


## Article

# Privacy-Preserving Federated Learning-Based Intrusion Detection System for IoHT Devices

Fatemeh Mosaiyebzadeh <sup>1,\*</sup>, Seyedamin Pouriyeh <sup>2</sup>, Meng Han <sup>3</sup>, Liyuan Liu <sup>4</sup>, Yixin Xie <sup>2</sup>,  
Liang Zhao <sup>2</sup> and Daniel Macêdo Batista <sup>1</sup>

<sup>1</sup> Department of Computer Science, University of São Paulo, São Paulo 05508-090, SP, Brazil; batista@ime.usp.br

<sup>2</sup> Department of Information and Technology, Kennesaw State University, Marietta, GA 30152, USA; spouriyeh@kennesaw.edu (S.P.); yxie11@kennesaw.edu (Y.X.); lzhaol0@kennesaw.edu (L.Z.)

<sup>3</sup> College of Computer Science and Technology, Zhejiang University, Hangzhou 310000, China; mhan@zju.edu.cn

<sup>4</sup> Department of Decision & System Sciences, Saint Joseph's University, Philadelphia, PA 19131, USA; lliu@sju.edu

\* Correspondence: fatemehm@ime.usp.br

**Abstract:** In recent years, Internet of Healthcare Things (IoHT) devices have attracted significant attention from computer scientists, healthcare professionals, and patients. These devices enable patients, especially in areas without access to hospitals, to easily record and transmit their health data to medical staff via the Internet. However, the analysis of sensitive health information necessitates a secure environment to safeguard patient privacy. Given the sensitivity of healthcare data, ensuring security and privacy is crucial in this sector. Federated learning (FL) provides a solution by enabling collaborative model training without sharing sensitive health data with third parties. Despite FL addressing some privacy concerns, the privacy of IoHT data remains an area needing further development. In this paper, we propose a privacy-preserving federated learning framework to enhance the privacy of IoHT data. Our approach integrates federated learning with  $\epsilon$ -differential privacy to design an effective and secure intrusion detection system (IDS) for identifying cyberattacks on the network traffic of IoHT devices. In our FL-based framework, SECioHT-FL, we employ deep neural network (DNN) including convolutional neural network (CNN) models. We assess the performance of the SECioHT-FL framework using metrics such as accuracy, precision, recall, F1-score, and privacy budget ( $\epsilon$ ). The results confirm the efficacy and efficiency of the framework. For instance, the proposed CNN model within SECioHT-FL achieved an accuracy of 95.48% and a privacy budget ( $\epsilon$ ) of 0.34 when detecting attacks on one of the datasets used in the experiments. To facilitate the understanding of the models and the reproduction of the experiments, we provide the explainability of the results by using SHAP and share the source code of the framework publicly as free and open-source software.

**Keywords:** IoHT; federated learning; differential privacy; deep learning; intrusion detection



Academic Editor: Antoni Morell

Received: 14 November 2024

Revised: 11 December 2024

Accepted: 20 December 2024

Published: 27 December 2024

**Citation:** Mosaiyebzadeh, F.; Pouriyeh, S.; Han, M.; Liu, L.; Xie, Y.; Zhao, L.; Batista, D.M. Privacy-Preserving Federated Learning-Based Intrusion Detection System for IoHT Devices. *Electronics* **2025**, *14*, 67. <https://doi.org/10.3390/electronics14010067>

**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Currently, the Internet of Healthcare Things (IoHT) is a promising technology for providing remote monitoring of patients. IoHT relies on high-speed Internet connectivity to connect smart sensors that collect health data about patients and reliably transmit them to hospitals and other medical facilities for use [1]. Using these real-time technologies in healthcare, however, could be more reliable for certain critical treatments that need to

be administered before the patient's condition worsens. Since all connections between IoHT devices and medical care units occur over open-access networks, they are highly susceptible to a wide range of potential attacks, such as privacy breaches [2].

To prevent cyberattacks on IoHT devices, intrusion detection systems (IDSs) have been developed as the main part of a security infrastructure. These systems are designed to identify security breaches within computer networks by continuously monitoring and assessing incidents. Machine learning-based IDSs, for instance, employ a model that can detect both attacks and normal behavior. It is important to note, however, that traditional learning-based IDSs usually require the collection and storage of centralized training data, which may pose a number of challenges, such as privacy concerns, the use of large networks, and power consumption issues [3].

To overcome privacy concerns and prevent the sharing of patients' private data, the concept of federated learning (FL) enables collaborative learning capabilities that preserve privacy and reduce training costs by allowing devices to jointly train a distributed model with an aggregation server while retaining all learning data on the device, thus separating machine learning capabilities from centralized storage [4]. However, in some cases, FL alone may not be sufficient to guarantee proper privacy protection [5]. For instance, by backtracking gradients and analyzing updates to the communication models, it is possible to obtain sensitive information [6]. Several types of attacks can be conducted on FL models, such as poisoning attacks [7], inference attacks [8], and backdoor attacks [9].

Several techniques are available to preserve privacy in the FL environment. In conventional FL, differential privacy (DP) is widely used to protect clients' sensitive information as a privacy-preserving technique [10]. Differential privacy is a mathematical framework that ensures the privacy of an individual's data by adding noise to the data and preventing the disclosure of sensitive information [11]. In [12], the authors used local differential privacy on heterogeneous IoT data. The works in [13,14] combine FL and differential privacy for mobility forecasting and smart cyber-physical grid stability assessment.

In this paper, we demonstrate that by using the DP technique, we can address the privacy problem, and FL-based IDSs can be more secure, efficient, and privacy-preserving. Moreover, to detect attacks in the IoHT environment, we propose a basic deep neural network (DNN) including a convolutional neural network (CNN). In our experiments, we use two public IoHT datasets: wustl-ehms-2020 [15] and ECU-IoHT [16]. Our proposed SECioHT-IDS achieves 93.20% accuracy (the higher the better) with a privacy budget ( $\epsilon$ ) of 0.43 (the lower the better) on the wustl-ehms-2020 dataset. On the ECU-IoHT dataset, the framework achieves 95.48% accuracy, with a privacy budget ( $\epsilon$ ) of 0.34. To the best of our knowledge, this is the first paper to apply differential privacy FL to specific IoHT datasets for the development and validation of an IDS. As an additional contribution, and to facilitate the understanding of the neural network models and the reproduction of the experiments, we provide the explainability of the results using SHAP and share the source code of the framework publicly as free and open-source software.

The remainder of this paper is structured as follows: Section 2 reviews prior research on IoT and IoHT security. Section 3 provides background information on FL and DL, as well as a description of our proposed method. In Section 4, we detail the datasets used, outline the data preprocessing steps, and evaluate the performance of our approach. Section 5 discusses the explainability of the results, while Section 6 concludes this paper and suggests directions for future work.

## 2. Related Work

The FL concept preserves data privacy and has significant implications for applications such as anomaly detection in IoT devices [17]. In this approach, sensitive data are trained

on local devices, eliminating the need to transfer personal data to a central server, which helps ensure the security of IoHT devices. However, there are certain attacks in the FL environment that could compromise privacy. In recent years, the use of privacy-preserving federated learning in anomaly detection has been shown to enhance patient data privacy.

In [18], the authors proposed an advanced federated transfer learning-based IDS specifically designed to enhance the security of healthcare-connected devices. The model uses a DNN algorithm to train the network and transfer knowledge from interconnected edge models. To evaluate the model's effectiveness, they utilized the CICIDS2017 (<https://github.com/elifnurkarakoc/CICIDS2017> (accessed on 3 July 2017)) dataset, assessing performance metrics such as accuracy, detection rate, and average training time.

Rashid et al. [19] proposed a federated learning-based IDS to identify and prevent intrusions in Industrial IoT (IIoT) networks using a CNN and recurrent neural network (RNN). This method prioritizes privacy and security by performing federated training with local IoT device data. To evaluate the approach, the authors used a novel dataset called Edge-IIoTset (<https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iiot> (accessed on 27 April 2023)), assessing performance metrics such as accuracy and training time through comprehensive experiments.

In [20], the authors introduce Fed-Inforce-Fusion, a privacy-preserving federated learning-based IDS designed for identifying cyberattacks in IoHT networks. This model utilizes reinforcement learning techniques to uncover complex relationships within medical data. The authors assessed the model using the ToN-IoT (<https://ieee-dataport.org/documents/toniot-datasets> (accessed on 17 May 2022)) dataset, evaluating metrics such as accuracy, loss, and detection rate. The experimental results show that Fed-Inforce-Fusion outperforms established benchmark IDS methods in detecting complex attack vectors.

Friha et al. [21] introduce the differentially private federated learning-based IDS (2DF-IDS), specifically designed to protect smart industrial environments. The proposed model ensures a differentially private gradient exchange within the FL framework. To evaluate its effectiveness, the authors used the same dataset as [19]. The experimental results highlight the exceptional performance of the 2DF-IDS in detecting various cyber threats within Industrial IoT setups, achieving impressive results in metrics such as accuracy, F1-score, recall, and precision.

In [22], the authors propose ImageFed, a federated learning-based IDS that uses a convolutional neural network (CNN) in a privacy-preserving federated environment. They investigate two scenarios that may impact the performance of ImageFed in real-world applications: non-independent and identically distributed (non-IID) clients, and a scarcity of training data. To evaluate the proposed model, the authors used a car-hacking dataset (<https://ocslab.hksecurity.net/Datasets/car-hacking-dataset> (accessed on 29 December 2021)) from the domain of the Internet of Vehicles (IoV), assessing performance metrics such as accuracy and F1-score.

In [23], the authors propose a federated learning architecture, Fed-IIoT, aimed at mitigating adversarial threats like model poisoning within industrial IoT systems. Their approach focuses on detecting Android malware in IIoT environments by leveraging a dual-component strategy. On the participant side, adversarial scenarios are simulated using GAN-based poisoning attacks, while on the server side, an anomaly detection and mitigation mechanism, termed A3GAN, is employed to ensure robust aggregation during model training. Additionally, Fed-IIoT incorporates GAN-based defense techniques to strengthen collaboration while preserving data privacy. The framework's effectiveness was validated on three IoT datasets, showing an 8% improvement in attack detection and

defense accuracy compared to existing methods, alongside strong privacy protection for Android users.

In our previous work [24], we proposed a federated learning-based intrusion detection system to secure Internet of Healthcare Things (IoHT) devices. It employs deep neural networks (DNNs) in a federated setting to detect anomalies in network traffic generated by IoHT devices, addressing privacy concerns by keeping sensitive health data on local devices. In the current paper, we enhance the federated learning approach from [24] by integrating differential privacy (DP) for increased security, which adds an extra layer of protection by introducing noise into model updates. This was not included in our previous work, which focused solely on federated learning without addressing the risk of information leakage from model updates.

Table 1 compares all the related work and our proposal in terms of the learning model employed, the availability of the code to allow the reproduction of experiments, the domain considered, the dataset used, the consideration of privacy budget, and explainability. The SECloHT-FL framework introduces a privacy-enhancing mechanism designed to strengthen resilience against adversarial threats in healthcare IoHT environments. The framework uses Locally Differential Privacy (LDP) to prevent the reverse engineering of individual client data and effectively mitigate the risk of malicious gradient manipulation. Compared to other approaches such as the FED-IIoT framework [23], which primarily addresses security in industrial IoT environments, SECloHT-FL focuses on the challenges faced by IoHT devices. These include heightened data sensitivity and heterogeneity, necessitating stronger privacy assurances during collaborative model training.

**Table 1.** Comparison to the related works.

References	Algorithm/Model	Code Availability	Domain	Dataset	Privacy Budget ( $\epsilon$ )	Explainability
[18]	DNN	×	IoHT	CICIDS2017	×	×
[19]	CNN RNN	×	IIoT	Edge-IIoTset	×	×
[20]	Reinforcement learning	×	IoHT	ToN-IoT	×	×
[21]	DNN	×	IIoT	Edge-IIoTset	✓	×
[22]	CNN	×	IoV	Car-hacking	×	×
[23]	FEDGAN	✓	IIoT	Drebin, Gnome, Contagio datasets	×	×
[24]	DNN-FL CNN-FL	×	IoHT	wustl-ehms-2020 ECU-IoHT	×	×
This work	DNN CNN	✓	IoHT	wustl-ehms-2020 ECU-IoHT	✓	✓

It is possible to observe that our work is the only privacy-preserving federated learning-based IDS that focuses on datasets from the IoHT. This sets our approach apart from others, even those that present themselves as IoHT-oriented. Our investigation revealed that the work [21] is the only one that addresses a privacy-preserving federated learning method for IIoT data, aiming to enhance privacy. However, in our research, we combine federated learning and  $\epsilon$ -differential privacy techniques to improve privacy while maintaining the confidentiality of health data in IoHT devices. Additionally, we share our source codes and experimental results on GitHub to ensure the reproducibility of our research and provide the explainability of our results.

### 3. The Proposed SECloHT-FL Method

This section covers the foundational concepts of federated learning (FL) and differential privacy (DP). We then introduce the architecture of our FL-based intrusion detection system (IDS) model and outline alternative learning approaches for comparison with our model.

#### 3.1. Federated Learning

FL is a machine learning approach designed to solve the “data island” problem. A data island occurs when a device has limited connectivity to other devices, making it difficult to share valuable information for a given application. FL also emphasizes data privacy preservation. In this approach, multiple clients, such as IoT devices, collaborate with one or more servers to implement decentralized learning configurations.

One of the key practical applications of FL has been demonstrated by Google, which uses FL to predict the next word entered by the user on the Android GBoard keyboard [25]. In this case, each user’s private data remain stored locally on their devices, eliminating the need for data transmission over the network. This reduces both the costs of data transfer (a significant contributor to mobile device battery consumption) and the privacy risks associated with sharing sensitive information.

In the context of our work, FL enables the implementation of learning models across multiple medical institutions and IoHT devices. It follows a client–server model where each IoHT device creates a local model based on its data. These local models are then aggregated on the server side to generate a unified global model, which is subsequently sent back to the clients for further refinement. This iterative process contributes to a collaborative and comprehensive learning system. A key advantage of FL is its emphasis on local training at medical institutions or devices, ensuring that sensitive data remain private and are not exposed to the global model.

#### 3.2. Differential Privacy

While FL enhances privacy compared to traditional machine learning methods by preventing direct data transmission to a central server, it still faces security vulnerabilities, and it has encountered challenges in healthcare, where privacy and security are paramount [26]. FL is susceptible to various attack types, and private information could potentially be inferred from the exchanged model updates. For instance, a membership inference attack can determine whether a particular sample was part of the training data based on the trained model, potentially revealing sensitive information in healthcare applications, such as disease classification models [27].

To address these security concerns and further protect privacy, one effective approach is the integration of privacy-enhancing technologies such as DP. The DP technique can reduce the risk of privacy leakage in the FL framework by adding random noise to the original data. The method masks sensitive private data in a dataset and prevents cyberattacks on FL using statistical probability models [28]. A randomized algorithm  $M$  is  $(\epsilon, \delta)$ -DP if for all datasets  $D$  and  $D'$  differ on at most one element, where  $P$  is a probability and all  $S \in \text{Range}(M)$ .

$$P[M(D) \in S] \leq e^\epsilon P[M(D') \in S] + \delta \quad (1)$$

In Equation (1),  $\delta$  captures the privacy protection of the Gaussian distribution. Furthermore,  $\epsilon$  shows the privacy leakage. The smallest  $\epsilon$  ensures the highest privacy.

There are two major categories of differential privacy algorithms: global differential privacy and local differential privacy. By using global differential privacy, information cannot be leaked from one site to another via the merging procedure on the central server. In the context of global differential privacy, the FL aggregation function undergoes per-



turbation by the server. This crucial step ensures that the aggregated output remains indistinguishable. However, in this method, each participant needs to trust the data curator both to share the updated model accurately and to perform the necessary perturbation by adding noise reliably [29]. The second DP method is local differential privacy, which prevents information from leaking from one site to the central server. It is a distributed variation of DP that ensures privacy for each local participant while eliminating the need for a trusted third party [30].

### 3.3. Architecture

We aim to develop local differential privacy approaches based on stochastic gradient descent (SGD) to ensure the privacy of gradients in FL. DP-SGD has the advantage of closely mimicking the classic stochastic gradient descent-based training of neural networks while applying differential privacy to deep learning models. This technique introduces differential privacy in the FL environment during local model training by perturbing model gradients. The DP-SGD algorithm starts by calculating per-sample gradients, clipping them to a predetermined threshold, and then aggregating them into a batch gradient. Finally, before updating the model parameters, DP-SGD adds Gaussian noise to the batch gradient. Many Python libraries, like Opacus for PyTorch (<https://opacus.ai/>, accessed on 12 November 2024) and TensorFlow Privacy for TensorFlow (<https://github.com/tensorflow/privacy>, accessed on 12 November 2024), implement the SGD approach.

Also, we used deep learning models, including CNNs and feedforward DNNs, as classifiers for each participant within the federation. We implemented these two deep learning methods with PyTorch in order to enhance the detection of anomalous behavior in IoHT network traffic. We performed extensive experiments, systematically adjusting model parameters to determine optimal values. In all experiments, we use PyTorch and Opacus to implement our privacy-preserving FL framework to detect potential attacks in IoHT applications.

#### Decentralized Model (FL-Based Architecture)

We evaluated the SECioHT-FL framework's performance by employing both feedforward DNN and CNN models as local classifiers. The federated learning process consisted of 100 rounds, where clients independently trained their models and shared differentially private gradients with the server for aggregation. Each communication round involved local model training using a batch size of 64 over 100 epochs per client.

Our proposed feedforward DNN architecture consists of one input layer, one hidden layer, and one output layer. It represents the 64 nodes ( $8 \times 8$ ) in the input layer, 32 nodes in the hidden layer, and 2 nodes in the output layer (because the target is 1 for attack instances and 0 for benign instances). For the forward function, we used the ReLU activation function and Softmax for output activation. The feedforward DNN is trained using the stochastic gradient descent (SGD) optimization algorithm with a learning rate of 0.01, employing binary cross-entropy as the loss function to reduce the prediction error.

Our proposed CNN architecture (a convolutional variant of the DNN model) comprises an input layer with 128 nodes, followed by three hidden convolutional layers with 64, 32, and 16 nodes, respectively. The output layer consists of 2 nodes, aligning with the binary nature of the target labels (1 for attack instances and 0 for benign instances). The forward function incorporates ReLU activation for hidden layers and Softmax activation for the output layer. The model was trained with SGD and binary cross-entropy, similarly to the DNN, ensuring consistency in optimization. Table 2 depicts the parameters common to both the feedforward DNN and CNN architectures.

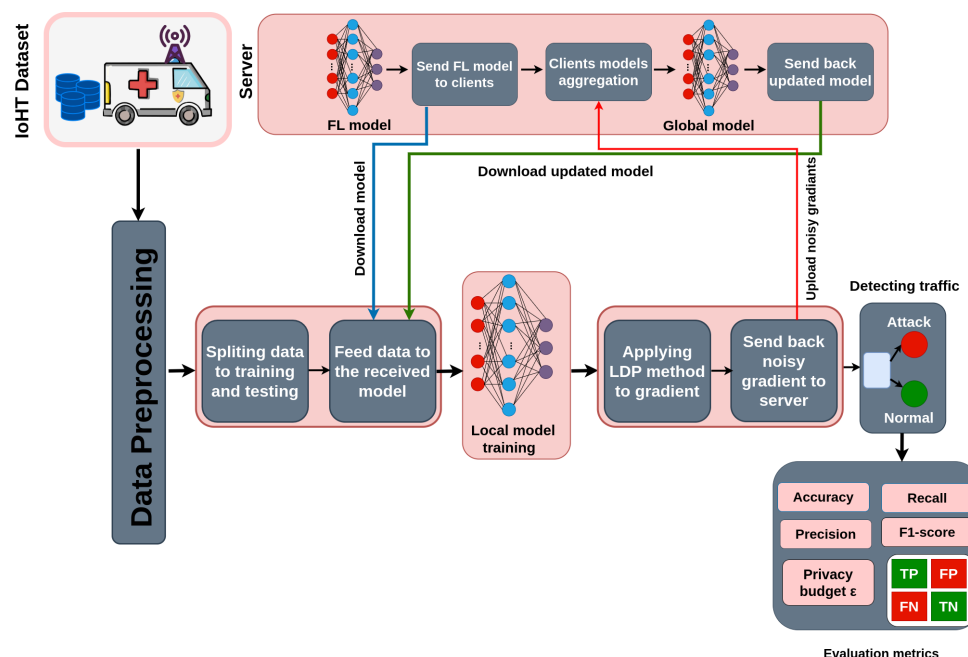
Privacy preservation was implemented using the Opacus PrivacyEngine, which applied differential privacy during local model training. Gradients were clipped to a maximum norm of  $10^{-4}$  to maintain training stability, and Gaussian noise with a multiplier of 0.5 and 1.5 was added to each gradient to increase privacy guarantees. These privacy-preserving mechanisms safeguarded sensitive data while maintaining robust performance. To identify the optimal noise level, we tested various noise values and evaluated their effect on performance metrics, including accuracy, precision, recall, and F1-score. The results were analyzed to select a configuration that balances strong privacy guarantees with high model performance.

**Table 2.** Parameters common to feedforward DNN and CNN architectures.

Activation function	Relu
Output activation	Softmax
Loss function	Binary cross-entropy
Optimizer	SGD
Learning rate	0.01
Epochs	100
Batch size	64
Delta	$1 \times 10^{-4}$
Noise multiplier	0.5, 1.5

The overview of the SECloHT-FL-based IDS is shown in Figure 1. In the proposed model, the training and aggregation process is structured to address data heterogeneity among IoHT devices while ensuring effective synchronization. Each IoHT device (client) trains its local model using private data and sends updates to the server, which manages the global model.

- Local model training:** Each client holds its own sensitive data. Initially, during each communication, the local client receives a training model (feedforward DNN and CNN) from the server (blue arrow). The client uses its local data to train the received model for 100 epochs on the IoHT device, generating gradients from the updated parameters. To ensure privacy, Opacus' PrivacyEngine is employed, introducing differential privacy by adding Gaussian noise to the gradients. Next, the noisy gradients and parameters with the injected Locally Differential Privacy (LDP) noise are uploaded to the server (red arrow).
- Parameter aggregation:** Subsequently, the server calculates the average of uploaded gradients received from the clients and updates the global model with the averaging algorithm. Ultimately, the server transmits the aggregated global model back to the IoHT devices for the subsequent iteration (depicted by the green arrow). In fact, by introducing LDP noise, the retrieval of users' information through the reversing of their uploaded gradients becomes unfeasible for servers or attackers.
- Handling data heterogeneity:** The server performs the aggregation of client parameters based on the volume of data each client processes. This is achieved by calculating a scaling factor derived from the ratio of a client's data samples to the total data across all clients. The client's model parameters are scaled accordingly before aggregation, ensuring that clients with larger datasets contribute proportionally more to the global model.
- Convergence criteria:** The training process runs iteratively until the convergence conditions are satisfied. These conditions include a fixed number of communication rounds (100 epochs) and a minimal reduction in loss between consecutive rounds ( $\delta < 10^{-4}$ ).



**Figure 1.** Architecture of our SECioHT-FL for anomaly detection.

Finally, we evaluate the performance of our proposed classifier based on confusion matrices, accuracy, recall, precision, F1-score, and the privacy budget. The bounds for the privacy budget  $\epsilon$  can vary depending on the application and configuration [31]. The theoretical lower bound of  $\epsilon$  in differential privacy is greater than 0 [11]. Common lower bounds in practice are typically around 0.1 or 0.5, though specific applications may use slightly lower values [32]. The upper bound of  $\epsilon$  is more flexible, depending on the acceptable trade-off between privacy and utility for a given application. In practice,  $\epsilon$  values between 1 and 10 are often considered reasonable, with values above 10 providing weaker privacy guarantees [33].

#### 4. Performance Evaluation

We compare our proposed SECioHT-IDS with conventional FL models and centralized machine learning methods, including feedforward DNN, LSTM, and CNN-LSTM. Additionally, we benchmark the SECioHT-IDS framework against results obtained from the same authors using the wustl-ehms-2020 dataset with the Support Vector Machine (SVM) method, as outlined in [15]. To facilitate the reproduction of our experiments and check the results, the source code is available for download in the GitHub public repository (<https://github.com/fatemehm/SECioHT-FL-based-IDS>, accessed on 12 November 2024).

##### 4.1. Experimental Setup

The models were developed using Python, with PyTorch 1.8.0 as the primary library for building the learning models. The experiments were conducted on a local machine featuring an Intel Core i5-7200U CPU @ 2.50GHz (×4), 8 GB of RAM, and a 256 GB hard drive, running Ubuntu 20.04.6 LTS. Additionally, we utilized the Pandas framework for dataset loading and manipulation.

##### 4.2. Datasets

In our performance evaluation, we focused on analyzing network traffic generated by IoHT devices. For this purpose, we evaluated the models using two distinct publicly available IoHT datasets, each featuring different characteristics. The following subsections provide detailed descriptions of each dataset.



#### 4.2.1. Wustl-Ehms-2020

The wustl-ehms-2020 dataset [15] was generated using a specialized testbed that incorporates a multi-sensor board designed for the comprehensive measurement of biometric data from patients' bodies. This testbed features a computer interface connected to the multi-sensor board, enabling data transmission to a server for storage and analysis to inform medical decisions. In parallel, another computer within the network is used to simulate attacks. The dataset contains 36 features, including both network and biometric data, and classifies attacks into two categories: spoofing attacks and data alteration. It comprises 2046 attack instances and 14,272 instances of normal behavior. In these scenarios, the attacker attempts to intercept packets, manipulate them through spoofing or alteration, and then reroute them to the server. The medical board consists of four key sensors, as follows:

- An electrocardiogram sensor, capable of capturing the electrical signals from the heart.
- An oxygen saturation sensor, designed to measure the oxygen levels in the blood.
- A temperature sensor, responsible for accurately measuring the patient's body temperature.
- A blood pressure sensor, which captures both systolic and diastolic pressure readings.

The authors describe that data are sent from the sensors to the gateway computer via a USB connection and then forwarded to the server over Wi-Fi using the TCP/IP protocols stack. However, the specific application protocols used for data transfer (such as HTTP or MQTT) between the gateway and the server are not mentioned.

#### 4.2.2. ECU-IoHT

The ECU-IoHT dataset [16] was generated in a comprehensive experimental setup. The environment for dataset generation includes Windows 10, Kali Linux, a mobile Wi-Fi hotspot, a wireless network adapter, and a Bluetooth adapter to facilitate Internet connectivity for the hosts. Additionally, the setup incorporates a healthcare kit, MySignals, which is equipped with various components and multiple sensors. These sensors monitor and store a range of biometric data, including the following:

- Temperature sensor.
- Blood pressure sensor.
- Heart rate sensor.

Each sensor collects data from the patient's body and transmits the information to the user's cloud. The paper describes the development and testing of the ECU-IoHT dataset but does not provide specific details about the application protocols used for data transfer for each sensor.

The dataset includes seven features related to network data, such as source, destination, protocol, and the nature of attacks. It contains 87,754 instances of normal behavior and 23,453 instances of attacks. The attack classes in the ECU-IoHT dataset are ARP spoofing, DoS, nmap port scan, and smurf.

#### 4.3. Data Preprocessing

Before feeding the models with the data, we first removed any NaN values and duplicate records. The "Label" column was binary-encoded, with normal instances labeled as 0 and attack instances labeled as 1. Due to the class imbalance in the datasets, we applied an oversampling technique to balance the data. Finally, feature scaling was performed using the StandardScaler() function to normalize the data within a range of 0 to 1.

#### 4.4. Baseline Models

To evaluate the efficiency of our proposed model, we conducted a comprehensive performance assessment, comparing it with two FL-based architectures that do not employ

differential privacy. This comparison aims to determine whether integrating differential privacy affects the model's accuracy. Additionally, we benchmarked our model against three centralized learning models from our previous research, which are based on a basic feedforward DNN, Long Short-Term Memory (LSTM), and CNN-LSTM [24]. This comparative analysis provides insights into the strengths and capabilities of our proposed approach, highlighting its advantages over both FL and centralized learning methods.

When implementing FL without incorporating DP, we used feedforward DNN and CNN architectures. The configurations of the input, hidden, and output layers were maintained consistently with those used in SECioHT-FL.

In the centralized deep learning method, the implemented feedforward DNN model is composed of three layers, with 64 neurons in the first and second layers and 32 neurons in the third layer. The activation function employed is ReLU following each DNN layer. The learning rate and batch size were set at 0.1 and 20, respectively. Additionally, the sigmoid activation function was utilized, aligning with our target range of values between 0 and 1.

The second centralized model is a LSTM network comprising one input layer and one output layer. For the LSTM model, we adopted binary cross-entropy as the loss function and employed the Adam optimizer. Key parameters such as LSTM size, dropout rate, learning rate, and epochs were configured at 4, 0.5, 0.1, and 100, respectively.

The final centralized model is the CNN-LSTM, characterized by a single input layer and a corresponding output layer. In this model, we employed the ReLU as the activation function, Adam as the optimizer, and binary cross-entropy as the loss function. The LSTM size was set to 20, while the dropout rate, learning rate, and epochs were configured at 0.2, 0.1, and 100, respectively.

#### 4.5. Results for the Wustl-Ehms-2020 Dataset

Table 3 presents the results obtained when detecting attacks in the wustl-ehms-2020 dataset. When applying SECioHT-FL, both feedforward DNN and CNN models achieved an F1-score of 96%. Additionally, the feedforward DNN and CNN models had similar accuracies of 93.20% and 93.10%, respectively, with training times of 0.045 and 0.039 s.

Within the SECioHT-FL framework, we employed DP to ensure the privacy of gradients in the FL process. Clients implement an  $\epsilon$ -DP mechanism on the gradient before transmitting it to the central server. After  $n$  epochs ( $n = 100$ ), the  $\epsilon$ -DP mechanism results in the  $(n \times \epsilon)$ -DP mechanism. This means that the privacy leakage is initially  $\epsilon$  at the first epoch and increases to  $n \times \epsilon$  after  $n$  epochs. In the SECioHT-FL framework,  $\epsilon$  is the privacy budget, and it is crucial to keep it within the range of 0 to 1, where a larger  $\epsilon$  indicates a less private model. Indeed, a lower epsilon value corresponds to a higher degree of privacy protection. As shown in Table 3, the privacy budgets of the feedforward DNN and CNN methods are 0.44 and 0.43, respectively, indicating a satisfactory level of privacy protection achieved by the proposed model.

**Table 3.** Results of accuracy, precision, recall, F1-score, and privacy budget for SECioHT-IDS compared to the baseline (FL without DP and centralized deep learning methods)—wustl-ehms-2020.

Methods	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Train Time (s)	Test Time (s)	Privacy Budget
FL without DP method	DNN	94.48	95	99	97	0.040	0.010	-
	CNN	94.37	95	99	97	0.055	0.019	-
SECioHT-FL	DNN (Noise = 1.5)	93.20	94	99	96	0.045	0.013	0.44
	CNN (Noise = 1.5)	93.10	94	99	96	0.039	0.013	0.43
	DNN (Noise = 0.5)	92.74	94	98	96	0.038	0.015	6.69
	CNN (Noise = 0.5)	89.44	89	100	94	0.036	0.011	6.69
Deep learning [24]	DNN	90.53	93	88	90	0.039	0.014	-
	LSTM	76.69	79	73	76	0.045	0.015	-
	CNN-LSTM	89.08	86	93	90	0.043	0.016	-
EHMS [15]	SVM	92.40	-	-	-	0.21	0.05	-

We conducted a comparative analysis between our SECioHT-FL proposal and the SVM model presented by the creators of the wustl-ehms-2020 dataset. Our feedforward DNN model in the SECioHT-IDS framework achieved the best accuracy of 93.20%, compared to 92.40% for the SVM model. Moreover, the training times of the DNN and CNN models in the SECioHT-IDS are 0.045 and 0.039 s, respectively, compared to 0.21 s for the SVM. Furthermore, the testing times for both of our models are 0.013 s, compared to 0.05 s for the SVM, demonstrating that our models run faster in both training and testing.

We tested the SECioHT-FL framework at two different noise levels (Noise = 1.5 and Noise = 0.5) to assess the trade-off between privacy and model performance. As shown in Table 3, the privacy budget ( $\epsilon$ ) increases significantly with lower Noise (0.5), with  $\epsilon = 6.69$ , compared to higher Noise (1.5), where  $\epsilon = 0.43$  for the CNN and 0.44 for the DNN. The results indicate that increasing the noise level enhances privacy guarantees without significantly impacting model performance. The DNN models maintained comparable accuracy (93.20% to 92.74%) and F1-scores (96%), while the CNN accuracy dropped more noticeably (93.10% to 89.44%). These findings underscore the SECioHT-FL framework's robustness under strong privacy settings and highlight the need to carefully select the appropriate noise level to balance privacy and utility.

We also compared the results of our proposed model (SECioHT-FL) with the FL without DP method. The results show that the DNN (in the FL without DP method) achieved an accuracy of 94.48%. Although this is 1.28% better than the accuracy of our proposed basic DNN (in SECioHT-FL), it may pose potential data exposure risks during federated updates. When comparing the performance of our proposed models with the centralized deep learning method, we found that the DNN model (in SECioHT-FL) achieved an accuracy that is 2.67% better than the centralized DNN model. Additionally, the centralized model lacks the decentralized privacy benefits and is more vulnerable due to centralized data storage.

Table 4 presents the confusion matrix for the CNN and basic feedforward DNN models with and without DP. The results indicate that in the SECioHT-FL framework, the DNN method accurately detected 24,047 attack instances as attack instances (true positives) and classified 1394 normal instances as attack instances (false positives), while the CNN model in the SECioHT-FL correctly identified 19,282 attack instances as attack instances (true positives) and accurately classified 1026 normal instances as normal instances (true negatives). Therefore, we note that the differentially private DNN model exhibits a higher number of detected attack instances compared to the other models, which is crucial for intrusion detection. Furthermore, we employed FL-based feedforward DNN and CNN models in an identical setting without applying DP to investigate whether adding noise to the gradients impacts the detection of attacks. The outcomes of the FL approach without DP reveal that in the SECioHT-FL framework, the feedforward DNN model, when integrated with differential privacy, achieved the best performance in identifying attacks compared to the FL-based model without DP.

When the noise multiplier is lowered to 0.5, the DNN shows a slight decrease in true positives (19,293) and true negatives (303), while false positives and false negatives increase slightly to 1093 and 1294, respectively. For the CNN, performance drops significantly with a higher false negative count of 2319 and a true negative count of 0, indicating the poor classification of benign instances. This highlights the CNN's sensitivity to reduced noise levels and its inability to maintain robustness under weaker privacy settings ( $\epsilon = 6.69$ ).

Overall, SECioHT-FL with Noise = 1.5 proves to be the most effective choice for IoHT applications, offering strong privacy guarantees and balanced performance, particularly for DNN models. Lowering noise to Noise = 0.5 affects robustness, especially for CNNs, making it less ideal for sensitive environments.

**Table 4.** Results of loss, TP, TN, FP, and FN for wustl-ehms-2020 datasets.

Methods	Algorithm	TP	TN	FP	FN	Loss
SECIoHT-FL	DNN (Noise = 1.5)	24,047	346	1394	1510	0.0059
	CNN (Noise = 1.5)	19,282	221	1026	1258	0.0060
	DNN (Noise = 0.5)	19,293	303	1093	1294	0.0060
	CNN (Noise = 0.5)	19,636	0	0	2319	0.0066
FL without DP method	DNN	23,893	225	1736	1279	0.0058
	CNN	23,891	218	1495	1296	0.0058

#### 4.6. Results for the ECU-IoHT Dataset

Table 5 presents the results achieved by all models on the ECU-IoHT dataset. In the SECIoHT-IDS framework, the CNN model, with an accuracy of 95.48% and an F1-score of 96%, achieved the best result. Overall, when comparing the results of our proposed SECIoHT-IDS with FL-based models without DP and a centralized deep learning model, our differentially private CNN model achieved satisfactory results. Furthermore, as shown in Table 5, the privacy budget of our differentially private DNN and CNN models is 0.34, which shows the acceptable level of privacy protection provided by them.

**Table 5.** Results of accuracy, precision, recall, F1-score, and privacy budget for SECIoHT-IDS compared to the baseline (FL without DP and centralized deep learning models)—ECU-IoHT dataset.

Methods	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Train Time (s)	Test Time (s)	Privacy Budget
FL without DP method	DNN	95.80	92	100	96	0.048	0.012	-
	CNN	94.52	90	100	95	0.039	0.012	-
SECIoHT-FL	DNN (Noise = 1.5)	94.35	90	100	95	0.065	0.026	0.34
	CNN (Noise = 1.5)	95.48	92	100	96	0.055	0.021	0.34
	DNN (Noise = 0.5)	95.68	92	100	96	0.041	0.012	5.20
	CNN (Noise = 0.5)	94.17	90	100	95	0.046	0.022	5.21
Deep learning [24]	DNN	94.56	90	100	95	0.028	0.030	-
	LSTM	94.48	90	100	95	0.043	0.016	-
	CNN-LSTM	94.72	91	100	95	0.042	0.015	-

In comparing our SECIoHT-FL model with the FL method without differential privacy (DP), we note that while the feedforward DNN in the FL without DP model achieves a slightly higher accuracy of 95.80%, which is 0.32% more than our CNN model in SECIoHT-FL, it poses a potential risk of sensitive data leakage. Additionally, the result of the centralized deep learning model shows that our CNN model in SECIoHT-FL achieved results very similar to the deep learning models. However, the centralized approach lacks the decentralized privacy advantages offered by federated models like SECIoHT-FL and is more vulnerable to security risks due to its reliance on centralized data storage.

The performance comparison of SECIoHT-FL with noise multipliers of 0.5 and 1.5 on the ECU-IoHT dataset highlights the trade-offs between privacy and model accuracy. At a noise multiplier of 1.5, the privacy budget ( $\epsilon$ ) stands at 0.34, which offers strong privacy protections. The DNN model achieves an accuracy of 94.35%. The CNN model, under the same settings, slightly edges out the DNN with an accuracy of 95.48%. These results indicate that SECIoHT-FL maintains strong privacy while still performing competitively in IoHT applications. In contrast, when the noise multiplier is reduced to 0.5, the privacy budget increases to 5.20, suggesting weaker privacy guarantees. The DNN model shows a slight accuracy boost, reaching 95.68%, but the CNN model's accuracy drops slightly to 94.17%. These results indicate that reducing the noise level improves DNN performance but slightly degrades CNN performance.

The comparison reveals the trade-off between privacy and performance in differential privacy mechanisms. A lower noise multiplier (0.5) improves DNN accuracy but at the expense of weaker privacy guarantees. Conversely, a higher noise multiplier (1.5) offers

better privacy protection with only a minor impact on performance, especially for CNN models. In conclusion, SECIOHT-FL with Noise = 1.5 offers a more favorable trade-off between privacy and performance, making it the optimal choice for IoHT applications that require robust privacy protections. The use of Noise = 0.5 is best suited for cases where privacy needs are less critical, and minor gains in DNN accuracy are acceptable.

Table 6 shows the confusion matrix results for the FL models applied to the ECU-IoHT dataset. Our findings demonstrate that in the SECIOHT-IDS framework, the feedforward DNN model successfully classified 22,116 attack instances as true positives (TPs) and identified 19,333 normal instances as false positives (FPs). Compared to the CNN model, the basic feedforward DNN model in SECIOHT-IDS achieved the best results in detecting attack instances. Moreover, when comparing the results of SECIOHT-IDS with the FL-based models without the DP method in an identical setting, excluding privacy considerations, we observe that incorporating differential privacy into the FL-based models and adding noise to the gradients does not compromise the effectiveness of our proposed model in attack identification.

**Table 6.** Results of loss, TP, TN, FP, and FN—ECU-IoHT dataset.

Methods	Algorithm	TP	TN	FP	FN	Loss
SECIOHT-FL	DNN (Noise = 1.5)	22,116	0	19,333	2481	0.0058
	CNN (Noise = 1.5)	21,824	1	20,110	1821	0.0055
	DNN (Noise = 0.5)	17,343	0	15,916	1500	0.0056
	CNN (Noise = 0.5)	17,460	0	15,304	2028	0.0058
FL without DP method	DNN	17,614	0	16,198	1482	0.0056
	CNN	17,452	0	15,587	1915	0.0058

The performance analysis of SECIOHT-FL under noise multipliers of 1.5 and 0.5 reveals the trade-offs between privacy and accuracy. At Noise = 1.5, the DNN model has 22,116 true positives (TP) and 19,333 false positives (FP), with no true negatives (TN = 0) and 2481 false negatives (FN). The CNN model has slightly fewer true positives (21,824) and one true negative (TN = 1), with a higher false positive count of 20,110 and a lower false negative count of 1821, suggesting better recall. The FP count for the CNN increases to 20,110, while the FN count decreases to 1821, suggesting better recall compared to the DNN. Loss values are stable across both models. When the noise multiplier is reduced to 0.5, the DNN's true positives decrease to 17,343, with improved precision and fewer false positives (15,916). The true negatives remain at zero, and false negatives drop to 1500. The CNN model shows similar trends with 17,460 true positives, 15,304 false positives, and no true negatives, with an FN count of 2028. Loss values remain stable across both models.

Comparing the two noise levels, Noise = 1.5 improves sensitivity to positive instances, indicated by higher true positive counts for both DNN and CNN models. However, this higher noise also leads to more false positives, reducing precision. On the other hand, Noise = 0.5 lowers false positives, improving precision, while also decreasing false negatives, suggesting better recall. The trade-off is evident in the privacy guarantees: Noise = 1.5 offers a stronger privacy budget ( $\epsilon = 0.34$ ), but Noise = 0.5 weakens privacy ( $\epsilon = 5.20$ ).

In summary, SECIOHT-FL with Noise = 1.5 offers a better balance between privacy and performance, especially for applications requiring strong privacy guarantees. Reducing noise to Noise = 0.5 enhances precision and recall but at the cost of significantly weaker privacy. For privacy-sensitive IoHT applications, Noise = 1.5 is the preferred choice, while Noise = 0.5 may be better suited for scenarios where detection performance takes priority over privacy.



## 5. Explainable AI

In this section, we explore the reasons behind the anomaly detection performance of our proposed model. As outlined in Section 4.2, we trained our model on two IoHT datasets. We employed the SHAP framework to analyze how the features contribute to the detection performance of our models. SHAP calculates the importance score of each feature. The most important features learned by our proposed model from the two datasets are illustrated in Figures 2 and 3.

The beeswarm plot in SHAP highlights the most important features. Each dot in the plot represents the Shapley value of a feature for a specific sample in the dataset. On the x-axis, the position indicates the Shapley values, while the y-axis shows the features, ordered by their importance. Each dot is colored according to the value of the feature for that sample. For example, red represents a high impact, purple indicates a medium effect, and blue signifies a low value of the feature.

Figure 2 shows the Shapley values ranging from  $-0.8$  to above  $0.4$  for the wustl-ehms-2020 dataset. The following features are considered the most important in detecting attacks:

- **Dur:** This feature refers to the duration of the flow. Duration is the most important feature, with higher durations (red points) strongly contributing to the prediction of an intrusion (moving the SHAP value to the right). Low values of “Dur” have little to no impact on the model’s attack detection.
- **Sport:** Refers to the source port. As shown in the figure, the source port plays a critical role, with higher values (red) pushing the model toward detecting an intrusion. High source port numbers may correlate with specific attack patterns or protocols used in intrusions.
- **Temp (temperature) and pulse rate:** These two biometric features are crucial in the IoHT intrusion detection model. Higher temperatures (red) and pulse rates contribute positively to intrusion detection, potentially indicating abnormal behavior associated with certain health conditions being monitored.
- **DstJitter (destination jitter) and SrcJitter (source jitter):** DstJitter: High jitter values (red) contribute positively to detecting intrusions, as they may indicate irregularities in network traffic.
- **Heart Rate and Resp Rate (respiratory rate):** Similar to temperature and pulse rate, these features may indicate health anomalies that align with network intrusions in an IoHT setting. Higher values push the model toward predicting an intrusion.

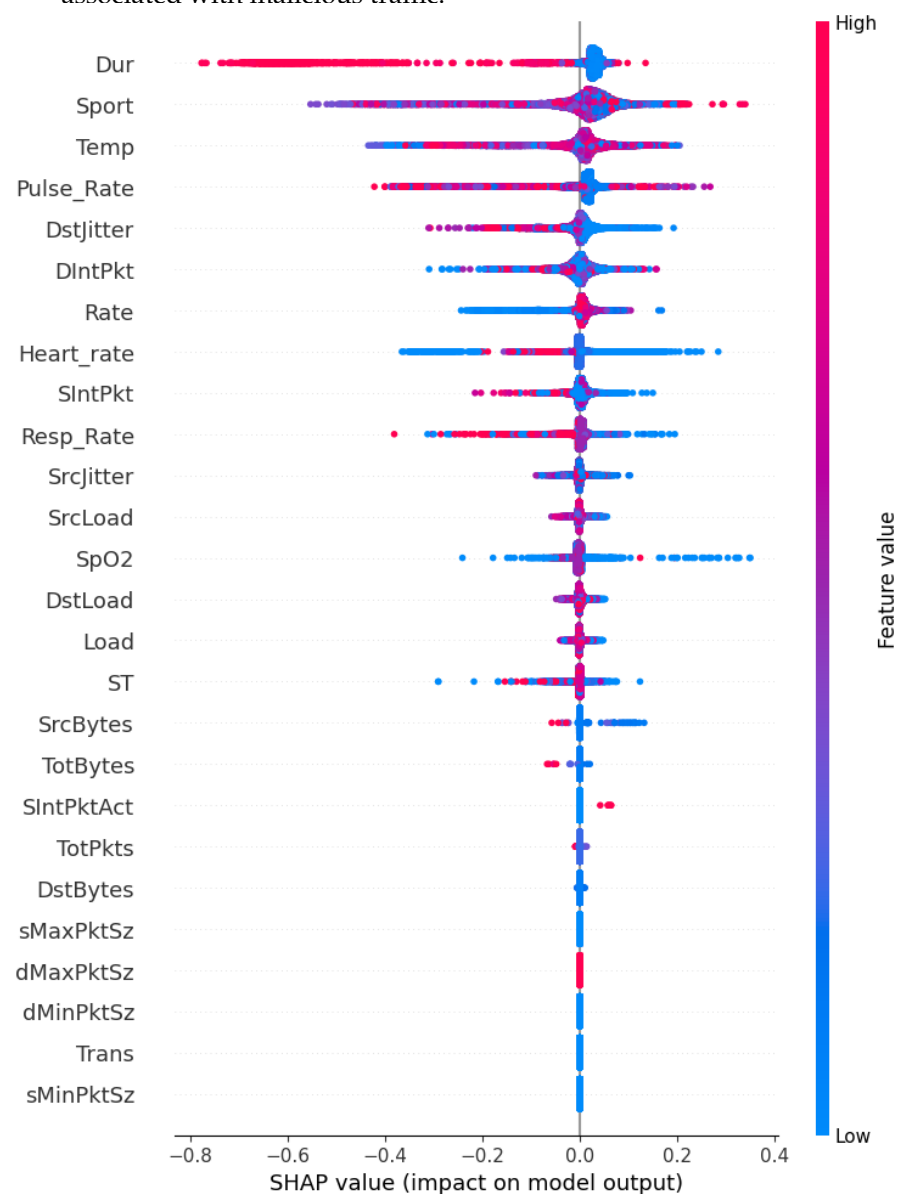
Figure 3 presents the beeswarm plot for the ECU-IoHT dataset, showing Shapley values ranging from  $-0.4$  to above  $0.4$ . The following features are considered the most important for detecting attacks, listed in order of their significance:

- **Length:** The packet or data length is the most important feature in this model. Lower values (blue points) have a negative SHAP value, meaning they reduce the probability of an intrusion. Higher values (red points) are spread between positive and negative SHAP values, indicating that larger packet sizes have a mixed effect, depending on the specific scenario. This suggests that both unusually high and low packet sizes are important for detecting certain types of attacks.
- **Protocol:** Higher values for each protocol (red points) contribute positively to predicting intrusions, while lower values (blue points) show less contribution. This indicates that certain protocols, such as ICMP and TCP, are strongly associated with intrusions, while others are not.
- **Destination:** Increasing the value of the “Destination” feature increases the probability of detecting an attack. Higher destination values (red points) are associated with a

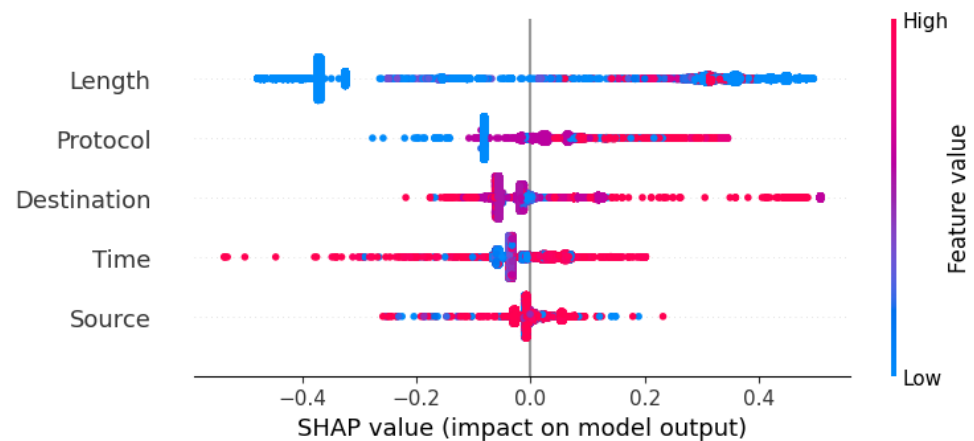


greater likelihood of intrusion detection, suggesting that specific destination addresses or ports are likely targeted during attacks.

- Time: Time also plays a key role. Higher values (red points) push the model toward predicting an intrusion, indicating that attacks might occur more frequently at certain times or during long sessions.
- Source: Higher source values (red points) have a stronger impact on pushing the model toward predicting an intrusion, while lower values (blue points) have less influence. This suggests that certain source IP addresses, devices, or ports are commonly associated with malicious traffic.



**Figure 2.** SHAP values for detected attacks in wustl-ehms-2020.



**Figure 3.** SHAP values for detected attacks in ECU-IoHT.

## 6. Conclusions and Future Works

In this paper, we proposed a privacy-preserving federated learning framework (SECIOHT-FL) for detecting attacks in IoHT network traffic. Our framework combines decentralized learning with  $\epsilon$ -differential privacy to design an FL-based intrusion detection system (IDS) for analyzing network traffic generated by IoHT devices and identifying cyber-attacks in the smart healthcare sector. We developed FL-based, differentially private DNN and CNN models to evaluate the privacy protection level of our proposed framework. For comparison, we also considered an FL-based model without applying  $\epsilon$ -differential privacy and a centralized deep learning model as baselines. We validated our framework using the wustl-ehms-2020 and ECU-IoHT datasets. The differentially private DNN model achieved the best performance on the wustl-ehms-2020 dataset, with an accuracy of 93.20%, an F1-score of 96, and a privacy budget ( $\epsilon$ ) of 0.44. For the ECU-IoHT dataset, the differentially private CNN model performed the best, with an accuracy of 95.48%, an F1-score of 96, and a privacy budget ( $\epsilon$ ) of 0.34. Our future work will focus on using real-time SHAP analysis in deployed environments to help healthcare providers understand why certain network activities are flagged as intrusions. Additionally, we will explore advanced differential privacy techniques, such as adaptive noise addition, to further enhance privacy guarantees in federated learning. Also, the aggregation server in federated learning combines client updates to form the global model. However, it is vulnerable to adversarial attacks like model poisoning. Enhancing the robustness of the aggregation process against more sophisticated attacks, such as targeted model poisoning, is an area for future development. Implementing advanced aggregation functions could improve the security and scalability of the framework.

**Author Contributions:** Conceptualization, F.M., S.P. and D.M.B.; methodology, F.M., S.P., M.H., L.L., Y.X., L.Z. and D.M.B.; investigation, F.M.; writing—original draft preparation, F.M.; writing—review and editing, F.M., S.P., M.H., L.L., Y.X., L.Z. and D.M.B.; supervision, S.P. and D.M.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior–Brasil (CAPES)–Finance Code 001, FAPESP proc. 14/50937-1 and proc. 15/24485-9. It is also part of the FAPESP proc. 21/06995-0. This research was additionally funded by the National Key Research and Development Program of China under Grant No. 2023YFB2704400.

**Data Availability Statement:** The data supporting the findings of this study are openly available on GitHub at <https://github.com/fatemehm/SECIOHT-FL-based-IDS> (accessed on 12 November 2024).

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

## References

1. Ahmed, J.; Nguyen, T.N.; Ali, B.; Javed, M.A.; Mirza, J. On the physical layer security of federated learning based IoMT networks. *IEEE J. Biomed. Health Inform.* **2022**, *27*, 691–697. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Deebak, B.D.; Al-Turjman, F. Smart mutual authentication protocol for cloud based medical healthcare systems using internet of medical things. *IEEE J. Sel. Areas Commun.* **2020**, *39*, 346–360. [\[CrossRef\]](#)
3. Ferrag, M.A.; Shu, L.; Friha, O.; Yang, X. Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions. *IEEE/CAA J. Autom. Sin.* **2021**, *9*, 407–436. [\[CrossRef\]](#)
4. Taheri, R.; Arabikhan, F.; Gegov, A.; Akbari, N. Robust Aggregation Function in Federated Learning. In Proceedings of the International Conference on Information and Knowledge Systems, Edinburgh, UK, 11–13 August 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 168–175.
5. Mosaiyebzadeh, F.; Pouriyeh, S.; Parizi, R.M.; Sheng, Q.Z.; Han, M.; Zhao, L.; Sannino, G.; Ranieri, C.M.; Ueyama, J.; Batista, D.M. Privacy-Enhancing Technologies in Federated Learning for the Internet of Healthcare Things: A Survey. *Electronics* **2023**, *12*, 2703. [\[CrossRef\]](#)
6. Yang, H.; Ge, M.; Xue, D.; Xiang, K.; Li, H.; Lu, R. Gradient Leakage Attacks in Federated Learning: Research Frontiers, Taxonomy and Future Directions. *IEEE Netw.* **2023**, *38*, 247–254. [\[CrossRef\]](#)
7. Xia, G.; Chen, J.; Yu, C.; Ma, J. Poisoning Attacks in Federated Learning: A Survey. *IEEE Access* **2023**, *11*, 10708–10722. [\[CrossRef\]](#)
8. Nair, A.K.; Raj, E.D.; Sahoo, J. A robust analysis of adversarial attacks on federated learning environments. *Comput. Stand. Interfaces* **2023**, *86*, 103723. [\[CrossRef\]](#)
9. Chen, P.; Du, X.; Lu, Z.; Chai, H. Universal Adversarial Backdoor Attacks to Fool Vertical Federated Learning. *Comput. Secur.* **2023**, *137*, 103601. [\[CrossRef\]](#)
10. Aziz, R.; Banerjee, S.; Bouzefrane, S.; Le Vinh, T. Exploring Homomorphic Encryption and Differential Privacy Techniques towards Secure Federated Learning Paradigm. *Future Internet* **2023**, *15*, 310. [\[CrossRef\]](#)
11. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [\[CrossRef\]](#)
12. He, Z.; Wang, L.; Cai, Z. Clustered federated learning with adaptive local differential privacy on heterogeneous iot data. *IEEE Internet Things J.* **2023**, *11*, 137–146. [\[CrossRef\]](#)
13. Errounda, F.Z.; Liu, Y. Adaptive differential privacy in vertical federated learning for mobility forecasting. *Future Gener. Comput. Syst.* **2023**, *149*, 531–546. [\[CrossRef\]](#)
14. Ren, C.; Yu, H.; Yan, R.; Li, Q.; Xu, Y.; Niyato, D.; Dong, Z.Y. SecFedSA: A secure differential privacy-based federated learning approach for smart cyber-physical grid stability assessment. *IEEE Internet Things J.* **2023**, *11*, 5578–5588. [\[CrossRef\]](#)
15. Hady, A.A.; 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*, 106576–106584. [\[CrossRef\]](#)
16. Ahmed, M.; Byreddy, S.; Nutakki, A.; Sikos, L.F.; Haskell-Dowland, P. ECU-IoHT: A Dataset for Analyzing Cyberattacks in Internet of Health Things. *Ad. Hoc. Netw.* **2021**, *122*, 102621. [\[CrossRef\]](#)
17. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [\[CrossRef\]](#)
18. Otoum, Y.; Wan, Y.; Nayak, A. Federated transfer learning-based ids for the internet of medical things (iomt). In Proceedings of the 2021 IEEE Globecom Workshops (GC Wkshps), Madrid, Spain, 7–11 December 2021; pp. 1–6.
19. Rashid, M.M.; Khan, S.U.; Eusufzai, F.; Redwan, M.A.; Sabuj, S.R.; Elsharief, M. A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks. *Network* **2023**, *3*, 158–179. [\[CrossRef\]](#)
20. Khan, I.A.; Razzak, I.; Pi, D.; Khan, N.; Hussain, Y.; Li, B.; Kousar, T. Fed-Inforce-Fusion: A federated reinforcement-based fusion model for security and privacy protection of IoMT networks against cyber-attacks. *Inf. Fusion* **2024**, *101*, 102002. [\[CrossRef\]](#)
21. Friha, O.; Ferrag, M.A.; Benbouzid, M.; Berghout, T.; Kantarci, B.; Choo, K.K.R. 2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT. *Comput. Secur.* **2023**, *127*, 103097. [\[CrossRef\]](#)
22. Taslimasa, H.; Dadkhah, S.; Neto, E.C.P.; Xiong, P.; Iqbal, S.; Ray, S.; Ghorbani, A.A. ImageFed: Practical Privacy Preserving Intrusion Detection System for In-Vehicle CAN Bus Protocol. In Proceedings of the 2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), New York city, NY, USA, 6–8 May 2023; pp. 122–129.
23. Taheri, R.; Shojafar, M.; Alazab, M.; Tafazolli, R. FED-IIoT: A robust federated malware detection architecture in industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *17*, 8442–8452. [\[CrossRef\]](#)
24. Mosaiyebzadeh, F.; Pouriyeh, S.; Parizi, R.M.; Han, M.; Batista, D.M. Intrusion Detection System for IoHT Devices using Federated Learning. In Proceedings of the IEEE INFOCOM 2023—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York City, NY, USA, 17–20 May 2023; pp. 1–6.
25. Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; Ramage, D. Federated learning for mobile keyboard prediction. *arXiv* **2018**, arXiv:1811.03604.

26. Rahman, A.; Hossain, M.S.; Muhammad, G.; Kundu, D.; Debnath, T.; Rahman, M.; Khan, M.S.I.; Tiwari, P.; Band, S.S. Federated learning-based AI approaches in smart healthcare: Concepts, taxonomies, challenges and open issues. *Clust. Comput.* **2023**, *26*, 2271–2311. [[CrossRef](#)]
27. Li, H.; Li, C.; Wang, J.; Yang, A.; Ma, Z.; Zhang, Z.; Hua, D. Review on security of federated learning and its application in healthcare. *Future Gener. Comput. Syst.* **2023**, *144*, 271–290. [[CrossRef](#)]
28. El Ouadrhiri, A.; Abdelhadi, A. Differential privacy for deep and federated learning: A survey. *IEEE Access* **2022**, *10*, 22359–22380. [[CrossRef](#)]
29. Naseri, M.; Hayes, J.; De Cristofaro, E. Local and central differential privacy for robustness and privacy in federated learning. *arXiv* **2020**, arXiv:2009.03561.
30. Wang, B.; Chen, Y.; Jiang, H.; Zhao, Z. PPeFL: Privacy-Preserving Edge Federated Learning with Local Differential Privacy. *IEEE Internet Things J.* **2023**, *10*, 15488–15500. [[CrossRef](#)]
31. Yousefpour, A.; Shilov, I.; Sablayrolles, A.; Testuggine, D.; Prasad, K.; Malek, M.; Nguyen, J.; Ghosh, S.; Bharadwaj, A.; Zhao, J.; et al. Opacus: User-friendly differential privacy library in PyTorch. *arXiv* **2021**, arXiv:2109.12298.
32. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.
33. Mironov, I. Rényi differential privacy. In Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF), Santa Barbara, CA, USA, 21–25 August 2017; pp. 263–275.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.