Intrusion Detection System for IoHT Devices using Federated Learning

Conference Paper · May 2023

DOI: 10.1109/INFOCOMWKSHPS57453.2023.10225932

5 authors, including:

Fatemeh Mosaiyebzadeh
University of São Paulo
6 PUBLICATIONS **47** CITATIONS
SEE PROFILE

Seyedamin Pouriyeh
Kennesaw State University
63 PUBLICATIONS **3,298** CITATIONS
SEE PROFILE

Reza M. Parizi
Kennesaw State University, Atlanta, USA
208 PUBLICATIONS **10,568** CITATIONS
SEE PROFILE

IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-9427-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOMWKSHPS57453.2023.10225932

Intrusion Detection System for IoHT Devices using Federated Learning

Fatemeh Mosaiyebzadeh*, Seyedamin Pouriyeh†, Reza M. Parizi‡, Meng Han §, Daniel Macêdo Batista*
* Department of Computer Science, University of São Paulo, Brazil
{fatemehm, batista}@ime.usp.br
† Department of Information and Technology, Kennesaw State University, Marietta, GA, USA
spouriye@kennesaw.edu
‡Decentralized Science Lab, Kennesaw State University, Marietta, GA, USA
rparizi1@kennesaw.edu
§ Binjiang Institute of Zhejiang University, Hangzhou, Zhejiang, China
mhan@zju.edu.cn

***Abstract*—With the growing number of sensitive data transmitted in IT infrastructures, healthcare organizations and companies that generate users' wearable data have become a target for attackers. To protect electronic healthcare data, Internet of Healthcare Things (IoHT) devices must be protected by robust Intrusion Detection Systems (IDS) to provide a secure environment. Since it is undesirable to collect this data and perform machine learning tasks directly, recently, to preserve users' privacy, federated learning has obtained attention from the government and healthcare organizations. Unlike the centralized paradigm, federated learning is a privacy-aware machine learning framework designed to analyze data without sharing it. This paper proposes a deep neural network in federated learning (DNN-FL) to detect anomalies in the IoHT traffic that may result in security threats. We evaluate the detection performance of our proposal using metrics such as accuracy and precision. The proposed DNN-FL is validated using the wustl-ehms-2020 and ECU-IoHT datasets. It reached 91.40% of accuracy in detecting attacks in the first dataset and 98.47% in the second. All the developed source code in this work is being made publicly available to ensure reproducibility.**

***Index Terms*—IoHT, Federated Learning, Deep Learning, Intrusion Detection.**

I. Introduction

The Internet of Things (IoT) is a growing technology with a broad range of applications consisting of globally connected everyday devices via the Internet. According to Statista [1], in 2025, the total number of IoT devices in the world will be about 75 billion, and 30.3% of these connected devices will be Internet of Healthcare Things (IoHT) devices. IoHT devices are designed with various sensors to monitor the human body, store health data and transmit healthcare data. For instance, smart watches equipped with accelerometers and pulse oximeters, can monitor patient data remotely and notify medical staff about patients' health problems. In general, it can be said that IoHT devices collect extremely sensitive health data, for which security and privacy protection are essential.

A large number of IoT devices are susceptible to cyberattacks and misuse due to their insecure design, implementation, and configuration. IoT devices are subject to a wide range of cyber attacks, including Ransomware, botnets, remote code execution, and DDoS attacks [2] [3] [4]. In order to protect IoT networks and nodes, Intrusion Detection Systems (IDSs) are used to track traffic, detect suspicious activities, and mitigate the negative effect of cyberattacks. IDSs detect intrusions based on the relationship between signatures previously learned or by monitoring and comparing network traffic with previously learned patterns, looking for anomalies. However, sometimes IDSs are ineffective in detecting new and unknown adversarial attacks. In addition, the effectiveness of current IDSs is in doubt with an increasing number of IoT devices.

Machine Learning (ML), with its ability to design and implement robust intelligent systems, has tremendously impacted cybersecurity solutions. Also, Deep Learning (DL) methods, as a sub-topic of ML, have shown their suitability for various data-driven issues such as those in cybersecurity. For instance, some DL methods have been recently used in several IDSs to detect anomalies [5]. However, using ML models for anomaly detection rely on computational power and needs to transmit sensitive data from devices to a centralized server. Despite this, due to regulations and policies such as the General Data Protection Regulation (GDPR), hospitals and organizations are not allowed to share data between them [6] [7].

To address the data privacy challenge and provide a more secure environment for patients' private data, Federated Learning (FL) has gained much attention. FL is a combination of federated and machine learning methods that can detect anomalies in a decentralized environment without compromising patients' data [8] [9] [10]. It enables the training procedure done by IoT devices using the private datasets distributed across different devices. Previous works show that the FL model should perform equivalent to the centralized training model in terms of accuracy and prediction [11].

The objective of this paper is to propose an FL-based IDS for IoHT networks, trained with two datasets generated by IoHT devices. Also, this paper advances the state-of-the-art by improving the open-source IDS for IoHT without a need

978-1-6654-9427-4/23/$31.00 ©2023 IEEE

for sharing private data with organizations. We trained the FL model by using the public datasets `wustl-ehms-2020` [12], which contains patients' data (heart rate, level of oxygen in the blood, temperature and systolic and diastolic pressure), and two types of attacks, and `ECU-IoHT` [13], which contains various biometrics data (blood pressure, heart rate, temperature), and four types of attacks. To the best of our knowledge, this is the first paper applying FL on the IoHT datasets aiming at the implementation of an Intrusion Detection System (IDS) and being validated with these specific datasets. Our IDS applies Deep Neural Network in FL (DNN-FL). The experiments to evaluate our proposed DNN-FL model showed that it achieved 91.40% of accuracy for attack detection in `wustl-ehms-2020` and 98.47% in `ECU-IoHT`.

The rest of this paper is organized as follows: We present related work about IoT and IoHT security in Section II. We present some background about FL, our research methodology, datasets used in this paper, and data preprocessing in Section III. In Section IV, we provide the performance evaluation of our proposal. Conclusions and future work are presented in Section V.

II. Related work

FL is a concept to preserve data privacy and one of the most prominent approaches for anomaly detection in IoHT traffic. It is used for training models locally without transferring private data to a central server, significantly increasing the privacy and security of the very sensitive data. Recently, FL has been used for anomaly detection to increase the privacy over the client's data and decrease the response time.

In [14], the authors proposed a privacy-preserving, lightweight user authentication scheme for healthcare-related IoT applications. The proposed scheme uses a lightweight hash function and XOR operations. As a result, the scheme can protect IoHT environments against some types of attacks like replay, Man In The Middle (MITM), and Denial of Service (DoS).

Muhammad et al. [15] presented a method to identify and classify contaminants in surface electromyography (sEMG) signals using a 1D convolutional neural network (1D-CNN). The experiments showed that the 1D-CNN enables a secure environment for IoHT devices to avoid sending false information to the cloud.

In [16], the authors proposed a novel scheme to detect adversarial attacks, which manipulate some features to confuse a machine learning model and produce a wrong prediction. They presented a deep learning model called Self-normalizing Neural Network (SNN) to classify intrusion attacks in IoT networks. Also, they used Feedforward Neural Networks (FNN) technique as a baseline to compare with the performance of SNN. They studied each technique under real traffic datasets, such as the BoT-IoT dataset published by the Cyber Range Lab of UNSW Canberra.

Bovenzi et al. [17] proposed a Hierarchical Hybrid Network Intrusion Detection (H2ID) approach to detect anomalies by MultiModal Deep AutoEncoder (M2-DAE) and attack classification using soft-output classifiers. They validated the proposed IDS also using the BoT-IoT dataset, similar to [16].

In [18], the authors proposed privacy-preserving federated learning for IoT malware detection without sharing data. They considered four different multi-layer perceptrons (MLP) architectures with one output neuron. To evaluate the performance of the proposed model, they used the N-BaIoT dataset, which has the network traffic of real IoT devices affected by malware.

Among the discussed works, our approach is the only federated learning-based model focusing on IoHT traffic for anomaly detection. We compared the recent efforts in this domain side-by-side in Table I. We investigated and concluded that [14] and [15] are the two closely related works to our research, however; in our work, we combine federated with deep learning techniques for training health data stored on end devices. In addition, we share our experiments on GitHub, ensuring the research's reproducibility.

TABLE I
Comparison to the related works

References	Algorithm/ Model	Code-availability	IoHT-Environment	Federated-Learning-based
[14]	One-way hash function	×	✓	×
[15]	1D-CNN	×	✓	×
[16]	SNN FNN	×	×	×
[17]	H2ID	×	×	×
[18]	MLP	×	×	✓
Our work	DNN-FL	✓	✓	✓

Our research aims to study and provide an intrusion detection system for IoHT devices. The research proposes the federated learning-based deep neural network method (DNN-FL), which can be suitable for IoHT environments where the medical devices are connected over the network. More details of this proposal are discussed in section III.

III. Research methodology

In this section, we describe the general principle of FL. Afterward, we present the architecture of the FL model for our intrusion detection system and the proposed deep learning methods to compare with our FL model. In IV-B and IV-C, we also discuss the IoHT datasets used for our proposed models and data preprocessing techniques.

A. Background

In recent years, ML techniques in healthcare have shown promising results in assisting healthcare professionals and patients. In this context, different ML-based models have been proposed for early detection of various diseases. However, those models rely on a centralized environment where the ML models reside in a hospital or a healthcare center with the assumption that patient's data can be collected using IoHT devices to train the ML models. Such models usually suffer from performance and accuracy issues due to the unavailability of sufficient data to reside centrally on the server-side for training because of direct access restrictions on such data, or data privacy regulations (HIPAA and GDPR [19]), where all may lead to biased models that cannot be trustworthy. Additionally, even with sufficient data, the training procedure in a centralized setting is time-consuming and expensive tasks make them out of interest of hospitals and research centers. With the emergence of decentralized AI and FL, healthcare providers/professionals and patients can tremendously benefit from AI-assisted analysis of distributed clinical data directly without jeopardizing their privacy or their proprietary rights.

B. Architecture

In this paper, we consider a Deep Neural Network (DNN-FL) as the classifier for each client in FL to analyze network traffic and detect attacks in the IoHT environment. The architecture was designed based on [18], however, we integrated this architecture with the TensorFlow Federated (TFF) framework for anomaly detection in IoHT applications. Then we went through excessive experiments to tweak the model with different parameters to find the best matched values to be used.

1) Decentralized model (FL-based architecture): The proposed DNN-FL model has one input layer, three hidden layers with 64, 64, and 32 neurons, and one output layer. Table II shows our federated learning setup in detail. We used the same architecture and configurations for one of our centralized learning models, which will be presented in Subsection III-B2 and that were designed to serve as a baseline for our proposed DNN-FL model.

TABLE II
FEDERATED LEARNING SETUP

Number of participants	15, 10, 5
Activation function	Relu
Output activation	Sigmoid
Loss function	Binary crossentropy
Client optimizer	Adam
Client learning rate	0.01
Server optimizer	Adam
Server learning rate	0.05
Epochs (Federated model)	100
Batch size	1024
Training rounds	100

The overview of our proposed DNN-FL based-IDS is illustrated in Fig. 1. Initially, the server distributes the general DNN-FL model to clients who participate in the FL (The first blue arrow with dashed line). The client uses its local data to run the DNN-FL model locally on the IoT device. Lastly, the IoT device sends its DNN-FL model to the central server along with its model parameters and corresponding weights. (The red arrow with dashed line). Next, The server aggregates all the local models received from the clients using the averaging algorithm. The aggregation employed by the central server is a way to average the parameters of the models sent by the clients and combine all the single local models to update the global model. Finally, the server returns the aggregated global model to the IoT devices for the next iteration (The second blue arrow with dashed line).

2) Centralized models: To evaluate the performance of the DNN-FL, we compared it with the performance of three centralized DL models: Deep Neural Network (DNN), Long Short-Term Memory (LSTM), and a mix of Convolutional and Recurrent Neural Networks (CNN-LSTM). Fig. 2 presents an overview of these models in a centralized system. The first step of the system is uploading private data to the centralized server. The second phase is a dataset's preprocessing, such as removing null and duplicated values, balancing a dataset, and feature scaling, which is explained in Subsection IV-C. The training of the DL model is the last phase of the centralized system.

The implemented DNN model contains three layers with 64, 64, and 32 neurons. The activation function is a ReLU that follows the DNN layer. The learning rate and batch size are equal to 0.1 and 20, respectively. Also, we use the sigmoid function because our target is between 0 and 1.

The second implemented centralized model is an LSTM that contains one single input layer and one output layer. In the LSTM model, we used the binary Cross-Entropy as a loss function and Adam as an optimizer. Also, the LSTM size, dropout, learning rate, and epochs are 4, 0.5, 0.1, and 100, respectively.

The last implemented centralized model is the CNN-LSTM. It contains one input layer and one single output layer. In the proposed CNN-LSTM model, we used ReLU as an activation, Adam as an optimizer, and binary-cross-entropy as a loss function. Also, the LSTM size is 20, and the dropout, learning rate, and epochs are 0.2, 0.1, and 100, respectively.

We applied the GridSearchCV function from Scikit-learn to obtain the optimal parameters of all the models [20]. They are implemented using TensorFlow, Keras, and TensorFlow-Federated (TFF) [21].

IV. Performance Evaluation

We compare our proposed DNN-FL with the centralized methods DNN, LSTM, and CNN-LSTM. Besides, we also compare the performance of our DNN-FL architecture with the results obtained by Support Vector Machines (SVMs) method implemented in [12] by the same authors of the `wustl-ehms-2020` dataset. The result obtained by SVM is the same present in [12] (we did not reproduce the experiments with this method because the dataset was the same). To facilitate the reproduction of our experiments, all the software developed is publicly available[1].

A. Experimental Setup

The experiments were run in Google Colaboratory with the basic configuration (12GB of RAM, 2 CPU Cores, and 2.30GHz CPU Freq). For loading and manipulating the datasets, the Pandas framework was used.

[1] https://github.com/fatemehm/Federated-Learning-IDS

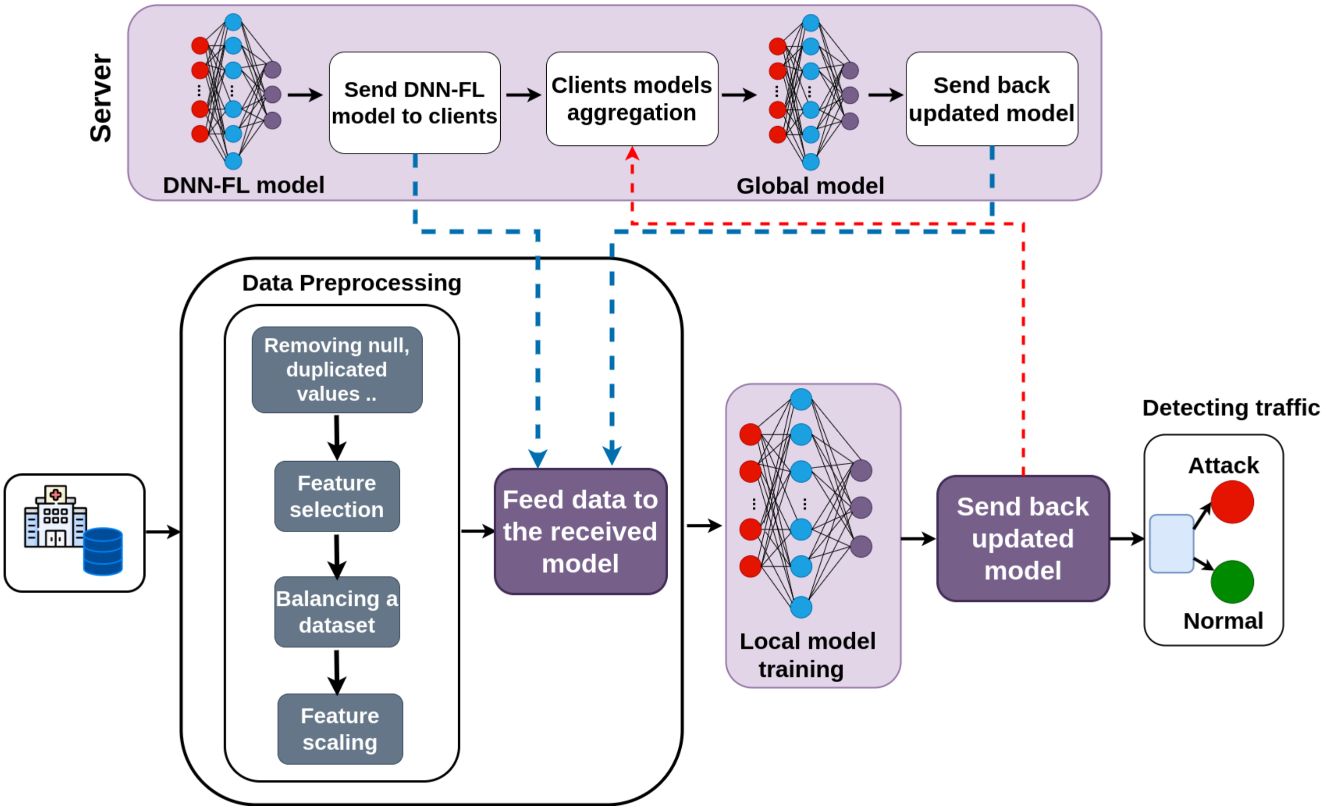

Fig. 1. Architecture of our DNN-FL for anomaly detection.

B. Datasets

The initial phase of our performance evaluation involved processing network traffic generated by the IoHT devices. In this research, we used two publicly available IoHT datasets, each one having different features:

1) wustl-ehms-2020: Hady et al [12] created the `wustl-ehms-2020` dataset from a specialized testbed. The testbed includes a multi-sensor board to measure the biometric data of the patient's body. This medical board consists of four sensors: 1) An electrocardiogram sensor able to collect the electrical signals of the heart; 2) An oxygen saturation level estimator sensor that can measure the level of oxygen in the blood; 3) A sensor to measure the body temperature; and 4) A sensor to collect the patient's systolic and diastolic pressure.

A computer is connected to this multi-sensor board to transfer data to a server, which saves and analyzes the received data to make a medical decision. Another computer in the network is used to emulate attacks. The `wustl-ehms-2020` dataset has 2046 attack instances and 14272 normal instances. It contains 36 features (both network and biometric data). The dataset has two attack categories: spoofing attacks and data alteration. The attacker tries to get the packets, spoofs/alter them, and redirects to the server.

2) ECU-IoHT: Ahmed et al [13] created the `ECU-IoHT` dataset. The environment to generate the dataset consists of a Windows 10, Kali Linux, a mobile Wi-Fi hotspot, a wireless network adapter, and a Bluetooth adapter to connect the hosts to the Internet. The environment also contains a healthcare kit called MySignals. The kit has several components and multiple sensors to monitor and store patients' body biometric data (temperature, blood pressure, and heart rate) and send it to the users' cloud. The dataset contains 7 features related to network data: source, destination, protocol, and type of attacks.

The `ECU-IoHT` dataset contains 87754 normal instances and 23453 attack instances. Four types of attacks were created by attacking devices: ARP spoofing, DoS attack, nmap port scan, and smurf attack.

C. Data Preprocessing

We converted the raw data into a CSV file, then divided the total data into five, ten, and fifteen clients to determine how increasing the number of clients affected the model's accuracy. Afterward, the NaN values and the duplicate instances were removed. The Label column, in the datasets, is encoded as 0 for normal and 1 for attack network instances. Due to an imbalance in the datasets, we used an undersampling method for balancing them. This balancing technique decreases the majority classes and avoids overfitting the model [22]. Finally, we used the StandardScaler() function for feature scaling to standardize the dataset into 0 to 1.

D. Results for the first experiment

As seen in Table III, our proposed DNN-FL with five, ten, and fifteen participants were trained. The table shows

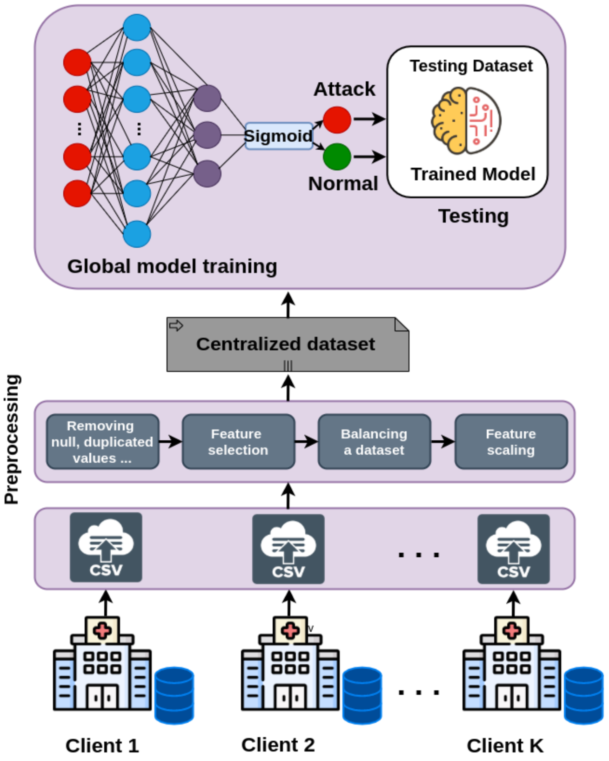

Fig. 2. Architecture of the centralized systems used as baselines to compare the performance of our proposal.

the results of all the models for the `wustl-ehms-2020` dataset. The DNN-FL method, with ten participants, achieved the highest accuracy in classification performance (91.40%) when compared with DNN, LSTM, and CNN-LSTM. In DNN-FL, with ten participants, the loss is equal to 0.25, which shows the absolute difference between our model attack detection rate and the actual value. The training times for DNN, CNN-LSTM, and LSTM are 0.039, 0.043, and 0.045 seconds, respectively, while the training time for DNN-FL, with five participants, is around 0.037 seconds. Moreover, the testing time, in our proposed FL model is 0.014 seconds which is similar to the centralized models.

TABLE III
RESULTS OBTAINED FOR `WUSTL-EHMS-2020` (ACC=ACCURACY, PRE=PRECISION, REC=RECALL)

Algorithm	Acc (%)	Pre (%)	Rec (%)	Train time (sec)	Test time (sec)	Loss
DNN-FL (15 clients)	90.64	70.39	49.76	0.033	0.011	0.35
DNN-FL (10 clients)	91.40	65.05	61.42	0.045	0.014	0.25
DNN-FL (5 clients)	90.70	56.40	76.63	0.037	0.014	0.35
DNN	90.52	93.00	88.00	0.039	0.014	0.23
LSTM	76.69	79.00	73.00	0.045	0.015	0.47
CNN-LSTM	89.08	86.00	93.00	0.043	0.016	0.25
SVM [12]	92.40	—	—	0.21	0.05	—

We also compared our proposal with the SVM model proposed by the authors of the `wustl-ehms-2020` dataset. With this comparison, we note that the SVM obtained the best accuracy (92.40% against 91.40% of our proposal). Nonetheless, the training time of the DNN-FL, with ten participants, was only 0.045 seconds, against 0.21 seconds of the SVM. Furthermore, the testing time of our DNN-FL proposal was only 0.014 seconds against 0.05 seconds of the SVM. So, despite having an accuracy 1.08% worst than that of the centralized SVM model, our decentralized DNN-FL model run faster, both on training (4.67 times faster) and testing (3.57 times faster), in addition to increase the privacy.

TABLE IV
RESULTS OBTAINED FOR `ECU-IoHT` (ACC=ACCURACY, PRE=PRECISION, REC=RECALL)

Algorithm	Acc (%)	Pre (%)	Rec (%)	Train time (sec)	Test time (sec)	Loss
DNN-FL (15 clients)	95.03	93.97	81.47	0.055	0.040	0.23
DNN-FL (10 clients)	98.47	93.88	99.18	0.045	0.014	0.11
DNN-FL (5 clients)	94.17	82.64	90.37	0.037	0.014	0.14
DNN	94.56	90.00	100	0.028	0.030	0.17
LSTM	94.48	90.00	100	0.043	0.016	0.15
CNN-LSTM	94.72	91.00	100	0.042	0.015	0.27

E. Results for the second experiment

Table IV shows the results obtained by all the models for the `ECU-IoHT` dataset. The DNN-FL model, with ten participants, achieved the highest accuracy in classification performance (98.47%). Its precision, loss, and testing time were better than the three centralized models. The best recalls were obtained by the centralized models, but all the other metrics obtained by our DNN-FL model are better or similar, besides the fact that the privacy is increased with our DNN-FL model.

Our proposed metrics are not present in the work where the `ECU-IoHT` was introduced [13]. Due to this reason, we could not compare our work with this related work, and it is not present in Table IV.

V. Conclusions and Future Works

Due to regulations and policies, hospitals and organizations must avoid sharing patient data when running IDSs to detect cyberattacks in IoHT traffic. In this scenario, federated learning is a solution to increase privacy without losing much accuracy in the detection process.

This paper advances the state of the art by improving the privacy-preserving DNN-FL as the base of an IDS for the IoHT environment using specific IoHT traffic datasets in order to enable a decentralized IDS. With our proposed FL method, organizations such as hospitals do not need to share their training data, which can ensure privacy-preserving for each patient. Compared with centralized anomaly detection learning methods, the DNN-FL achieved the equivalent or best performance, with an accuracy of 91.40% for the `wustl-ehms-2020` dataset and an accuracy of 98.47% for the `ECU-IoHT` dataset. Moreover, on average, the training and testing times of DNN-FL were always better or similar than all centralized models, such as SVM technique. In future works, we will improve our model by implementing CNN in FL and using larger datasets with different feature spaces.

Acknowledgments

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 FAPESP proc. 15/24485-9.

References

[1] Statista, “Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025,” 2016, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ Accessed at May 22nd, 2022.

[2] F. Farhin, M. S. Kaiser, and M. Mahmud, “Towards Secured Service Provisioning for the Internet of Healthcare Things,” in *Proceedings of the IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, 2020, pp. 1–6.

[3] A. S. Jat and T.-M. Grønli, “Blockchain for Cybersecure Healthcare,” in *International Conference on Mobile Web and Intelligent Information Systems*, 2022, pp. 106–117.

[4] K. Sadiq, A. Thompson, and O. Ayeni, “Toward Healthcare Data Availability and Security Using Fog-to-Cloud Networks,” in *Intelligent Interactive Multimedia Systems for e-Healthcare Applications*, 2022, pp. 81–103.

[5] F. Mosaiyebzadeh, L. G. Araujo Rodriguez, D. Macêdo Batista, and R. Hirata, “A Network Intrusion Detection System using Deep Learning against MQTT Attacks in IoT,” in *Proceedings of the IEEE Latin-American Conference on Communications (LATINCOM)*, 2021, pp. 1–6.

[6] J. P. Albrecht, “How the GDPR will Change the World,” *Eur. Data Prot. L. Rev.*, vol. 2, p. 287, 2016.

[7] M. Parasol, “The impact of China’s 2016 Cyber Security Law on Foreign Technology Firms, and on China’s Big Data and Smart City Dreams,” *Computer Law & Security Review*, vol. 34, no. 1, pp. 67–98, 2018.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[9] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated Optimization: Distributed Machine Learning for on-Device Intelligence,” *arXiv preprint arXiv:1610.02527*, 2016.

[10] Y. Xu, M. Z. A. Bhuiyan, T. Wang, X. Zhou, and A. K. Singh, “C-FDRL: Context-Aware Privacy-Preserving Offloading Through Federated Deep Reinforcement Learning in Cloud-Enabled IoT,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1155–1164, 2022.

[11] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, “An Adaptive Federated Learning Scheme with Differential Privacy Preserving,” *Future Generation Computer Systems*, vol. 127, pp. 362–372, 2022.

[12] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, “Intrusion Detection System for Healthcare Systems using Medical and Network Data: A Comparison Study,” *IEEE Access*, vol. 8, pp. 106 576–106 584, 2020.

[13] M. Ahmed, S. Byreddy, A. Nutakki, L. F. Sikos, and P. Haskell-Dowland, “ECU-IoHT: A Dataset for Analyzing Cyberattacks in Internet of Health Things,” *Ad Hoc Networks*, vol. 122, p. 102621, 2021.

[14] M. Masud, G. S. Gaba, K. Choudhary, M. S. Hossain, M. F. Alhamid, and G. Muhammad, “Lightweight and Anonymity-Preserving User Authentication Scheme for IoT-based Healthcare,” *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2649–2656, 2022.

[15] M. U. Abbasi, M. Kamal, and M. Tariq, “Improved and Secured Electromyography in the Internet of Health Things,” *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2032–2040, 2021.

[16] O. Ibitoye, O. Shafiq, and A. Matrawy, “Analyzing Adversarial Attacks against Deep Learning for Intrusion Detection in IoT Networks,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[17] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, “A Hierarchical Hybrid Intrusion Detection Approach in IoT Scenarios,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2020, pp. 1–7.

[18] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, “Federated Learning for Malware Detection in IoT Devices,” *Computer Networks*, vol. 204, p. 108693, 2022.

[19] C. Braghin, S. Cimato, and A. Della Libera, “Are mHealth Apps Secure? A Case Study,” in *Proceedings of the IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 02, 2018, pp. 335–340.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn: Machine Learning in Python,” *the Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] K. Bonawitz, H. Eichner, W. Grieskamp *et al.*, “TensorFlow Federated: Machine Learning on Decentralized Data,” https://www.tensorflow.org/federated?hl=en. Accessed at Mar 6th, 2023.

[22] R. Mohammed, J. Rawashdeh, and M. Abdullah, “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results,” in *Proceedings of the 11th International Conference on Information and Communication Systems (ICICS)*, 2020, pp. 243–248.