# Privacy-Preserving IoT Intrusion Detection: Challenges and Solutions in Implementing the CSAI-4-CPS Model

Hebert Silva [1] [2] and Regina Moraes [1] [3]

[1]Universidade Estadual de Campinas, Limeira, Brazil,

[2]National Industrial Training Service, SENAI, Sao Paulo, Brazil,

[3]University of Coimbra, CISUC, DEI, Coimbra, Portugal,

**Abstract.** The CSAI-4-CPS model leverages federated learning to collaboratively train machine learning models, providing accurate and up-to-date results while preserving data privacy. This approach is particularly beneficial in complex and dynamic Cyber-Physical Systems (CPS) environments where traditional centralized machine learning models may fall short. This paper presents the first validation of the CSAI-4-CPS model using a framework implemented for an IoT system and describes the new features of its expanded version. Real-time threat detection, consideration of false positives, and verification and validation of results on nodes that benefit from federated learning are among these new capabilities. It also compares the results obtained with and without the model. IoT systems often represent the most challenging scenarios in CPS cybersecurity, and in most cases, IoT devices are part of a more complex CPS structure, where they are usually the most vulnerable assets. The application of CSAI-4-CPS to predict malicious traffic in Internet of Things (IoT) networks appears promising. The results demonstrate that the model effectively detects intrusions in these datasets. By employing federated learning and a self-adaptive architecture, the model maintains its accuracy and relevance as new data emerges.

**Keywords:** Federation Learning, Data Privacy, Cybersecurity, CPS, IoT

## 1 Introduction

The increasing adoption of IoT (Internet of Things) devices in modern society has raised concerns about the security of these devices and the confidentiality of the data they generate. The security of these devices can be improved by employing machine learning models designed for intrusion detection, capable of identifying and thwarting cyberattacks. However, applying machine learning models to IoT devices presents several challenges.

In addition, the centralized solutions for handling the large amounts of data generated by these devices, that prevent malicious actors and cyberattacks (for example, Intrusion Prevention Systems - IPS and Intrusion Detection Systems - IDS)

David C. Wyld et al. (Eds): SEC, NLPML, CYBI, SIGL, CRIS, AIAPP, CoSIT- 2024
pp. 107-125, 2024.CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140709

used to protect IoT devices, can do not address adequately the privacy concerns. Especially when using a traditional approach that involves sharing data from the nodes to detect and respond.

The use of sensitive IoT data also raises significant privacy concerns because this data often contains sensitive information that must be protected. To overcome these challenges, security solutions for IoT often combine edge computing (to process data closer to the source), federated learning (for privacy-preserving collaborative model updates), and lightweight cryptographic techniques to ensure both security and privacy without overburdening the IoT devices.

In this context, federated learning emerges as a promising solution to protect IoT devices against cyberattacks while preserving data privacy. Federated learning is a distributed machine learning training method in which the data remains on user devices, preserving data privacy, while the model is trained collaboratively [28]. This approach enables the data to remain on the devices where it was generated while the model is trained by all participants.

This task includes ensuring the quality and improving the accuracy of the training data, managing the communication between the central server and the user devices, and dealing with the potential for data imbalance across user devices.

While federated learning offers a promising solution for preserving data privacy in network IoT intrusion detection systems, there are still several challenges that must be overcome to ensure its effective implementation. This is where machine learning, and more specifically, federated learning, can play a significant role. Federated learning can be particularly useful for intrusion detection in scenarios where privacy is paramount, such as in healthcare or industrial IoT. It is also valuable in scenarios where IoT devices have limited connectivity to central servers, or where regulatory constraints limit data sharing across borders or between organizations [8]. In summary federated learning enhances the capabilities of IDSs by enabling collaborative, privacy-preserving, and scalable machine learning across distributed IoT devices, which is critical for the dynamic and diverse nature of IoT security threats. Integrating Intrusion Detection Systems with Federated Learning aims to preserve user privacy, enhance anomaly detection accuracy, minimize network bandwidth usage, ensure system scalability, maintain robustness against uneven data distribution, and facilitate ongoing adaptive learning in the ever-evolving cybersecurity landscape.

Feature selection and an adequate pre-process in each dataset node are essential steps in building local machine learning models that can effectively detect cyberattacks on IoT devices. These tasks involve identifying the most relevant data features that are most predictive of the target variable while discarding irrelevant or redundant ones that may introduce noise and reduce model accuracy.

Selecting features for machine learning models remains a challenging task, particularly in the context of IoT devices, where data is often unstructured and com-

plex. Therefore, developing effective feature selection methods that can be combined with federated learning is critical for the development of accurate and privacy-preserving intrusion detection models. Therefore, addressing these challenges is fundamental to the development of effective and privacy-preserving IoT network intrusion detector systems. These systems are becoming increasingly important as IoT devices continue to be integrated into various aspects of our daily lives.

This paper expands and validate a solutions for achieving privacy-preserving IoT intrusion detection using the CSAI-4-CPS model characterize in [10]. The work emphasizes the importance of federated learning in training machine learning models collaboratively, enabling accurate and up-to-date results while safeguarding data privacy. In addition to presenting and describing the CSAI-4-CPS model, the paper presents a validation of the proposed method by implementing a framework based on the model and highlighting key findings through case studies using realistic data.

The remainder of this work is divided as follows: Section 2 presents some background and the related work; the CSAI-4-CPS model expanded is presented in Section 3; Section 4 presents case studies carried out to validate the approach; the results and discussions can be found in Section 5; finally Section 6 brings our conclusions and future works.

## 2 Background and Related Work

This section outlines the concepts that underpin this work and resumes a literature review. Essentially, it involves subjects focused on cybersecurity, artificial intelligence, machine and federated learning, and data privacy.

### 2.1 Background

The concept of cybersecurity is frequently treated as synonymous with information security; however, cybersecurity extends beyond the traditional boundaries of information security. It encompasses not only the digital assets but also considers the users of the systems, focusing on their responsibilities and interaction with the systems [2]. Notably, cybersecurity is one of the fastest-growing industries today. The most common threats are the exploitation of vulnerabilities in end-user computers, IoT devices, web pages, and cryptocurrency technologies; the exposition of data made available to third parties; the management of security patches; and phishing attacks [3]. The CSAI-4-CPS model [10] addresses cybersecurity in cyber-physical systems, considering the protection of computational assets and the role of people involved in processes and system usage. It can provide relevant guidelines for implementing cybersecurity measures in cyber-physical systems. When we enter the topic of Artificial Intelligence (AI) in cybersecurity, we see that its adoption is synergistic and presents great opportunities. However, it brings several challenges to be

overcome. New attack methods will be postulated due to the advancement of technologies, using technologies that are often still in the state of the art or not trivial in the field of AI. These technologies increase the degree of sophistication of cyber-attacks, allowing them to be faster, better targeted, and more destructive [4]. It is increasingly common to use Botnets, for example, which are normally distributed logically and geographically. This makes it necessary to build more effective tools, raise user awareness, avoid cyber incident costs, improve technical knowledge, and seek market incentives [5].These attacks have evolved from simply obtaining access passwords, exploiting backdoors, and denial of service to the use of automated attacks using bots, botnets, and zombies, the use of force to crack hashes, and even cryptography exploitation. The proposed model recognizes the importance of adopting AI technologies to strengthen cybersecurity in cyber-physical systems. It can provide guidelines for addressing the challenges and developing effective tools, as well as promoting user awareness and enhancing technical knowledge in cybersecurity and AI.

The most diverse techniques and approaches can be used in cybersecurity to protect networks and systems. The work presented by SARKER et al. [6] considers the most popular AI techniques, which include machine learning (ML) and deep learning (DL) methods, natural language processing (NLP), Knowledge Representation and Reasoning (KRR), as well as the concept of knowledge- or rule-based Expert Systems (ES) modeling, to meet needs related to the use of AI in cybersecurity. According to the authors a security model for machine learning is usually characterized by a set of data related to security events, such as network behavior, databases, application activities, and users, among others. The authors emphasize that a major requirement in automating cybersecurity systems is the necessity to uncover patterns of security incidents or insights from cybersecurity data in order to construct data-driven models. Furthermore, FENG et al. [7] highlights that a better balance between AI, cybersecurity, and data protection in relation to legislative regulations such as GDPR is required [8]. Our model emphasizes the importance of utilizing AI techniques such as machine learning and deep learning in data analysis and decision-making in cyber-physical systems. It can provide relevant guidance for developing data-driven security models and finding a balance between AI, cybersecurity, and data protection, taking into account applicable regulations.

Another useful technique is feature selection, which enables the selection and utilization of higher quality data subsets, resulting in stronger learning models and increased prediction accuracy. SILVA et al. [11] sought to comprehend the impact of feature selection on the performance of supervised machine learning systems during the classification assignment. The work was able to verify, through experiments on CPS data, the increase in classification accuracy in machine learning when feature selection was employed in the pre-processing step. There are two main feature selection techniques: supervised and unsupervised. Supervised methods can be divided

into three types [11]: i) wrapper: subsets of features with good performance; ii) filter: use statistical measures to choose the most relevant attributes; and iii) intrinsic: automatically select features during training. The main type of unsupervised method is clustering, which partitions data into k groups. The CSAI-4-CPS model integrates the attribute selection method proposed by SILVA et al. [11], embedding it within the workflow to identify and retain only the most pertinent attributes before forwarding them to federated learning local node. This pre-selection process effectively reduces the volume of data that needs to be processed by the local models, easing the computational load and optimizing network traffic. Consequently, only the most relevant attributes are shared with the centralized server in the federated learning (FL) setup, thereby enhancing the overall efficiency and effectiveness of the system.

Aiming to preserve data privacy, federated learning was proposed, which is a distributed machine learning approach where several devices at the edge of the network work collaboratively to train a shared prediction model on top of data maintained locally by these devices [28], keeping sensitive data private [12] [13]. In this approach, devices at the edge of the network download the global model and perform a training step on it, generating a local model. After that, the models updated by all clients are sent again to the central server, which updates the global model of the server, and sent again to the edges of the network so that the steps can be repeated for each training round [12] [13]. The federated learning architecture can be defined as horizontal (the same attributes, but the data sources are different), vertical (similar samples, and it is necessary to intersect the spaces of the samples to test the model), or by transfer (participants only transfer knowledge from one model to another without exposing the used dataset).

### 2.2 Related Works

Some works propose different learning techniques to improve the algorithm's performance and capacity for detecting attacks. The work by JAHROMI et al. [14] present a method utilizing a stacked autoencoder in an unsupervised manner. This technique is designed to process original features and distill them into a more compact, reduced-dimensional representation. Besides, six classification algorithms were analyzed to verify the efficiency of the proposed technique. The results showed an improvement in the performance of all algorithms, with a special result for weak algorithms on unbalanced data.

In the research conducted by SIDDIQUE et al. [15], a novel approach is employed that merges lateralization with modular learning throughout varying stages of abstraction in a machine learning system designed for classifying images. The findings indicate that this system adeptly masters hierarchical knowledge structures and exhibits a performance that rivals or surpasses contemporary advanced deep learning systems, especially when it utilizes a variety of representations.

To train and create a robust intrusion detection model for IoT devices, SHAHID et al. [16] propose the use of federated learning techniques. Their approach suggests a decentralized and cooperative method for training machine learning models that complies with GDPR regulations by maintaining user data on the local IoT device. The approach evaluation uses three different use cases to show the security enhancements, resulting in a more reliable model. They also conclude that the way the dataset is distributed impacts performance. Our proposal also considers the Federated Learning (FL) technique, but it is complemented with feature selection (SF) to optimize the process. Moreover, our approach self-adapts the model, considering the false positives, to refine the model that returns to the server for retraining. Our goal is to propose a methodology that uses FL, SF, and end-user knowledge as possible techniques to classify broad types of intrusion.

The work published in ALVES et al. [17] presents MLPrivacyGuard, which implements a countermeasure against black-box model inversion attacks. This preventative strategy involves the introduction of deliberate noise into the output produced by a confidence function. The work corroborates with the present research when it highlights the importance of preserving the accuracy of the prediction or classification for the real users of the model, preventing intruders from deducing confidential data. This involves constantly seeking to balance the trade-off between classification error and effectiveness in the detention of attacks.

KANIMOZHI and JACOB [18] develop a system to detect a Botnet attack classification based on Artificial Neural Networks, presenting exceptional performance. The authors suggest that the proposed system is capable of being deployed in machinery for analyzing standard networks, monitoring traffic in real time, and managing Cyber-Physical Systems (CPS). Attack detection is also the focus of the study published in CAO et al. [19], which proposes agnostic attack detection in interactive recommendation systems based on reinforcement learning. The study demonstrates, via comprehensive experimentation, that the majority of adversarial attacks prove to be effective, with the intensity of the attacks and the rate at which they are carried out both influencing the overall effectiveness of the attacks.

HU, ZHU, and LIU [20] also employed reinforcement learning to design an adaptive approach based on Bayesian classifiers to improve the cost-effectiveness of attack identification and the interactions between the attacker and network modeling. Also, it formulates the advocacy problem as a partially observable Markov decision-process problem. Furthermore, it leverages Thompson sampling to estimate transition probabilities and uses reinforcement learning to choose optimal detention actions using measured utility values. The performance of the algorithm was verified through numerical simulations based on real-world attacks.

The work published in XU et al. [21] investigates the evolutionary process, which captures the natural interactions and evolutions of attackers and defenders, as well as their strategies to launch data integrity attacks and protect IoT systems,

respectively. The realistic IoT environment simulator (WCPS) was used, and the results indicate the effectiveness of defensive strategies in protecting against various forms of data integrity attacks with increasing time and knowledge throughout the evolutionary process.

It is noted that the works presented in this section meet specific demands and complex contexts in their niches. Therefore, they do not configure a framework that can guide a model or the construction of more generic solutions, as is the case of the CSAI-4-CPS proposal.

## 3 The CSAI-4-CPS Model

The CSAI-4-CPS model is designed based on two main parts, one that is responsible for the feature selection process and the other responsible for horizontal federated learning. The next subsections describe the model and give details about its implementation (the framework). Figure 1 exemplifies the model and the Figure 2 shows the framework.

### 3.1 The Model Description

The first part of the model is responsible for data preprocessing and feature selection. Three feature selection methods are used: filter, wrapper, and embedded. Each method should generate its own list of the best features. From these lists, the 10 main attributes of each of the methods are selected, which will be used for classification tests using the Random Forest algorithm, as suggested in the work of [11]. The output of this process will be the highest accuracy found and the corresponding features. In the case of a tie, the set with the lowest number of characteristics will be selected.

Horizontal federated learning is the focus of the second part of the model. A main loop is created that iterates four times, representing rounds of detecting and correcting data (false positives and false negatives) for each client. Each client iteration performs the following steps: i) extracts and stores its dataset locally; ii) makes predictions using the global model on the unlabeled data; iii) corrects misclassified data (false positives and negatives); iii) preprocesses the newly corrected dataset; and iv) combines it with existing datasets for further training of the global model and client models.

The training of the global model updates the model weights based on the corrected data from the clients, simulating the correction of false positives and negatives. The process is repeated multiple times, recording training accuracy metrics and checking if the current accuracy is higher than the previously recorded best accuracy. Training is stopped if accuracy does not increase after a specific number of rounds.

The items inside the light green box in the middle of Figure 1 represent the feature selection, machine learning, and federated learning in the local process. This part represents the pipeline of machine learning associated with feature selection, as explained before. This process involves the items 1 until 3. The horizontal federated learning starts in item 4.1, where the process begins with the initialization of a global model. This model is typically pre-trained on a large dataset or initialized randomly. Furthermore, the data is distributed across multiple client devices or nodes. Each client has its own local dataset, which may have variations, such as different data samples or distributions.

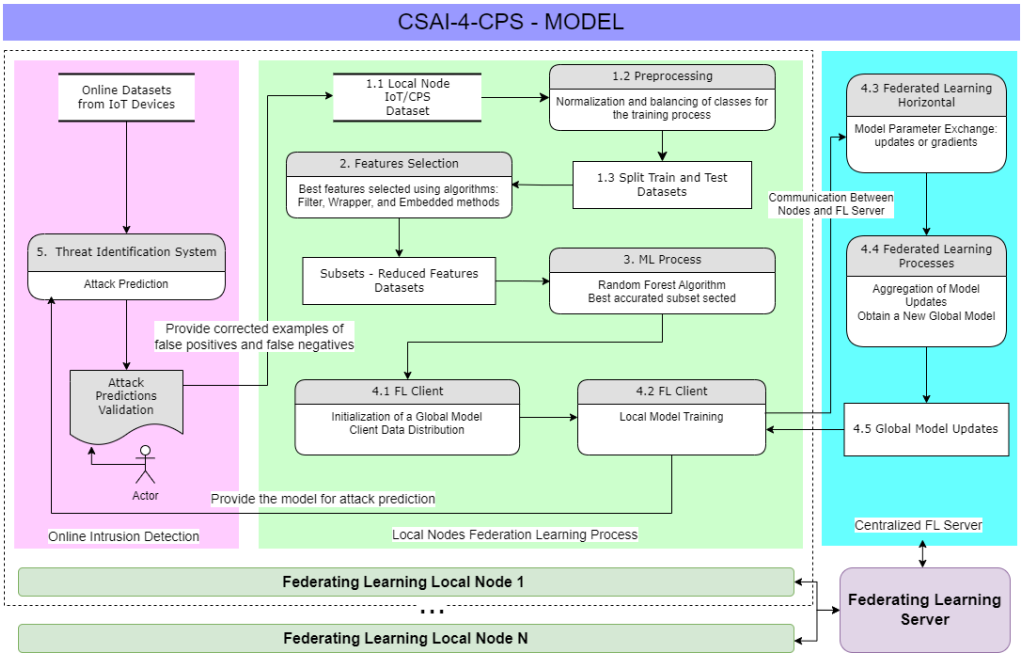


**Fig. 1.** CSAI-4-CPS Model for Intrusion Detection

The leftmost box of Figure 1 (the pink one), represents the CSAI-4-CPS model [10] that is expanded by this work to handle data collected and the constant monitoring, where the prediction of potential attacks takes place. The data may have been the target of physical attacks (defective devices) and/or cyberattacks, whether intentional or not. Part of this data is used to train the model locally to ensure data privacy by not sharing it with other nodes. A report is generated with the results of the prediction process plus the false positives and negatives indication. At this point, humans interact with the model to analyze if any action needs to be taken.

The rightmost box (the light blue one) is where the models of each participant are exported to a platform that compares the models received by its n participants

(nodes) and aggregates and generates a new global model using the techniques of federated learning. After characterizing the best model, it is returned to its n participants. This process is done in a feedback way, creating a virtuous cycle of self-adaptation of the learning models of all participating nodes for the identification of cyberattacks in the datasets. In the next round, all nodes will take advantage of the new version of the model, which was obtained in collaboration with all participants.

### 3.2 Framework Implementation Details

We use Colab [22] and Keras [24] because they are popular tools for implementing the framework, as both provide an accessible, user-friendly, and powerful platform for developing federated learning models. Colab, short for Google Colaboratory, is a free, cloud-based platform for developing and running machine learning models using Python. It offers a Jupyter notebook environment with access to Graphics Processing Units (GPU) and Tensor Processing Units (TPU) resources, making it a popular choice for data scientists, researchers, and machine learning practitioners [22] [23]. In addition, the users can access pre-built libraries and frameworks, such as TensorFlow, Keras, and PyTorch [22] [23]. The implementation framework code is available on the GitHub repository for the project CSAI-4-CPS [1].

With Colab, users can write and execute Python code in a web browser, without the need to install any software on their computer. Colab notebooks can be easily shared and facilitate collaboration with others. In addition, the users can access pre-built libraries and frameworks, such as TensorFlow, Keras, and PyTorch [22] [23].

Keras is a high-level open-source neural network library written in Python. It was developed as a user-friendly interface for building and training deep learning models, and it can run on top of various backends such as TensorFlow, Theano, and Microsoft Cognitive Toolkit [23]) [24]. Keras also offers a wide range of built-in functions and tools for data preprocessing, model evaluation, and visualization. It has a large and active community, which contributes to its ongoing development and improvement. Keras is widely used in various domains, including computer vision, natural language processing, and speech recognition, among others [24].

Keras provides a simple and intuitive interface for constructing neural networks, allowing users to quickly build and train models without requiring in-depth knowledge of the underlying algorithms. It supports various types of neural networks, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep neural networks (DNNs). Keras also offers a wide range of built-in functions and tools for data preprocessing, model evaluation, and visualization. It has a large and active community, which contributes to its ongoing development and improvement. Keras is widely used in various domains, including computer vision, natural language processing, and speech recognition, among others [24].

[1] https://github.com/hebertos/CSAI-4-CPS

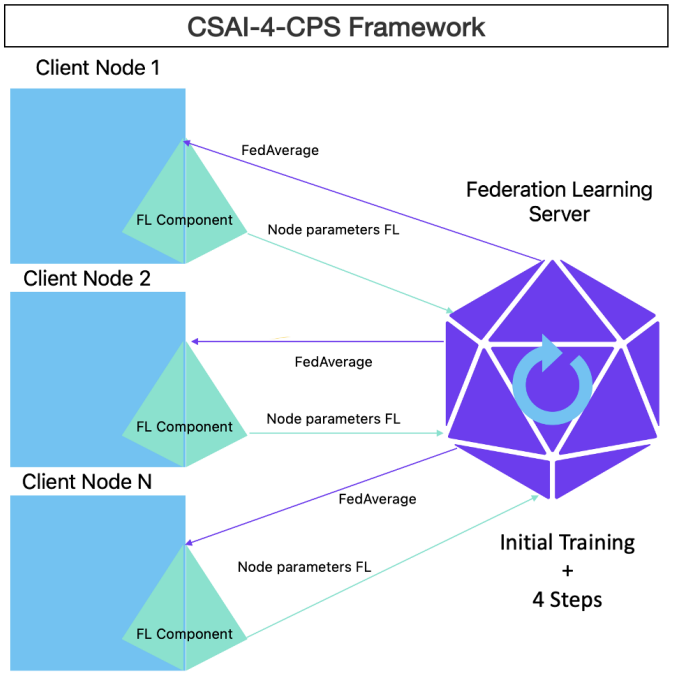


**Fig. 2.** CSAI-4-CPS Framework

The implementation of the framework in this work also utilizes several libraries such as NumPy, Pandas, TensorFlow, Scikit-learn, and Matplotlib, among others. These libraries provide useful functionalities for data preprocessing, feature selection, and classification model training. The main objective is to simulate the execution of the CSAI-4-CPS model in a usable framework including four client nodes. In this setup, the approach bases the concept of the best parameters calculated by horizontal federated learning with the Fed Average algorithm.

In the first part, (see Figure 2 Nodes and FL component) the implemented framework performs data preprocessing and feature selection. Three feature selection methods are used: filter, wrapper, and embedded. Each method generates your best features list. From these lists, the top 10 attributes from each method are selected, and classification tests are performed using the Random Forest algorithm, as suggested in the work by [11], and that for the data tested they proved to be effective. The output of this process is the highest accuracy found and the corresponding features. In case of a tie, the set with the smallest number of features is selected. The implementation of horizontal federated learning uses the TensorFlow Federated (TFF) library to perform the Federation Average algorithm.

Thus, a main loop is created that iterates four times, representing rounds of detecting and correcting data (false positives and false negatives) for each client.

Each client iteration performs the following steps: i) extracts and stores its dataset locally; ii) makes predictions using the global model on the unlabeled data, iii) corrects misclassified data (false positives/negatives); iii) preprocesses the newly corrected dataset; and iv) combines it with existing datasets for further training of the global model and client models.

The training of the global model updates the model weights based on the corrected data from the clients, simulating the correction of false positives/negatives. The process is repeated multiple times, recording training metrics such as accuracy and checking if the current accuracy is higher than the previously recorded best accuracy. After the initial training round, up to four rounds is established if the accuracy does not increase. This complete process can be seen in Figure 2, and the code can be verified at CSAI-4-CPS project Github.

## 4 Case Study

This section describes the experiments carried out as a model proof of concept using the proposed framework. Firstly, it presents the used datasets and the cyberattacks implemented, followed by the two test scenarios.

### 4.1 The Datasets

The datasets capture network traffic from a realistic IoT network in a controlled lab environment provided by database N-BaIoT [26]. It includes several IoT devices and provides information such as IP addresses, protocols, and payloads. The datasets enable analysis of security intrusion attempts (like DDoS attacks) and can be used to develop intrusion detection and prevention systems for IoT devices [26]. The database N-BaIoT is composed of eight datasets with a total of 7,062,606 instances and 115 attributes. It possesses both multivariate and sequential characteristics.

The feature headers in the N-BaIoT dataset (Stream Aggregation, Time-frame and Statistics Extracted from the Packet Stream) describe different aspects of the packet stream and its statistics. Here's a breakdown of each feature header [27]:

Stream Aggregation:

- H: Statistics of the traffic from the packet's host (IP);
- HH: The same statistics plus the destination host;
- HpHp: The same statistics plus both IP addresses and port numbers;
- HH_jit: Statistics of the jitter (variation in packet arrival times) of the traffic from the packet's host (IP) to the destination host.

Time-frame (Decay Factor Lambda):

- L1, L3, L5, and Ln: Represent different time frames used in the statistics calculation. They determine how much the stream recent history is captured in these

statistics. Smaller values indicate a shorter time frame with more emphasis on recent events, while larger values indicate a longer time frame with a broader historical perspective.

Statistics extracted from the packet stream:

- weight: The number of items observed in recent history;
- mean: The stream average value;
- std: The stream standard deviation;
- radius: The square root of the stream sum of the variances. It represents the overall spread of the data;
- magnitude: The square root of the stream sum of the means. It represents the data overall magnitude (or scale);
- cov: Approximated covariance between two streams (degree of correlation or relationship);
- pcc: Approximated covariance between two streams (Pearson correlation coefficient - the linear relationship between the streams).

The N-BaIoT database includes different feature headers that describe various aspects of the packet stream and its statistics. These feature headers can be categorized into three groups: Stream Aggregation, Time-frame (Decay Factor Lambda), and Statistics extracted from the packet stream. The Stream Aggregation headers, such as H, HH, HpHp, and HH_jit, provide statistics related to the traffic from the packet's host and destination host, including IP addresses and port numbers. Additionally, HH_jit focuses on the jitter, which represents the variation in packet arrival times. The Time-frame headers, represented by L1, L3, L5, and Ln, determine the time frames used in calculating the statistics. They capture different portions of the stream's recent history, with smaller values indicating a shorter time frame and larger values indicating a longer time frame. The Statistics extracted from the packet stream include various measures, like Weight, Mean, Std Variation, Radius, Magnitude, COV and PCC. These feature headers provide insights into the characteristics and statistical properties of the packet stream, allowing for analysis and modeling of the network traffic in the N_BaIoT dataset. Considering the scenario of horizontal federated learning, the datasets were separated by devices. According to the tree graph in Figure 3, these device-specific datasets were divided equally among the nodes. Within each node, the data was further divided into sets for training, testing, and simulating online detection of new attacks.

### 4.2 The Cyberattacks

Gafgyt and Mirai [25] are two well-known botnets that have been responsible for several high-profile cyberattacks. Both botnets are known for their ability to infect and control vulnerable Internet of Things (IoT) devices, such as routers, cameras,

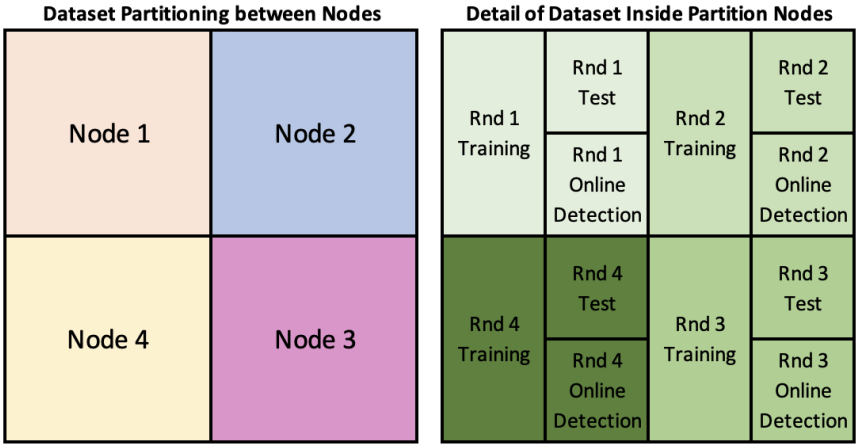


**Fig. 3.** Distribution of dataset usage in an experimental setup

*Table 1: Federation Learning Results*

A)

| DANMINI GAFGYT ATTACKS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| #Records | Step | Node 1 | | Node 2 | | Node 3 | | Node 4 | |
| | | ACC | RT | ACC | RT | ACC | RT | ACC | RT |
| 1831 | Init. Trng | 65.68 | 1 | 86.77 | 1 | 86.31 | 1 | 86.33 | 2 |
| 2198 | Rnd1 | 86.62 | 2 | 86.57 | 2 | 86.43 | 2 | 86.49 | 1 |
| 2565 | Rnd2 | 81.98 | 2 | 86.56 | 3 | 86.50 | 3 | 86.44 | 3 |
| 2932 | Rnd3 | 86.51 | 2 | 86.47 | 2 | 86.50 | 2 | 86.39 | 3 |
| 3299 | Rnd4 | 86.53 | 2 | 86.45 | 2 | 86.48 | 4 | 85.59 | 4 |

B)

| DANMINI GAFGYT ATTACKS | | | |
|---|---|---|---|
| #Records | Step | Sgl Node | |
| | | ACC | RT |
| 7324 | Init. Trng | 83.53 | 3 |
| 8792 | Rnd1 | 83.52 | 15 |
| 10260 | Rnd2 | 86.47 | 19 |
| 11728 | Rnd3 | 86.46 | 20 |
| 13196 | Rnd4 | 86.46 | 25 |

C)

| RT SF | |
|---|---|
| Node 01 | 29 |
| Node 02 | 20 |
| Node 03 | 19 |
| Node 04 | 17 |

Step RT = RT SF + Node RT

*Table 2: No Federation Learning Results*

A)

| DANMINI GAFGYT ATTACKS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| #Records | Step | Node 1 | | Node 2 | | Node 3 | | Node 4 | |
| | | ACC | RT | ACC | RT | ACC | RT | ACC | RT |
| 1831 | Init. Trng | 65.68 | 3 | 86.77 | 1 | 86.31 | 1 | 86.33 | 1 |
| 2198 | Rnd1 | 86.62 | 1 | 86.57 | 1 | 86.43 | 1 | 86.49 | 1 |
| 2565 | Rnd2 | 82.48 | 1 | 86.56 | 1 | 86.50 | 2 | 86.44 | 1 |
| 2932 | Rnd3 | 86.33 | 2 | 86.47 | 1 | 86.50 | 1 | 86.39 | 1 |
| 3299 | Rnd4 | 45.75 | 1 | 45.75 | 2 | 86.48 | 2 | 85.60 | 2 |

B)

| DANMINI GAFGYT ATTACKS | | | |
|---|---|---|---|
| #Records | Step | Sgl Node | |
| | | ACC | RT |
| 7324 | Init. Trng | 81.27 | 3 |
| 8792 | Rnd1 | 86.53 | 7 |
| 10260 | Rnd2 | 81.00 | 6 |
| 11728 | Rnd3 | 81.63 | 8 |
| 13196 | Rnd4 | 86.22 | 9 |

C)

| RT SF | |
|---|---|
| Node 01 | 22 |
| Node 02 | 13 |
| Node 03 | 10 |
| Node 04 | 8 |

Step RT = RT SF + Node RT

**Fig. 4.** Results for Dataset Danmini Gafgyt Attacks

and DVRs. It is common to use them to launch massive Distributed Denial of Service (DDoS) attacks. Gafgyt is typically spread through malware that targets IoT devices with weak or default login credentials, allowing the botnet to gain control of the device and use it to launch attacks [25]. Mirai is similar to Gafgyt in that it targets vulnerable IoT devices, but it also includes a worm component that enables it to spread rapidly across devices [25]. All datasets used in this work register previous attacks using one of these botnets.

The Gafgyt and Mirai botnets highlight the vulnerabilities of IoT devices and the importance of their protecton against cyberattacks. It is important to ensure that IoT devices implement strong passwords, up-to-date firmware, and security measures. Additionally, network traffic analysis and machine learning-based intru-

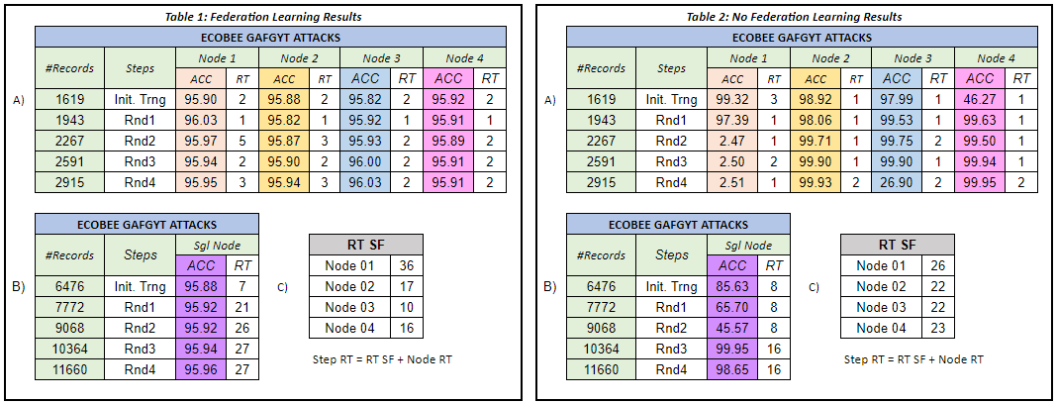

*Table 1: Federation Learning Results*

ECOBEE GAFGYT ATTACKS

A)

| #Records | Steps | Node 1 | | Node 2 | | Node 3 | | Node 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | ACC | RT | ACC | RT | ACC | RT | ACC | RT |
| 1619 | Init. Trng | 95.90 | 2 | 95.88 | 2 | 95.82 | 2 | 95.92 | 2 |
| 1943 | Rnd1 | 96.03 | 1 | 95.82 | 1 | 95.92 | 1 | 95.91 | 1 |
| 2267 | Rnd2 | 95.97 | 5 | 95.87 | 3 | 95.93 | 2 | 95.89 | 2 |
| 2591 | Rnd3 | 95.94 | 2 | 95.90 | 2 | 96.00 | 2 | 95.91 | 2 |
| 2915 | Rnd4 | 95.95 | 3 | 95.94 | 3 | 96.03 | 2 | 95.91 | 2 |

ECOBEE GAFGYT ATTACKS

B)

| #Records | Steps | Sgl Node | |
|---|---|---|---|
| | | ACC | RT |
| 6476 | Init. Trng | 95.88 | 7 |
| 7772 | Rnd1 | 95.92 | 21 |
| 9068 | Rnd2 | 95.92 | 26 |
| 10364 | Rnd3 | 95.94 | 27 |
| 11660 | Rnd4 | 95.96 | 27 |

C)

| RT SF | |
|---|---|
| Node 01 | 36 |
| Node 02 | 17 |
| Node 03 | 10 |
| Node 04 | 16 |

Step RT = RT SF + Node RT

*Table 2: No Federation Learning Results*

ECOBEE GAFGYT ATTACKS

A)

| #Records | Steps | Node 1 | | Node 2 | | Node 3 | | Node 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | ACC | RT | ACC | RT | ACC | RT | ACC | RT |
| 1619 | Init. Trng | 99.32 | 3 | 98.92 | 1 | 97.99 | 1 | 46.27 | 1 |
| 1943 | Rnd1 | 97.39 | 1 | 98.06 | 1 | 99.53 | 1 | 99.63 | 1 |
| 2267 | Rnd2 | 2.47 | 1 | 99.71 | 1 | 99.75 | 2 | 99.50 | 1 |
| 2591 | Rnd3 | 2.50 | 2 | 99.90 | 1 | 99.90 | 1 | 99.94 | 1 |
| 2915 | Rnd4 | 2.51 | 1 | 99.93 | 2 | 26.90 | 2 | 99.95 | 2 |

ECOBEE GAFGYT ATTACKS

B)

| #Records | Steps | Sgl Node | |
|---|---|---|---|
| | | ACC | RT |
| 6476 | Init. Trng | 85.63 | 8 |
| 7772 | Rnd1 | 65.70 | 8 |
| 9068 | Rnd2 | 45.57 | 8 |
| 10364 | Rnd3 | 99.95 | 16 |
| 11660 | Rnd4 | 98.65 | 16 |

C)

| RT SF | |
|---|---|
| Node 01 | 26 |
| Node 02 | 22 |
| Node 03 | 22 |
| Node 04 | 23 |

Step RT = RT SF + Node RT

**Fig. 5.** Results for Dataset Ecobee Mirai Attacks

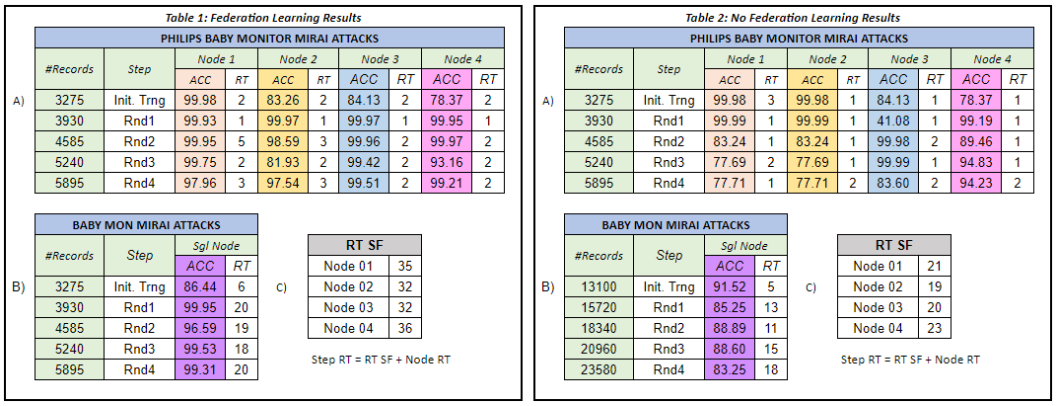

*Table 1: Federation Learning Results*

PHILIPS BABY MONITOR MIRAI ATTACKS

A)

| #Records | Step | Node 1 | | Node 2 | | Node 3 | | Node 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | ACC | RT | ACC | RT | ACC | RT | ACC | RT |
| 3275 | Init. Trng | 99.98 | 2 | 83.26 | 2 | 84.13 | 2 | 78.37 | 2 |
| 3930 | Rnd1 | 99.93 | 1 | 99.97 | 1 | 99.97 | 1 | 99.95 | 1 |
| 4585 | Rnd2 | 99.95 | 5 | 98.59 | 3 | 99.96 | 2 | 99.97 | 2 |
| 5240 | Rnd3 | 99.75 | 2 | 81.93 | 2 | 99.42 | 2 | 93.16 | 2 |
| 5895 | Rnd4 | 97.96 | 3 | 97.54 | 3 | 99.51 | 2 | 99.21 | 2 |

BABY MON MIRAI ATTACKS

B)

| #Records | Step | Sgl Node | |
|---|---|---|---|
| | | ACC | RT |
| 3275 | Init. Trng | 86.44 | 6 |
| 3930 | Rnd1 | 99.95 | 20 |
| 4585 | Rnd2 | 96.59 | 19 |
| 5240 | Rnd3 | 99.53 | 18 |
| 5895 | Rnd4 | 99.31 | 20 |

C)

| RT SF | |
|---|---|
| Node 01 | 35 |
| Node 02 | 32 |
| Node 03 | 32 |
| Node 04 | 36 |

Step RT = RT SF + Node RT

*Table 2: No Federation Learning Results*

PHILIPS BABY MONITOR MIRAI ATTACKS

A)

| #Records | Step | Node 1 | | Node 2 | | Node 3 | | Node 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | ACC | RT | ACC | RT | ACC | RT | ACC | RT |
| 3275 | Init. Trng | 99.98 | 3 | 99.98 | 1 | 84.13 | 1 | 78.37 | 1 |
| 3930 | Rnd1 | 99.99 | 1 | 99.99 | 1 | 41.08 | 1 | 99.19 | 1 |
| 4585 | Rnd2 | 83.24 | 1 | 83.24 | 1 | 99.98 | 2 | 89.46 | 1 |
| 5240 | Rnd3 | 77.69 | 2 | 77.69 | 1 | 99.99 | 1 | 94.83 | 1 |
| 5895 | Rnd4 | 77.71 | 1 | 77.71 | 2 | 83.60 | 2 | 94.23 | 2 |

BABY MON MIRAI ATTACKS

B)

| #Records | Step | Sgl Node | |
|---|---|---|---|
| | | ACC | RT |
| 13100 | Init. Trng | 91.52 | 5 |
| 15720 | Rnd1 | 85.25 | 13 |
| 18340 | Rnd2 | 88.89 | 11 |
| 20960 | Rnd3 | 88.60 | 15 |
| 23580 | Rnd4 | 83.25 | 18 |

C)

| RT SF | |
|---|---|
| Node 01 | 21 |
| Node 02 | 19 |
| Node 03 | 20 |
| Node 04 | 23 |

Step RT = RT SF + Node RT

**Fig. 6.** Results for Dataset Philips Baby Monitor Gafgyt Attacks

sion detection systems can help to detect and prevent attacks from botnets like Gafgyt and Mirai [25].

### 4.3 First Scenario Simulation

To execute the case study, the N_BaIoT dataset was divided into nine separate datasets based on equipment type. This division was done to simulate horizontal federated learning, where data originating from the same context is tested across equipment of the same brand and model. This approach allows for a more accurate and realistic evaluation of machine learning models in a federated environment. Afterward, each dataset was further divided into four equal parts to simulate data from four clients participating in the collaboration. This division ensures that the data from each client is represented in the training and evaluation process, allowing for a simulation of a collaborative setting. In each client, the data was further split

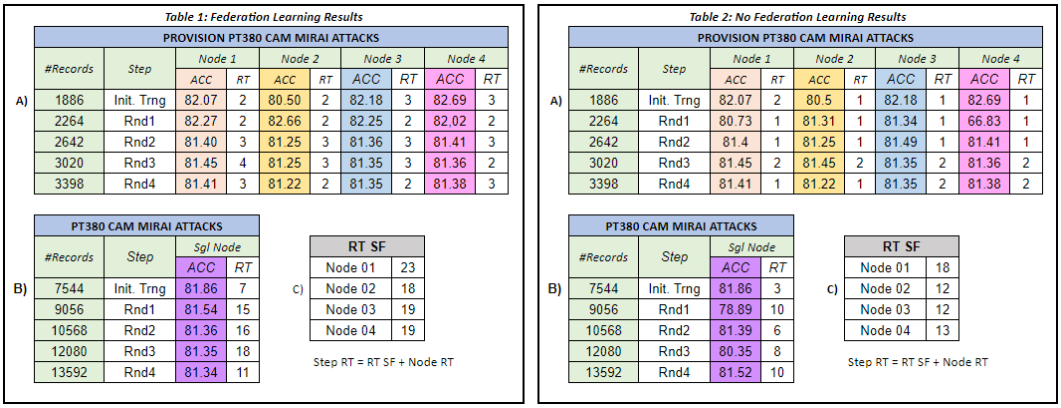

*Table 1: Federation Learning Results*

PROVISION PT380 CAM MIRAI ATTACKS

A)

| #Records | Step | Node 1 | | Node 2 | | Node 3 | | Node 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | ACC | RT | ACC | RT | ACC | RT | ACC | RT |
| 1886 | Init. Trng | 82.07 | 2 | 80.50 | 2 | 82.18 | 3 | 82.69 | 3 |
| 2264 | Rnd1 | 82.27 | 2 | 82.66 | 2 | 82.25 | 2 | 82.02 | 2 |
| 2642 | Rnd2 | 81.40 | 3 | 81.25 | 3 | 81.36 | 3 | 81.41 | 3 |
| 3020 | Rnd3 | 81.45 | 4 | 81.25 | 3 | 81.35 | 3 | 81.36 | 2 |
| 3398 | Rnd4 | 81.41 | 3 | 81.22 | 2 | 81.35 | 2 | 81.38 | 3 |

PT380 CAM MIRAI ATTACKS

B)

| #Records | Step | Sgl Node | |
|---|---|---|---|
| | | ACC | RT |
| 7544 | Init. Trng | 81.86 | 7 |
| 9056 | Rnd1 | 81.54 | 15 |
| 10568 | Rnd2 | 81.36 | 16 |
| 12080 | Rnd3 | 81.35 | 18 |
| 13592 | Rnd4 | 81.34 | 11 |

C)

| RT SF | |
|---|---|
| Node 01 | 23 |
| Node 02 | 18 |
| Node 03 | 19 |
| Node 04 | 19 |

Step RT = RT SF + Node RT

*Table 2: No Federation Learning Results*

PROVISION PT380 CAM MIRAI ATTACKS

A)

| #Records | Step | Node 1 | | Node 2 | | Node 3 | | Node 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | ACC | RT | ACC | RT | ACC | RT | ACC | RT |
| 1886 | Init. Trng | 82.07 | 2 | 80.5 | 1 | 82.18 | 1 | 82.69 | 1 |
| 2264 | Rnd1 | 80.73 | 1 | 81.31 | 1 | 81.34 | 1 | 66.83 | 1 |
| 2642 | Rnd2 | 81.4 | 1 | 81.25 | 1 | 81.49 | 1 | 81.41 | 1 |
| 3020 | Rnd3 | 81.45 | 2 | 81.45 | 2 | 81.35 | 2 | 81.36 | 2 |
| 3398 | Rnd4 | 81.41 | 1 | 81.22 | 1 | 81.35 | 2 | 81.38 | 2 |

PT380 CAM MIRAI ATTACKS

B)

| #Records | Step | Sgl Node | |
|---|---|---|---|
| | | ACC | RT |
| 7544 | Init. Trng | 81.86 | 3 |
| 9056 | Rnd1 | 78.89 | 10 |
| 10568 | Rnd2 | 81.39 | 6 |
| 12080 | Rnd3 | 80.35 | 8 |
| 13592 | Rnd4 | 81.52 | 10 |

C)

| RT SF | |
|---|---|
| Node 01 | 18 |
| Node 02 | 12 |
| Node 03 | 12 |
| Node 04 | 13 |

Step RT = RT SF + Node RT

**Fig. 7.** Results for Dataset Provision PT838 Security Camera Mirai Attacks

into two equal and balanced parts. The first part was used for the local learning process, where the client trains its own machine learning model using its respective data. The second part was reserved for real-time prediction testing, where the client evaluates the performance of its trained model on unseen data. This separation allows for an evaluation of both the local learning capability and the real-time prediction accuracy of each client in the collaborative setting. We proceeded to execute the workflow as illustrated in Figure 1, and we measured the execution time required for the attribute selection process, as well as the accuracy and loss achieved by each individual client. Following this, we recorded the time taken for the federated learning process to complete.

### 4.4 Second Scenario Simulation

In the second testing scenario, we performed the machine learning process on each node, without including the feature selection and federated learning process, simulating a traditional pipeline of isolated machine learning applied to discover attacks. This approach allowed us (by comparison) to explore the effectiveness of feature selection and the collaborative learning technique in improving model performance and overall accuracy in a federated setting.

## 5 Discussion of Results

In this section, we concentrate on discussing the most pertinent findings from the case study. We have chosen to highlight certain results because of their significant impact. These results are important for understanding how well machine learning models can detect security threats in IoT devices when using federated learning methods. By doing so, we can gain a deeper understanding of the approach implications for federated learning in IoT security.

Figures 4, 5, 6, 7, Table 1 - A shows the results obtained for Federated Learning Results in each round (1 to 4) plus in the Initial Training round (Init. Trmg). Table 1 - B shows the results when all data is used in a single node. These are divided into several columns: #Records (number of records), Step of the tests for Nodes 1, 2, 3 e 4. The ACC (accuracy in percentage) and RT (response time in seconds) columns display the corresponding values for each node at each stage. Table 1 - C shows the run time (RT) required to execute the selection feature (SF) plus the federated learning (FL) process for each node. Table 2 has a similar structure, but it presents the results obtained without the use of federated learning. Figures 4 and 5 show the results obtained when the dataset was intruded by Gafgyt attacks, and 6 and 7 show the results under Mirai attacks.

When observing the results (Figures 4–7), there are no established patterns in terms of accuracy. This is due to the dependency on the data arriving at each node. However, it is observed that the accuracy remains stable in all cases when the FL process is present, unlike the behavior that is observed when FL is not used. In this latter case, accuracy presents sudden drops in several cases (for example, Table 2 in Figure 7 for nodes 1 and 3). The stability in the FL model occurs because with each new round, as stated by the model in Figure 1 (item 5), the online detections at each node are validated by an expert who checks false positives and negatives and updates the training data in the local models, which are sent back to the server that updates the global model and sends it back to everyone involved. Therefore, all the nodes benefit from sharing the knowledge obtained in any local retraining. This is one of the major advantages of federated learning, as the nodes that did not have this knowledge are benefited by the received global model without having to share their data, thus ensuring privacy.

The FedAvg algorithm is used in this process and helps mitigate drastic changes or extreme oscillations in the local models [28]. It performs a weighted average of the parameters of the local models, where the weights are determined based on the amount of training data at each node. This means that nodes with more training data have a greater contribution to the global model. By performing this weighted average, FedAvg smooths out the differences between the local models and prevents a single model from having excessive influence on the global model [28].

Regarding execution time, in most rounds and on average, the time was higher for the FL process. As expected, when comparing the time spent on data processing with the FL process (Table 1 - part B) and without the FL process (Table 2 - part B), it is significantly higher. In some cases, it can be more than 60% (in fact, the average is around this mark). Another advantage of the proposed model is that it conceives of data processing in a distributed manner and therefore benefits from the advantages of distribution. The execution time spent processing the distributed data (any table A) is lower when compared with the process on a single node (any table B), in some cases almost three times lower.

## 6 Conclusion and Future Works

This work presented a self-adaptive model that is based on feature selection and federated learning for detecting cyberattacks in IoT datasets. Furthermore, the work brings two sets of experiments. By comparing the results with and without the use of federated learning and feature selection, it is possible to infer that the model has good accuracy in detecting attacks but increases the response time up to around 60% in average.

One potential approach for further work is to explore the application of vertical federated learning by incorporating heterogeneous types of devices that share similar contexts. This approach would involve leveraging the collective intelligence and computational capabilities of diverse devices, such as sensors, actuators, and controllers, to collaboratively train a shared model while preserving data privacy and security. Additionally, it would be valuable to validate the effectiveness and practicality of the developed model in a real-world scenario. This validation could involve deploying the federated learning system in an actual cyber-physical system environment, collecting real-time data, and evaluating the model's performance in detecting and mitigating cybersecurity attacks. Such research would provide valuable insights into the feasibility and benefits of vertical federated learning for enhancing the security and resilience of cyber-physical systems.

## Acknowledgment

This work has been partially supported by **ADVANCE**(http://advance-rise.eu/ - Horizon 2020-MSCA-RISE grant agreement No 2018-823788) projects. Also, it was financed by CAPES - Brasil, Finance code 001. Special thanks to the Senior Information Technology Management of SESI-SP and SENAI-SP for their cooperation and support.

## References

1. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B. Communication-efficient learning of deep networks from decentralized data. In Proc. of the 20th Int. Conf. on Artificial Intelligence and Statistics (AISTATS), 2017.
2. Solms, R. V., Niekerk, J. V. From information security to cybersecurity. Computers & Security, v. 38, p. 97-102, 2013.
3. Kenyon, T. Top 10 cybersecurity threats. Cybermagazine.com. 2021 Available at https://cybermagazine.com/cyber-security/top-10-cyber-security-threats. Last access Feb/2022.
4. Lorenzo, P., Stefano F, Ferreira A, Carolina P. Artificial Intelligence and Cybersecurity: Technology, Governance and Policy Challenges. 2021.
5. Reznik, L. Computer Security with Artificial Intelligence, Machine Learning, and Data Science Combination. In: Intelligent Security Systems: How Artificial Intelligence, Machine Learning and Data Science Work For and Against Computer Security , IEEE, 2022, pp.1-56, doi: 10.1002/9781119771579.ch1.

6. Sarker, I.H., Furhad, M.H., Nowrory, R. AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions. SN COMPUT. SCI. 2, 173 (2021). https://doi.org/10.1007/s42979-021-00557-0.
7. Feng X., Feng Y., Dawam E. S. Artificial Intelligence Cyber Security Strategy. In: IEEE Intl DASC/PiCom/CBDCom/CyberSciTech Conferences. IEEE, 2020. p. 328-333.
8. GDPR. General Data Protection Regulation. Available at https://gdpr-info.eu. Last access Aug/2023.
9. SAS. Machine learning: o que é e qual sua importância? Sas.com. Available at https://www.sas.com/pt_pt/insights/analytics/machine-learning.html. Last access Mar/2022.
10. Silva, H. "CSAI-4-CPS: A Cyber Security characterization model based on Artificial Intelligence For Cyber Physical Systems," 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S), Baltimore, MD, USA, 2022, pp. 47-48, doi: 10.1109/DSN-S54099.2022.00032.
11. Silva, H., Basso, T., Moraes, R. Feature Selection: supporting the mining process on cyber-physical systems result datasets. In: Proc.of XXII WTF-SBC, 2021. p. 15-28.
12. GOOGLE. Federated Learning: Collaborative Machine Learning without Centralized Training Data. Available at https://ai.googleblog.com/2017/04/ federated-learning-collaborative.html. Last access Ago/2023.
13. Bonawitz, K., Eichner H., Grieskamp W., Huba D., Ingerman A., Ivanov V., Kiddon C., Konecny J., Mazzocchi S., Mcmahan H., Van O. T., Petrou D., Ramage D., Toselander J. Towards federated learning at scale: System design. Proc. of Machine Learning and Systems 1, pp. 374-388, 2019.
14. Jahromi, A. N., Sakhnini, J., Karimpour, H., & Dehghantanha, A. A deep unsupervised representation learning approach for effective cyber-physical attack detection and identification on highly imbalanced data. In: Proc. of the 29th Annual Int. Conf. on Computer Science and Software Engineering, pp. 14-23, 2019.
15. Siddique, A., Browne, W. N., Grimshaw, G. M. Lateralized learning for robustness against adversarial attacks in a visual classification system. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference, pp. 395-403, 2020.
16. Shahid O., Mothukuri V., Pouriyeh S., Parizi R. M., Shahriar H. Detecting Network Attacks using Federated Learning for IoT Devices. In: IEEE 29th Int. Conf. on Network Protocols (ICNP), 2021.
17. Alves, T. A., França, F. M., Kundu, S. Mlprivacyguard: Defeating confidence information based model inversion attacks on machine learning systems. In Proceedings of the 2019 on Great Lakes Symposium on VLSI, pp. 411-415, 2019.
18. Kanimozhi, V., Jacob, T. P. Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. In: Int. conf. on communication and signal processing (ICCSP), pp. 0033-0036, IEEE, 2019.
19. Cao, Y., Chen, X., Yao, L., Wang, X., Zhang, W. E. Adversarial attacks and detection on reinforcement learning-based interactive recommender systems. In: Proc. of the 43rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 1669-1672, 2020.
20. Hu, Z., Zhu, M., Liu, P. Adaptive cyber defense against multi-stage attacks using learning-based POMDP. ACM Trans. on Privacy and Security (TOPS), 24(1), 1-25, 2020.
21. Xu, H., Yu, W., Liu, X., Griffith, D., Golmie, N. On data integrity attacks against industrial Internet of Things. In 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp. 21-28. IEEE, 2020.
22. COLAB. Overview of Colaboratory Features. Google.com. Available at https://colab.research.google.com/notebooks/basic_features_overview.ipynb/. Last access July/2023.

23. KDNUGGETS. Deep Learning Development with Google Colab, TensorFlow, Keras & PyTorch. KDnuggets. Available at https://www.kdnuggets.com/2018/02/google-colab-free-gpu-tutorial-tensorflow-keras-pytorch.html. Last access July/2023.

24. KERAS. Deep Learning for humans. Keras.io. Available at https://keras.io/. Last access July/2023.
25. Palotay, D. The IoT Botnet Report 2021: Malware and Vulnerabilities Targeted. Available at https://cujo.com/blog/iot-botnet-report-2021-malware-and-vulnerabilities-targeted/. Last access jan/2024.
26. Y. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y. N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. In IEEE Pervasive Computing, vol. 17, no. 3, pp. 12-22, Jul.-Sep. 2018, doi: 10.1109/MPRV.2018.03367731.
27. UCI. N BaIoT dataset description. Uci.edu. 2018. Available at https://archive.ics.uci.edu/dataset/442/detection+of+iot+botnet+attacks+n+baiot Last access jan/2024.
28. Guendouzi, B. S., Ouchani, S., Assaad, H. E., & Zaher, M. E. 2023. A systematic review of federated learning: Challenges, aggregation methods, and development tools. Journal of Network and Computer Applications, 103714.

## Authors

**Hebert de Oliveira Silva** received a Master of Technology in the area of Information Systems and Communication from the State University of Campinas (2019). He completed his degree in Computer Network Management in Information Technology from University Paulista (2010) and pursued a postgraduate Master of Business Administration in IT Governance at SENAC University (2014). Currently, he is pursuing his Ph.D. in Computer Science at the State University of Campinas. Additionally, he is the Cybersecurity Coordinator at SENAI Sao Paulo, and his research interests include Information Security, Cryptography, Cybersecurity, Data Privacy, and Security for CPS (Cyber-Physical Systems) and IoT (Internet of Things).

**Regina Lúcia de Oliveira Moraes** Graduated in Computer Science from the State University of Campinas (1978), she obtained her Master's degree in Computer Science from the State University of Campinas (2003), followed by a Ph.D. in Computer Science from the same university in 2006. She completed her postdoctorate at the Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS) in Toulouse, France. She is currently a retired full professor at the State University of Campinas and collaborates in the Postgraduate Program at the Faculty of Technology at UNICAMP. She is also an invited full professor at University of Coimbra, Portugal. She has experience in the field of Computer Science, with an emphasis on Software Engineering, and works mainly on the following topics: software testing, software security, data privacy, fault injection, software management, and educational software.