = \frac{1}{n_j} \sum_{i=1}^{n_j} x_{ij} \tag{6}$$

where n_j is the number of non-missing entries in column j.

Feature scaling was then applied using standardization to ensure each feature has a mean of 0 and a standard deviation of 1, computed as:

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{7}$$

where:

$$\mu_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \tag{8}$$

and

$$\sigma_j^2 = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \mu_j)^2 \tag{9}$$

Class imbalance is a challenge in the dataset, as the number of attack traffic samples is significantly smaller compared to normal traffic samples. This imbalance can lead to a poor detection of attack traffic, favoring the majority class resulting in a biased model.

To address this, we apply **Synthetic Minority Over-sampling Technique (SMOTE)**, which creates artificial samples for the underrepresented instances (attack traffic) to balance the dataset:

$$X_{\text{res}}, y_{\text{res}} = \text{SMOTE(random\_state=42).fit\_resample}(X_{\text{before}}, y) \tag{10}$$

where:

- X_{before} is the feature matrix before applying SMOTE.
- y is the target variable.
- X_{res} is the feature matrix after applying SMOTE.
- y_{res} is the target variable after applying SMOTE.

Using SMOTE helps prevent the model from favoring the dominant class, resulting in enhanced generalization and performance.

5.2.2 IHS Dataset

In the initial step, we load 3 CSV files: *Attack.csv*, *environmentMonitoring.csv*, and *patientMonitoring.csv*.

The three datasets, $X_i \in \mathbb{R}^{n_i \times m_i}$, where n_i denotes the number of samples and m_i represents the features, are concatenated into a single dataset:

$$X = \text{concat}(X_1,\ X_2,\ X_3)$$

The missing values in the numeric columns are filled with the mean of each respective column. Finally, feature scaling was then applied using standardization.

This dataset is already reasonably balanced. The difference between the percentage of the two classes is 15%. Thus, we don't need to apply a balancing technique.

5.3 Feature Selection

Since feature selection greatly influences accuracy, it is a key process before training ML and DL models [32]. It aims to enhance model performance by selecting the most relevant features. Additionally, this process improves computational efficiency. There are three feature selection methods:

1. **Filter Methods:** They evaluate the importance of features using statistical techniques, such as Chi-square tests and correlation coefficients, without relying on any ML model.
2. **Wrapper Methods:** They select feature subsets by using a ML model and evaluate its performance. The most commonly used methods are forward selection and backward elimination.
3. **Embedded Methods:** They conduct feature selection during the model's training process, employing a particular model such as DT.

In our work, we employ an embedded feature selection method that uses **LightGBM** to extract the features that contribute significantly in the classification task. Initially, a LightGBM classifier is trained on the dataset. The model constructs DTs and computes feature importance scores, which represent how much each feature contributes to reducing the loss at each node. The importance of a feature is determined by aggregating its contributions across all DTs in the ensemble. The importance scores I_j for each feature j are calculated from the model during the training phase, based on how each importance score I_j for the feature j is used in tree splits. Then, we select features whose importance scores exceed the median importance score. Formally, the selected feature set X_{selected} consists of features j where:

$$I_j > \text{median}(I) \tag{11}$$

This thresholding approach ensures that we focus our training on features that have a significant influence on the classification tasks.

Figs. 6 and 7 represent the importance scores of the top selected features from the two datasets. The highest importance scores highlight the most influential features in model performance, demonstrating their significant contribution to the prediction outcomes.

Each dataset is then split into three subsets: 70% is allocated for training, 10% for validation, while the remaining 20% is reserved for testing.

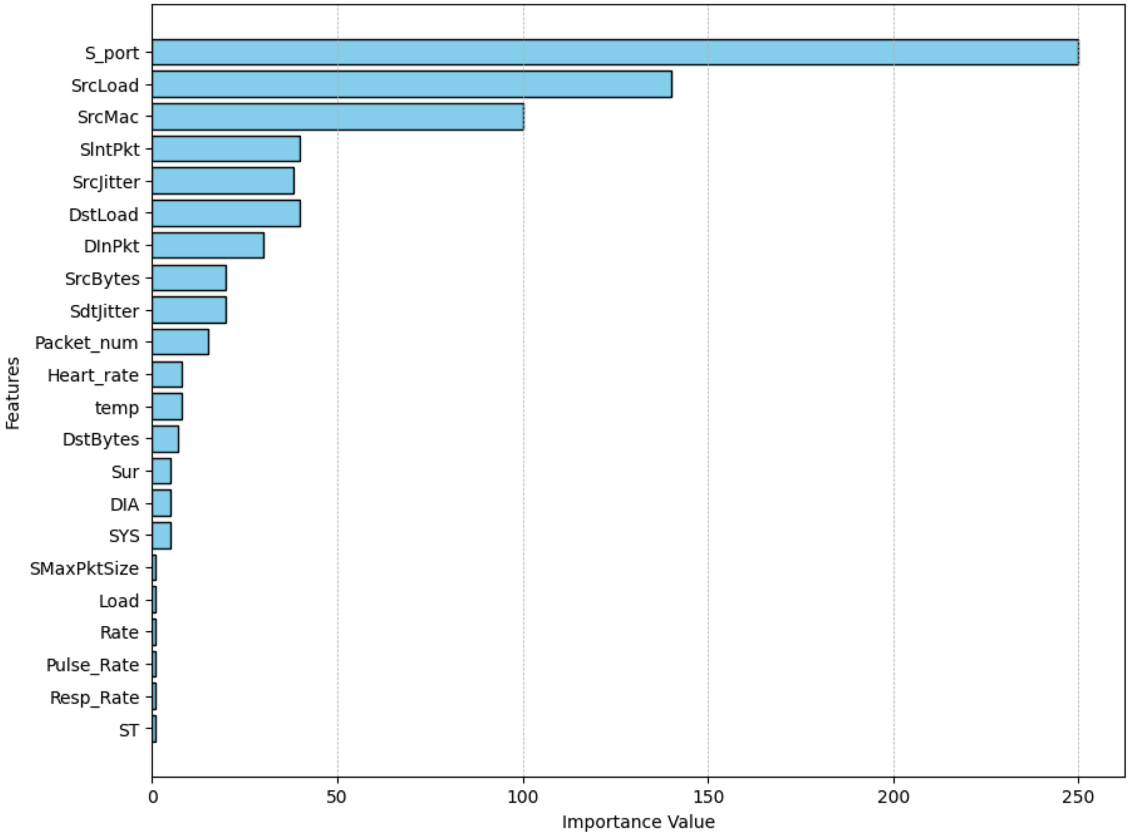


Figure 6: Feature importance scores for W-EHMS dataset

6 Evaluation Metrics and Experimental Validation

In this section, we provide an overview of the evaluation metrics and explain the experimental validation methodology adopted to ensure the model's effectiveness.

6.1 Performance Metrics

To assess the robustness of the model, numerous performance metrics are used. They are derived from the Confusion Matrix (CM) [17]. CM summarizes results on a classification scenario providing True Positive (TP), True Negative (TN), False Negative (FN), and False Positive (FP).

6.1.1 Accuracy

Accuracy represents the proportion of instances that are correctly classified (both positive and negative) out of all instances, reflecting the model's overall performance. The formula used is:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{12}$$

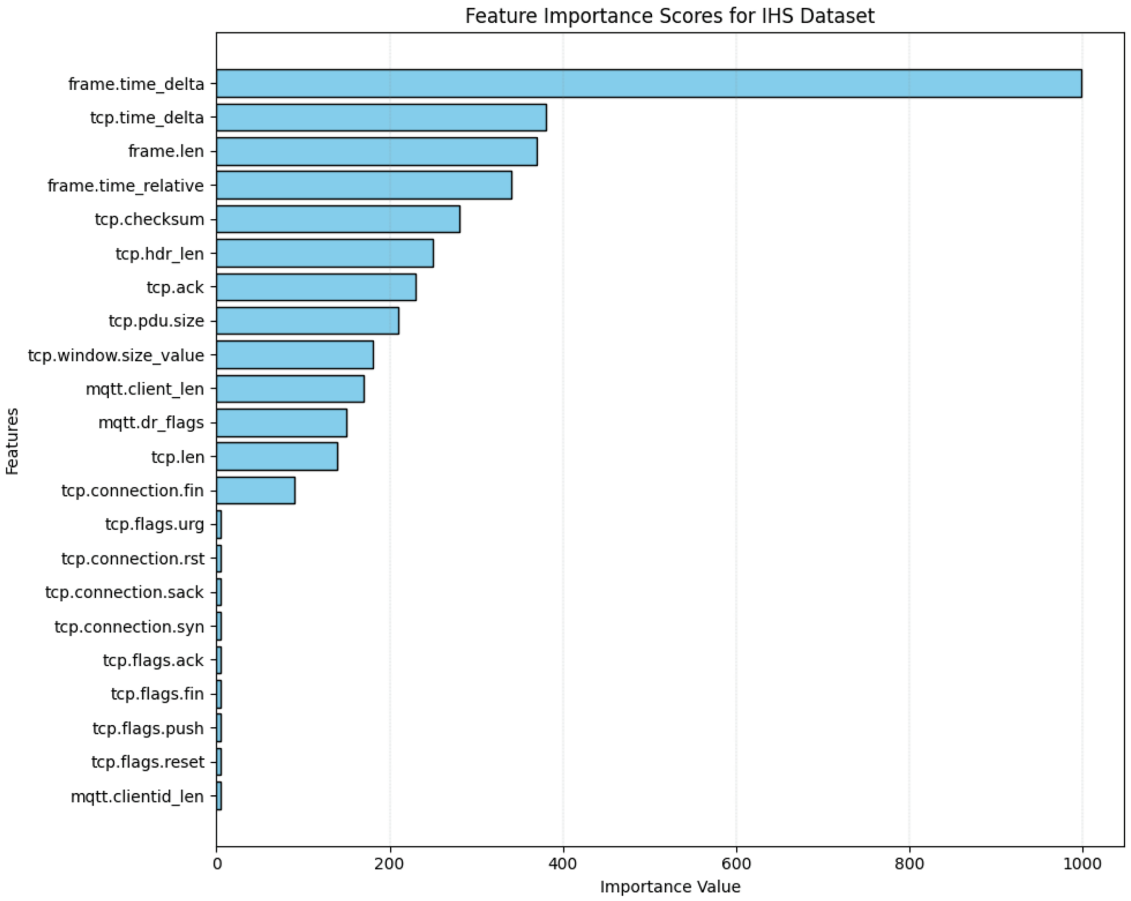


Figure 7: Feature importance scores for IHS dataset

6.1.2 Recall

Recall assesses the proportion of true positives to the whole actual positive cases. It is very important when missing positive cases is costly in security applications, since a high rate of undetected attacks can lead to severe consequences. It is calculated using the equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

6.1.3 Precision

This metric examines the ratio of true positives to the overall number of samples classified as positive. This metric is particularly important because false alarms in real-world intrusion detection (false positives) can lead to unnecessary interventions. Its formula is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

6.1.4 F1-Score

It combines precision and recall into a single metric by calculating their harmonic mean, balancing the trade-off between them. The formula used is:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

6.1.5 Area Under the Curve (AUC)

It denotes the area under the Receiver Operating Characteristics (ROC) curve, where the True Positive Rate (TPR) is plotted against the False Positive Rate (FPR) using several classification thresholds. The TRP and FRP are calculated as:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \text{Recall} \tag{16}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{17}$$

The AUC assesses the model's proficiency in distinguishing between classes, with values falling between 0 and 1. A higher AUC value signifies that the model ranks positive instances better than negative ones. It is determined by:

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR} \tag{18}$$

6.1.6 Logistic Loss (Log Loss)

Log Loss evaluates the performance by measuring the probability output, that ranges between 0 and 1. It assigns a higher penalty to incorrect predictions made with high confidence. A lower Log Loss means that the model assigns meaningful probability scores, which is important for risk assessment. It is calculated as:

$$\text{Log Loss} = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{19}$$

where $\hat{y}_i$ is the predicted probability for the i-th instance, y_i is the actual label, and n is the total number of instances.

6.2 Model Evaluation

One of the challenges we face during the stacking process is the limited data available for training the meta-learner. In stacking, we use the predictions generated by the validation process of the base learners to train the meta-learner. Usually, the validation data represents a small portion of the whole dataset, which leads to a weak generalization and a low AUC, sometimes below 0.5, indicating that the meta-model predicts randomly. This is a common issue in stacking architectures where the meta-learner suffers from insufficient and potentially noisy data, leading to poor performance.

Another major challenge we encounter in developing a stacked architecture is overfitting. This problem happens when a model captures the noise. As a result, the model performs well on the training data but struggles to generalize to unseen test data [33].

To address these two major challenges in the implementation of our model (SNN-IoMT), we employ the **k-fold cross-validation** approach. This method is based on dividing the training dataset into k approximately equal subsets or folds, in our case $k = 5$, denoted as $D_1, D_2, \ldots, D_5$. In each iteration, a different fold D_i is used for validation, and the remaining subsets $D - D_i$ serve as the training subsets. This process guarantees that every data point is used both for training and validation at least once. Algorithm 1 and Fig. 8 briefly explain this mechanism.

Algorithm 1: 5-fold cross-validation process for model training and meta-learner integration

Input: Dataset D of size n
Parameters: Number of folds $k = 5$
Divide dataset D into k folds: D_1, D_2, D_3, D_4, D_5
Generate combinations of training and validation sets:
for each fold i from 1 to k **do**
    Set validation set $V \leftarrow D_i$
    Set training set $T \leftarrow D \backslash D_i$ (i.e., all folds except D_i)
    Store combination *(T, V)*
end for
for each combination *(T, V)* **do**
    for each model in {CNN, LSTM, MLP} **do**
        Train model on T
        Validate model on V
        Compute performance metrics, and store as P
        Store predictions on V as Predictions
    end for
    Aggregate predictions from all models for the current combination
end for
Train meta-learner (ANN) on aggregated predictions from all combinations
Output: Trained meta-learner model

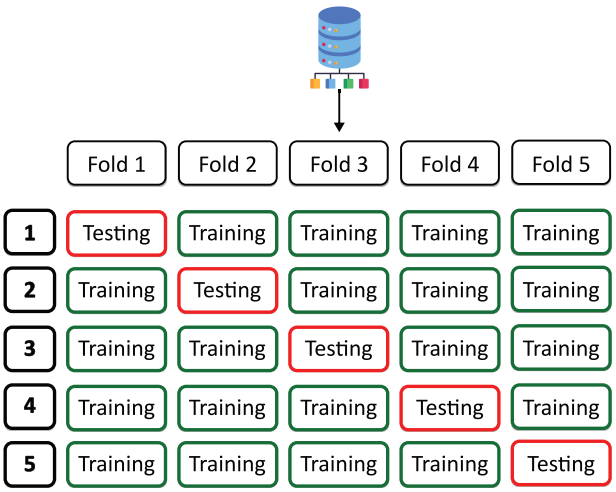


Figure 8: 5-fold cross-validation

This method has several benefits:

- **Improve Meta-Learner Performance:** The meta-model benefits from k-fold cross-validation by being trained on more diverse predictions from the base-learners, increasing the variability of training data for the meta-learner. This mechanism allows to reduce the risk of poor generalization caused by limited data.
- **Reduce Overfitting:** By using different subsets of the data for model training, this technique results in a more robust evaluation. Since the model is repeatedly trained and validated multiple times on different sub-datasets, it is forced to learn patterns that generalize well to new data, rather than memorizing specific patterns.
- **Bias-Variance Trade-off:** Bias refers to the expected difference between the estimated and true accuracy, while variance represents the variability of the model's accuracy across different data splits. K-fold cross-validation provides insight into bias and variance, which are important for understanding its predictive behavior [34].

7 Results and Discussions

This section highlights the findings of the experiments on the two datasets. The classification task involves a binary classification, where the model SNN-IoMT is trained and assessed based on the following metrics: accuracy, recall, precision, F1-Score, AUC, and Log Loss. These metrics highlight the model's effectiveness and performance in classifying the traffic, and generalizing across different folds of the dataset.

Binary classification focuses on identifying adversarial traffic within the network, distinguishing between normal traffic and potential attacks.

Additionally, our results are compared against established state-of-the-art methods, showcasing the improvements of each model.

7.1 Performance Evaluation

7.1.1 W-EHMS Dataset

Table 6 illustrates the performance metrics of each base learner model (CNN, LSTM, MLP) and the SNN-IoMT classifier across this dataset.

Table 6: Performance of models on W-EHMS dataset (Average over 5 folds)

Model	Accuracy	Recall	F1-Score	Precision	AUC	Log loss
CNN	97.18%	99.29%	97.29%	95.48%	99.62%	22.89%
LSTM	97.13%	98.28%	97.22%	96.33%	99.30%	9.78%
MLP	99.07%	99.92%	99.08%	98.27%	99.84%	15.93%
SNN-IoMT	**99.65%**	**99.84%**	**99.65%**	**99.46%**	**99.93%**	**1.17%**

The table demonstrates that all models perform well in detecting anomalies in IoMT, showing high accuracies. However, SNN-IoMT shows a noticeable improvement across several metrics compared to the base learners. For accuracy, SNN-IoMT achieves 99.65%, outperforming all base models, with CNN, LSTM, and MLP achieving 97.18%, 97.13% and 99.07%, respectively. SNN-IoMT also achieves a very high recall (99.84%), ensuring that few malicious patterns are missed. This very high value of recall is crucial for security-focused applications. The F1-Score of SNN-IoMT achieves 99.65%, reflecting an excellent balance between precision and recall, and demonstrating a high performance across the two metrics. Although the CNN, LSTM and MLP models exhibit high F1-Scores (97.29%, 97.22% and 99.08%, respectively), the stacking model

reflects its ability to fine-tune the combined outputs of the base learners. For precision, MLP is the best model among the base learners with 98.27%, followed by LSTM with 96.33%, and CNN with 95.48%. The SNN-IoMT exceeds the base learners achieving 99.46%, indicating its possibility to decrease the false positive, which is very important in environments with low tolerance for false alarms, like IoMT security. The AUC for SNN-IoMT is very higher, achieving 99.93%, compared to the base learners, showing that it has a better ability to differentiate between standard and adversarial traffics. Finally, SNN-IoMT has the lowest Log Loss (1.77%) among all models, reflecting its superior confidence in predictions.

7.1.2 IHS Dataset

Table 7 illustrates the performance metrics of each base learner model (CNN, LSTM, MLP) and the SNN-IoMT classifier across this dataset.

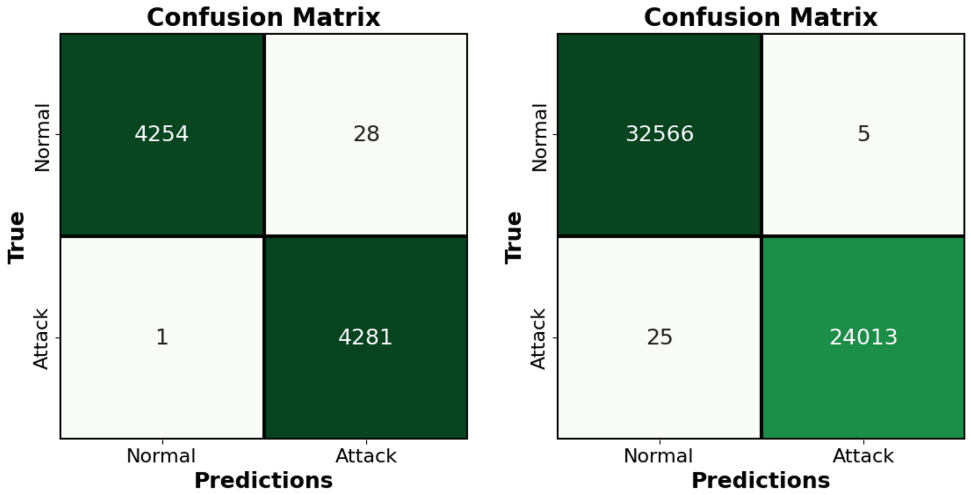
Table 7: Performances of models on IHS dataset (Average over 5 folds)

Model	Accuracy	Recall	F1-Score	Precision	AUC	Log loss
CNN	99.93%	99.93%	99.92%	99.91%	100.00%	0.13
LSTM	99.74%	99.46%	99.68%	99.91%	99.94%	1.57%
MLP	99.92%	99.95%	99.90%	99.86%	100.00%	0.13%
SNN-IoMT	**99.95%**	**99.90%**	**99.94%**	**99.98%**	**100.00%**	**0.19%**

The table shows that all models are very effective in identifying anomalies within the IoMT environment, demonstrating high accuracies. Similar to the previous dataset results, the stacking model, SNN-IoMT, exhibits notable performance improvements across several metrics compared to the base learners. Specifically, SNN-IoMT achieves an accuracy of 99.95% surpassing all base models, with CNN, LSTM, and MLP attaining accuracies of 99.93%, 99.74%, and 99.92%, respectively. In terms of recall, SNN-IoMT achieves 99.90%, indicating a minimal chance of missing adversarial traffic patterns. The F1-Score of SNN-IoMT reaches 99.94%, reflecting a well-balanced relationship between precision and recall. While CNN, LSTM, and MLP also exhibit high F1-Scores (99.92%, 99.68%, and 99.90%, respectively), SNN-IoMT further refines the predictions by leveraging the strengths of each base learner. For precision, the stacking model achieves 99.98%, showing its effectiveness in minimizing false positives, which is important for IoMT security systems. The AUC also achieves an impressive value of 100%, surpassing the base models and demonstrating its capabilities in distinguishing between normal and abnormal traffic. Lastly, the Log Loss of SNN-IoMT is 0.19%, the lowest value among all models, reflecting a high degree of confidence in its predictions.

7.1.3 Overall Discussion

Fig. 9a,b shows the confusion matrices for the two datasets: W-EHMS and IHS, respectively. As shown in the confusion matrices, the model identifies true positives, while the number of false positives and false negatives remains extremely low, contributing to the high performance of our model.



(a) CM for W-EHMS Dataset **(b)** CM for IHS Dataset

Figure 9: Comparison of confusion matrices for W-EHMS and IHS datasets

7.2 Training and Validation Curves Analysis

Fig. 10 represents the accuracy and loss curves of training and validation of the base learners in the first two folds of the W-EHMS dataset. Furthermore, Fig. 11 illustrates the curves of the IHS dataset.

The plots clearly show strong performance, with both training and validation accuracy increasing together over time, indicating that the models generalize well to unseen data and learn efficiently. Additionally, for each model, the training values closely match the validation values, reflecting that the models are not overfitting and the validation performance is consistently strong. These results highlight the robustness and stability of the model's effectiveness on various data splits, confirming the effectiveness of the training process and model generalization.

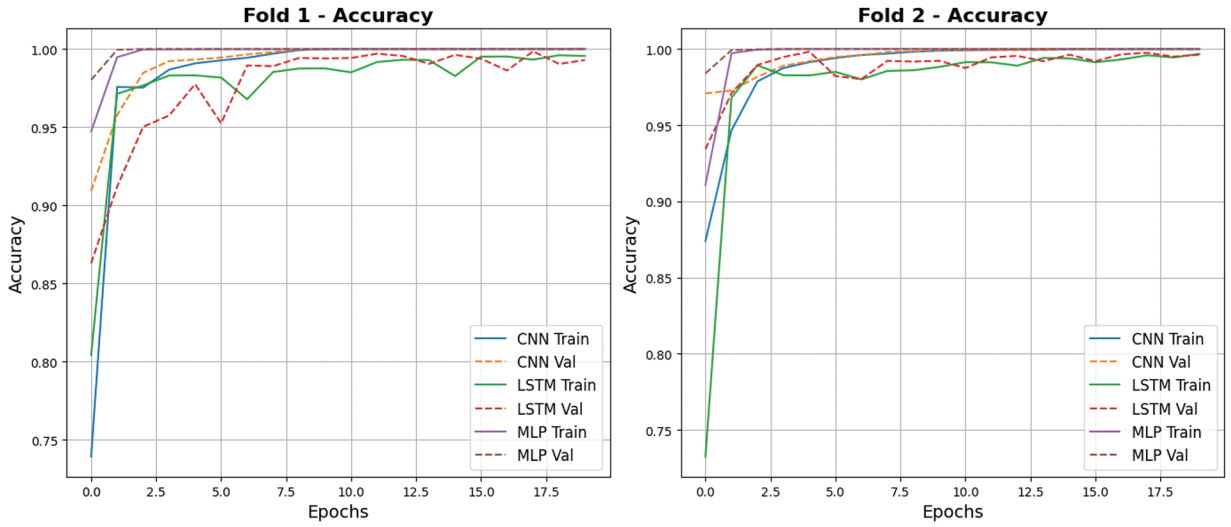


Figure 10: (Continued)

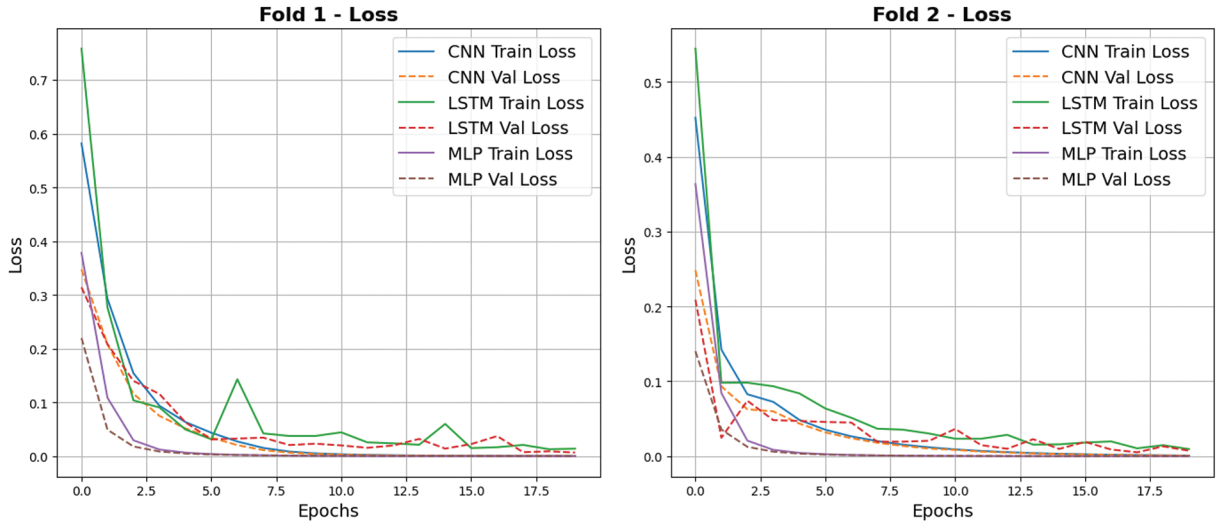


Figure 10: Training and validation curves for W-EHMS dataset

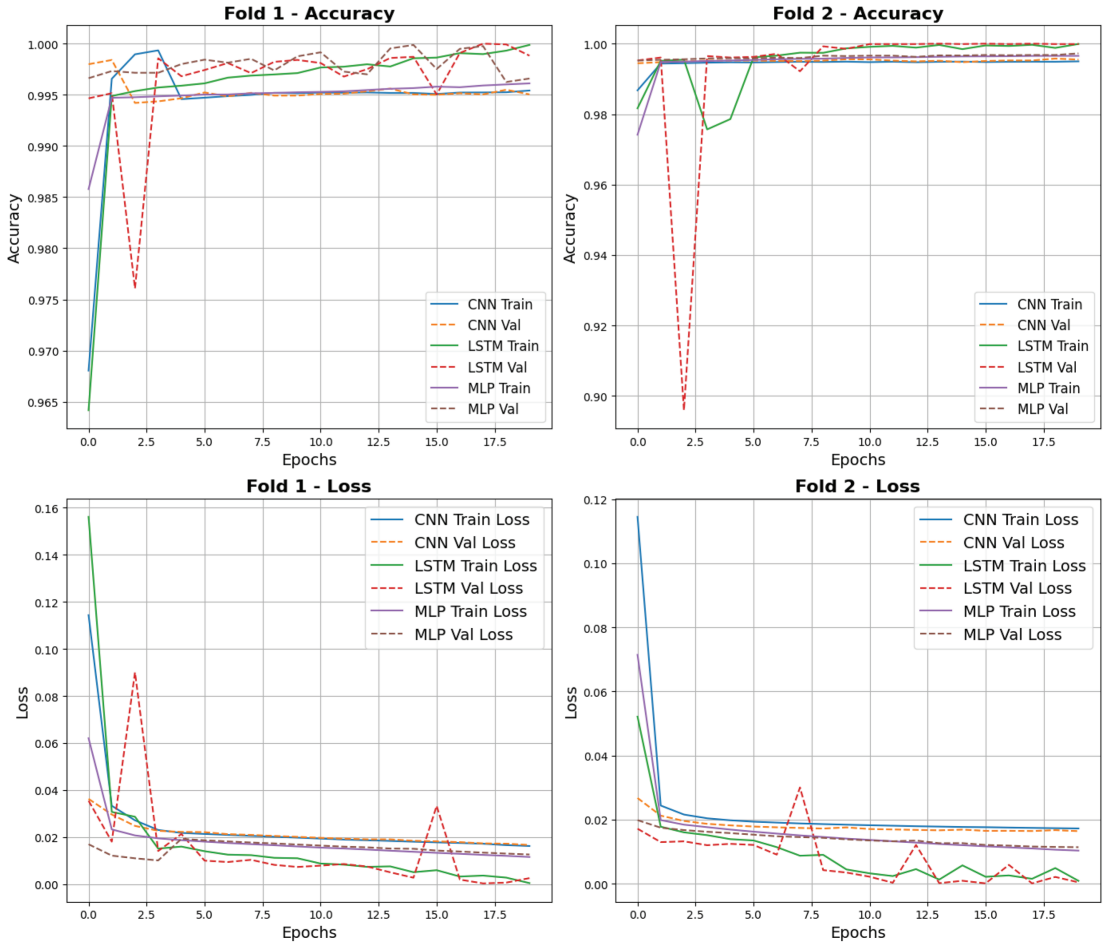


Figure 11: Training and validation curves for IHS dataset

7.3 Feature Selection Impact

To assess the impact of the feature selection process (FS) impact on model performance, we conduct experiments with and without FS on both datasets. The training times in two cases are summarized in Table 8.

Table 8: Training time comparison with and without feature selection

Feature selection	Dataset	Training time
With FS	W-EHMS	919.81 s
	IHS	4664.25 s
Without FS	W-EHMS	1319.247 s
	IHS	8809.90 s

The results demonstrate that applying FS enhances efficiency by reducing training time. Removing irrelevant or redundant attributes allows the model to focus on the most informative features, leading to faster convergence and reduced computational cost.

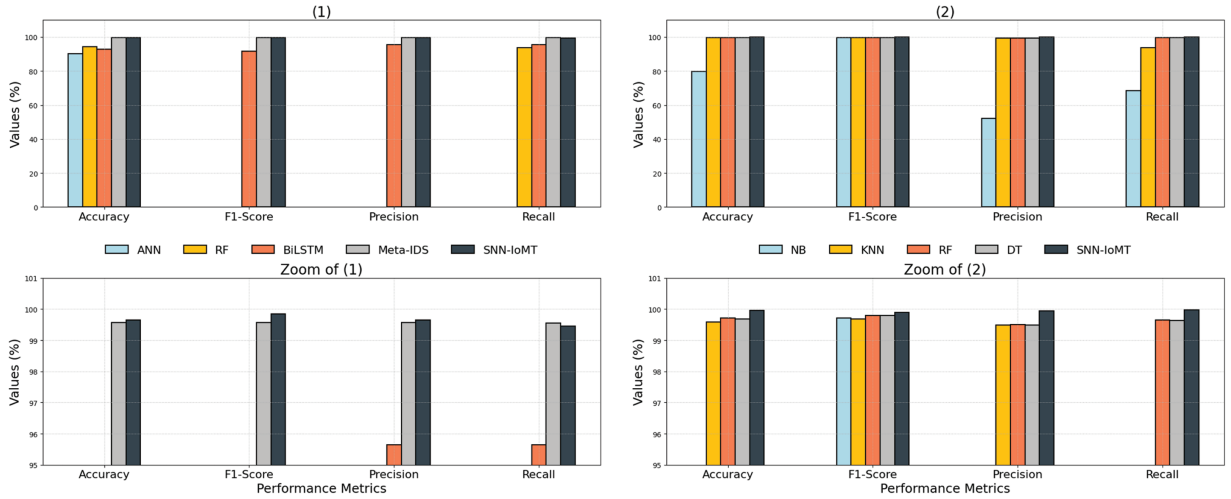
7.4 Comparison with Previous Works

This section focuses on comparing the results obtained by SNN-IoMT with those of state-of-the-art approaches. The studies being compared utilized the same datasets as in our research. Table 9 and Fig. 12a display a comparison of the results for the W-EHMS dataset with previous work, while Table 10 and Fig. 12b show the results for the IHS dataset.

For W-EHMS dataset, Meta-IDS [28] emerged as the top classifier, achieving an accuracy of 99.57%. However, our SNN-IoMT model outperforms all existing works, achieving an accuracy of 99.65%, a precision of 99.84%, and a recall of 99.65%. For the second dataset, our approach achieves an accuracy of 99.95%, surpassing all previous methods. Previous methods such as RF, KNN, and DT [15] reached high accuracy levels, ranging from 99.59% to 99.71%. However, the SNN-IoMT demonstrates superior performance across all metrics. These improvements highlight the robustness of our architecture in anomaly detection tasks across these datasets.

Table 9: Comparison of classifier performance on W-EHMS dataset

Work	Classifier	Accuracy	Precision	Recall	F1-Score
[20]	ANN	90.04%	–	–	–
[26]	RF	94.23%	–	–	93.8%
[27]	BiLSTM	92.95%	91.61%	95.64%	95.64%
[28]	Meta-IDS	99.57%	99.57%	99.57%	99.56%
Our work	**SNN-IoMT**	**99.65%**	**99.84%**	**99.65%**	**99.46%**



(a) Classifier Performance on W-EHMS Dataset **(b)** Classifier Performance on IHS Dataset

Figure 12: Classifier performance on W-EHMS and IHS datasets

Table 10: Comparison of classifier performance on IHS dataset

Work	Classifier	Accuracy	Precision	Recall	F1-Score
[15]	NB	79.67%	99.71%	52.18%	68.51%
[15]	KNN	99.59%	99.69%	99.49%	93.8%
[15]	RF	99.71%	99.80%	99.51%	99.65%
[15]	DT	99.69%	99.80%	99.48%	99.64%
Our work	**SNN-IoMT**	**99.95%**	**99.90%**	**99.94%**	**99.98%**

8 Challenges and Deployment Considerations

While our model (SNN-IoMT) demonstrates high accuracy, various challenges in real-world deployment in IoMT systems have to be addressed:

- IoMT devices often have limited computational power, making real-time intrusion detection challenging. Our stacking DL approach is based on lightweight architectures and an optimized feature extraction process, to reduce the complexity. However, other improvements, such as model pruning, can be used to reduce computational overhead and maintain high performance.
- Since IoMT devices are resource-constrained, deploying DL models is very challenging. Our stacking model, while optimized, may require other optimizations like reducing redundant parameters, to minimize memory usage and improve deployment feasibility.

Addressing these challenges ensures a balance between high accuracy and practical usability, making it well-suited for real-time intrusion detection on resource-limited IoMT devices.

9 Conclusion

Detection of adversarial attacks and malicious activities in IoMT infrastructures has become increasingly vital. As the number of cyberattacks increases, developing tools capable of identifying anomalies efficiently and accurately is essential. In this work, we presented a novel framework, **SNN-IoMT**, which is a

stack of CNN, LSTM, and MLP models. The main objective of this ensemble model is to minimize the rates of FPs and FNs, thereby enhancing the reliability and trustworthiness of IoMT security systems. Moreover, our SNN-IoMT model not only enhances the intrusion detection capabilities but also protects patient data confidentiality and integrity, which is essential for preserving trust in AI-driven healthcare solutions.

Using two publicly available datasets, WUSTL-EHMS-2020 and IoT-Healthcare-Security, our results showed that our classifier outperforms the current state-of-the-art approaches in terms of accuracy, precision, recall, and F1-Score. While the simple classifiers provide comparatively lower performance, our advanced DL framework has been shown to provide very high effectiveness for IoMT threat detection.

In future work, we recommend optimizing our model by tuning hyperparameters using other techniques, including grid search, Bayesian optimization, and reinforcement learning, to enhance model performance and generalization. Furthermore, we recommend integrating federated learning to improve privacy preservation, enabling secure and decentralized anomaly detection. Moreover, extending the evaluation to larger and more diverse datasets would better demonstrate the model's applicability in production environments. Finally, exploring advanced AI techniques, such as transfer learning and self-supervised learning, may enhance scalability and robustness against evolving cyber attacks.

Acknowledgement: The authors gratefully acknowledge Al Yamamah University for supporting this research.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Study Conception and Design: Mourad Benmalek, Abdessamed Seddiki; Data Collection, Experiment and Analysis: Mourad Benmalek, Abdessamed Seddiki; Supervision, Discussion: Mourad Benmalek, Abdessamed Seddiki, Kamel-Dine Haouam; Writing, Editing: Mourad Benmalek, Abdessamed Seddiki, Kamel-Dine Haouam. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in the current study are publicly available at: https://www.cse.wustl.edu/~jain/ehms/index.html (accessed on 06 March 2025) and https://www.kaggle.com/datasets/faisalmalik/iot-healthcare-security-dataset (accessed on 06 March 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Li S, Xu LD, Zhao S. The Internet of things: a survey. Inf Syst Front. 2015;17:243–59. doi:10.1016/j.comnet.2010.05.010.
2. Allifah NM, Zualkernan IA. Ranking security of IoT-based smart home consumer devices. IEEE Access. 2022;10:18352–69. doi:10.1109/ACCESS.2022.3148140.
3. Gaikwad PP, Gabhane JP, Golait SS. A survey based on smart homes system using Internet-of-Things. In: 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC); 2015 Mar 26–27; Melmaruvathur, India: IEEE; 2015. p. 330–5.
4. Lee I. The Internet of Things for enterprises: an ecosystem, architecture, and IoT service business model. Internet Things. 2019;7:100078. doi:10.1016/j.iot.2019.100078.
5. Miazi MNS, Erasmus Z, Razzaque MA, Zennaro M, Bagula A. Enabling the Internet of Things in developing countries: opportunities and challenges. In: 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV); 2016 May 13–14; Dhaka, Bangladesh: IEEE; 2016. p. 564–9.

6. Keertikumar M, Shubham M, Banakar RM. Evolution of IoT in smart vehicles: an overview. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT); 2015 Oct 8–10; Greater Noida, India: IEEE; 2015. p. 804–9.
7. John Dian F, Vahidnia R, Rahmati A. Wearables and the Internet of Things (IoT), applications, opportunities, and challenges: a survey. IEEE Access. 2020;8:69200–11. doi:10.1109/ACCESS.2020.2986329.
8. Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M. Internet of Things for smart cities. IEEE Internet Things J. 2014;1(1):22–32. doi:10.1109/JIOT.2014.2306328.
9. Rejeb A, Rejeb K, Treiblmaier H, Appolloni A, Alghamdi S, Alhasawi Y, et al. The Internet of Things (IoT) in healthcare: taking stock and moving forward. Internet Things. 2023;22:100721. doi:10.1016/j.ipha.2024.01.003.
10. Sadhu PK, Yanambaka VP, Abdelgawad A. Internet of Things: security and solutions survey. Sensors. 2022;22(19):7433. doi:10.3390/s22197433.
11. Yaacoub JPA, Noura M, Noura HN, Salman O, Yaacoub E, Couturier R, et al. Securing Internet of Medical Things systems: limitations, issues, and recommendations. Future Gener Comput Syst. 2020;105:581–606. doi:10.1016/j.future.2019.12.028.
12. Hernandez-Jaimes ML, Martinez-Cruz A, Ramírez-Gutiérrez KA, Feregrino-Uribe C. Artificial intelligence for IoMT security: a review of intrusion detection systems, attacks, datasets, and cloud-fog–edge architectures. Internet Things. 2023;23:100887. doi:10.1016/j.iot.2023.100887.
13. Rani S, Kataria A, Kumar S, Tiwari P. Federated learning for secure IoMT applications in smart healthcare systems: a comprehensive review. Knowl-Based Syst. 2023;274:110658. doi:10.1016/j.knosys.2023.110658.
14. Aman AHM, Hassan WH, Sameen S, Attarbashi ZS, Alizadeh M, Latiff LA. IoMT amid COVID-19 pandemic: application, architecture, technology, and security. J Netw Comput Appl. 2021;174:102886. doi:10.1016/j.jnca.2020.102886.
15. Hussain F, Abbas SG, Shah GA, Pires IM, Fayyaz UU, Shahzad F, et al. A framework for malicious traffic detection in IoT healthcare environment. Sensors. 2021;21(9):3025. doi:10.3390/s21093025.
16. Lallie HS, Shepherd LA, Nurse JRC, Erola A, Epiphaniou G, Maple C, et al. Cyber security in the age of COVID-19: a timeline and analysis of cyber-crime and cyber-attacks during the pandemic. Comput Secur. 2021;105:102248. doi:10.1016/j.cose.2021.102248.
17. Lazzarini R, Tianfield H, Charissis V. A stacking ensemble of deep learning models for IoT intrusion detection. Knowl-Based Syst. 2023;279:110941. doi:10.1016/j.knosys.2023.110941.
18. Zhou ZH. Ensemble methods: foundations and algorithms. Boca Raton, FL, USA: CRC Press; 2012. p. 1–218.
19. Yaman MA, Rattay F, Subasi A. Comparison of bagging and boosting ensemble machine learning methods for face recognition. Procedia Comput Sci. 2021;194:202–9. doi:10.1016/j.procs.2021.10.074.
20. Hady AA, Ghubaish A, Salman T, Unal D, Jain R. Intrusion detection system for healthcare systems using medical and network data: a comparison study. IEEE Access. 2020;8:106576–84. doi:10.1109/ACCESS.2020.3000421.
21. Saheed YK, Arowolo MO. Efficient cyber attack detection on the Internet of Medical Things-smart environment based on deep recurrent neural network and machine learning algorithms. IEEE Access. 2021;9:161546–54. doi:10.1109/ACCESS.2021.3128837.
22. Subasi A, Algebsani S, Alghamdi W, Kremic E, Almaasrani J, Abdulaziz N. Intrusion detection in smart healthcare using bagging ensemble classifier. In: Proceedings of the International Conference on Medical and Biological Engineering; 2020 Apr 21–24; Mostar, Bosnia and Herzegovina.
23. Nandy S, Adhikari M, Khan MA, Menon VG, Verma S. An intrusion detection mechanism for secured IoMT framework based on swarm-neural network. IEEE J Biomed Health Inf. 2021;26(5):1969–76. doi:10.1109/JBHI.2021.3101686.
24. Binbusayyis A, Alaskar H, Vaiyapuri T, Dinesh M. An investigation and comparison of machine learning approaches for intrusion detection in IoMT network. J Supercomput. 2022;78(15):17403–22. doi:10.1007/s11227-022-04568-3.
25. Zubair M, Ghubaish A, Unal D, Al-Ali A, Reimann T, Alinier G, et al. Secure Bluetooth communication in smart healthcare systems: a novel community dataset and intrusion detection system. Sensors. 2022;22(21):8280. doi:10.3390/s22218280.

26. Gupta K, Sharma DK, Gupta KD, Kumar A. A tree classifier based network intrusion detection model for Internet of Medical Things. Comput Electr Eng. 2022;102:108158. doi:10.1016/j.compeleceng.2022.108158.
27. Wagan SA, Koo J, Siddiqui IF, Qureshi NMF, Attique M, Shin DR. A fuzzy-based duo-secure multi-modal framework for IoMT anomaly detection. J King Saud Univ Comput Inf Sci. 2022;35(1):131–44. doi:10.1016/j.jksuci.2022.11.007.
28. Zukaib U, Cui X, Zheng C, Hassan M, Shen Z. Meta-IDS: meta-learning-based smart intrusion detection system for Internet of Medical Things (IoMT) network. IEEE Internet Things J. 2023;11(13):23080–95. doi:10.1109/JIOT.2024.3387294.
29. Kulshrestha P, Kumar TVV. Machine learning based intrusion detection system for IoMT. Int J Syst Assur Eng Manage. 2024;15:1802–14. doi:10.1016/j.asoc.2023.110227.
30. Mathkor DM, Mathkor N, Bassfar Z, Bantun F, Slama P, Ahmad F, et al. Multirole of the internet of medical things (IoMT) in biomedical systems for managing smart healthcare systems: an overview of current and future innovative trends. J Infect Public Health. 2024;17(4):559–72. doi:10.1016/j.jiph.2024.01.013.
31. Benmalek M, Haouam KD. Advancing network intrusion detection systems with machine learning techniques. Adv Artif Intell Mach Learn. 2024;4(3):2575–92. doi:10.54364/AAIML.2024.43150.
32. Alhowaide A, Alsmadi I, Tang J. PCA, random-forest and Pearson correlation for dimensionality reduction in IoT IDS. In: Proceedings of IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS); 2020 Sep 9–11; Vancouver, BC, Canada: IEEE; 2020. p. 1–6.
33. Bejani MM, Ghatee M. A systematic review on overfitting control in shallow and deep neural networks. Artif Intell Rev. 2021;54:6391–438. doi:10.1007/s10462-021-09975-1.
34. Wong TT. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recognit. 2015;48(9):2839–46. doi:10.1016/j.patcog.2015.03.009.