Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT

Pedro Ruzafa-Alcázar, Pablo Fernández-Saura, Enrique Mármol-Campos, Aurora González-Vidal, José L. Hernández-Ramos, Jorge Bernal-Bernabe, and Antonio F. Skarmeta, *Member, IEEE*

***Abstract*—Federated learning (FL) has attracted significant interest given its prominent advantages and applicability in many scenarios. However, it has been demonstrated that sharing updated gradients/weights during the training process can lead to privacy concerns. In the context of the Internet of Things (IoT), this can be exacerbated due to intrusion detection systems (IDSs), which are intended to detect security attacks by analyzing the devices' network traffic. Our work provides a comprehensive evaluation of differential privacy techniques, which are applied during the training of an FL-enabled IDS for industrial IoT. Unlike previous approaches, we deal with nonindependent and identically distributed data over the recent ToN_IoT dataset, and compare the accuracy obtained considering different privacy requirements and aggregation functions, namely FedAvg and the recently proposed Fed+. According to our evaluation, the use of Fed+ in our setting provides similar results even when noise is included in the federated training process.**

***Index Terms*—Differential privacy (DP), federated learning (FL), Internet of Things (IoT), intrusion detection systems (IDSs), machine learning.**

I. Introduction

As THE Internet of Things (IoT) expands, there is a significant increase in the number and impact of security vulnerabilities and threats associated with IoT devices and systems. To cope with such concerns, intrusion detection systems (IDSs) represent a well-known approach for early detection of IoT attacks and cyber-threats [1]. In recent years, IDS mechanisms are usually based on artificial intelligence (AI) techniques, so that the system is trained with devices' network traffic to accurately detect any anomalous behavior, which could represent a certain type of attack [2]. Indeed, AI-based IDS are trained, considering monitored network traffic and behavioral data from heterogeneous IoT devices deployed in remote, possibly untrusted, and distributed domains and systems, to increase the overall accuracy for attack detection. However, this approach sparks privacy issues as different domains might need to share their private data [3].

As an alternative to typical centralized learning approaches, federated learning (FL) was proposed in 2016 [4] as a collaborative learning approach, in which an AI algorithm is trained locally across multiple decentralized edge devices, called clients or parties, and the information is continuously updated onto a global model through several training rounds. Instead of sharing their data, parties share their models with an aggregator, which computes a global model. Nonetheless, FL suffers from privacy issues, as the global model's updates provided by parties could be used to launch several attacks to infer the private information of the training data [5]. To mitigate such privacy concerns, differential privacy (DP) [3] can be employed to obfuscate either the training data or model updates, giving statistical privacy guarantees over the data against an adversary. DP is usually considered in the scope of FL settings due to the stringent communication requirements of other privacy-preserving approaches, such as secure multiparty computation (SMC) [6].

While the use of DP techniques has been considered in FL [7], [8], existing works do not analyze the impact of such techniques in the scope of IDS approaches, and do not address the impact of different aggregation methods considering nonindependent and identically distributed (non-i.i.d.) data distributions, which are common in real-world scenarios. In this direction, our work provides a comprehensive evaluation of DP approaches through several additive noise techniques based on Gaussian and Laplacian distributions, which are applied during the training of an FL-enabled IDS for industrial IoT (IIoT). Unlike previous approaches, our evaluation is based on an instance selection process to deal with non-i.i.d. data distributions over the recent ToN_IoT dataset [9], which contains recent attacks related to such scenarios. Furthermore, unlike the current state of the art, our evaluation compares the accuracy obtained by using different DP techniques and a recently proposed aggregation function

Manuscript received 30 July 2021; revised 25 October 2021; accepted 31 October 2021. Date of publication 9 November 2021; date of current version 13 December 2022. This work was supported in part by the UMU-Campus Living Lab under Grant EQC2019-006176-P funded by ERDF funds, by Grant PID2020-112675RB-C44 funded by MCIN/AEI/10.13039/5011000011033 (ONOFRE Project), by the European Commission through EU Projects the PHOENIX under Grant 893079 and through the DEMETER under Grant 857202; in part by the European Social Fund (ESF); and in part by the Youth European Initiative (YEI) under the Spanish Seneca Foundation (CARM). Paper no. TII-21-3297. *(Corresponding author: Aurora González-Vidal.)*

Pedro Ruzafa-Alcázar, Pablo Fernández-Saura, Enrique Mármol-Campos, Aurora González-Vidal, Jorge Bernal-Bernabe, and Antonio F. Skarmeta are with the Department of Information and Communication Engineering, University of Murcia, 30100 Murcia, Spain (e-mail: pedro.ruzafaa@um.es; pablo.fernandezs2@um.es; enrique.marmol@um.es; aurora.gonzalez2@um.es; jorgebernal@um.es; skarmeta@um.es).

José L. Hernández-Ramos is with European Commission, Joint Research Centre, 21027 Ispra, Italy (e-mail: jose-luis.hernandez-ramos@ec.europa.eu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3126728.

Digital Object Identifier 10.1109/TII.2021.3126728

called Fed+ [10], which provides significantly better accuracy results compared to the traditional FedAvg function [4]. To the best of our knowledge, this is the first effort to analyze the impact of non-i.i.d. data and aggregation functions for the implementation of a privacy-preserving FL-enabled IDS in IoT/IIoT scenarios.

Based on this, the main contributions and novelties of this article are as follows.

1) A thorough evaluation on the feasibility and performance of applying DP-enabled FL to detect attacks in IoT scenarios, adapting and partitioning the ToN_IoT dataset considering non-i.i.d. data distributions.
2) An empirical analysis of using different FL-aggregation methods with DP techniques, and their impact on the effectiveness for intrusion detection in IoT.
3) This article accomplishes the first complete quantitative and computational performance analysis of diverse DP perturbation mechanisms applied to FL for intrusion detection in IoT, using different privacy-factor values and FL settings.

The rest of this article is organized as follows. Section II provides an overview of FL, highlighting the main privacy issues and potential mitigation approaches. Section III describes the DP-enabled FL architecture, including the proposed training algorithm based on several DP techniques. Section IV describes our methodology for the proposed privacy-preserving FL-enabled IDS, including the aspects of the used dataset, classification model, and aggregation functions. Section V describes the evaluation results. Section VI analyzes the current state of the art. Finally, Section VII concludes this article.

II. Privacy-Preserving FL

The training process in FL is based on a set of rounds, in which the coordinator selects a subset of clients depending on the problem context, and sends the parameters of a global model to them. Then, those parameters are updated by each client using their own collected data and they are sent back to the coordinator. The entities make use of a certain aggregation method to fusion the parameters received from the clients. While FedAvg [4] represents the most common approach, in which a simple average is applied over such parameters, other methods have recently been proposed to increase the accuracy of the trained model, particularly in settings with non-i.i.d. data distributions. Indeed, as described in Section V, our work evaluates the recent Fed+ algorithm [10], which clearly improves FedAvg for our particular scenario.

The main advantage of FL is that parties do not need to share their data for training a certain model. However, recent works highlight the need to apply privacy-preserving mechanisms to address potential attacks derived from the sharing of parameters/weights throughout the training rounds [5]. In particular, a malicious aggregator could modify the received parameters to fool the model being trained. Even an honest-but-curious aggregator might perform a *reconstruction attack* to infer training data from these parameters using several techniques, such as generative adversarial networks (GANs). Indeed, the use of GAN can also be considered to launch *membership inference attacks*, in which an attacker could infer if local data of a certain party were used for the training process [6]. These attacks can be also carried out by external entities to the training process, as well as compromised FL clients. In the context of IoT/IIoT, the impact of these attacks can be significant due to the potential sensitivity of the network data required for the implementation of an IDS in such scenarios.

To deal with the aforementioned privacy issues, different techniques have been postulated, including SMC and DP [5]. SMC uses different cryptographic protocols to jointly calculate a function by using a set of input values, which are kept private for the parties. In the case of an FL environment, the parameters/weights produced by FL clients are kept private when they are fused by the aggregator. However, recent works [8] highlight the high computational and communication requirements associated with the use of SMC that can make these techniques unfeasible for IoT environments. Moreover, DP is based on injecting random noise into a dataset so that, looking at the output of a certain function over the dataset, it is not possible to discern if a certain data was used in such a dataset.

According to [3], the formal definition of DP is based on two concepts.

Definition 1 (Mechanism): A mechanism κ is a random function that takes a dataset D and outputs a random variable $\kappa(D)$.

For example, if the input is an IoT attacks dataset, then the output can be the flow duration plus noise from the standard normal distribution. In our case, the inputs will be the weights of a model.

Definition 2 (Distance): The distance of two datasets D and D' denotes the minimum number of sample changes that are required to change D into D'.

For example, if D and D' differ on at most one individual, there is $d(D, D') = 1$. We also call such a pair of datasets neighbors.

Definition 3 (Differential Privacy): A mechanism κ satisfies (ϵ, δ)-DP if and only if for all neighbor datasets D and D', and $\forall S \subseteq \text{Range}(\kappa)$, as long as the following probabilities are well defined, there holds

$$Pr(\kappa(D) \in S) \leq e^{\epsilon} Pr(\kappa(D') \in S) + \delta$$
$$\text{where } \delta, \epsilon > 0$$

δ represents the probability that a κ output varies by more than a factor of e^{ϵ} when applied to a dataset and any one of its neighbors. This definition captures the intuition that a computation on private data will not reveal sensitive information about individuals in a dataset if removing or replacing an individual in the dataset has a negligible effect on the output distribution.

A lower value of δ implies a greater confidence, and a smaller value of ϵ tightens the privacy protection. This can be seen because the lower δ and ϵ are, the closer $Pr(\kappa(D) \in S)$ and $Pr(\kappa(D') \in S)$, and therefore, the protection is stronger. As described in [3], when $\delta = 0$, (ϵ,0)-differential is simplified to ϵ-DP. If $\delta > 0$, there is still a small chance that some information is leaked. In the case of $\delta = 0$, the guarantee of information leakage is not probabilistic. Therefore, ϵ-DP provides stronger

TABLE I
DIFFERENT VARIABLES

Federated Learning Parameters	
W^G	Global model weights
w_i^G	Global model weight i
W^k	Party k weights
w_i^k	Party k weight i
r	Round
E	Epoch
α	Learning rate
Differencial Privacy Parameters	
ϵ	privacy parameter (user-defined)
δ	prob. of leaking information (user-defined)
r	correlation coefficient
x_i	values of the x-variable in a sample
$\bar{x}$	mean of the values of the y-variable
y_i	values of the y-variable in a sample
$\bar{y}$	mean of the values of the y-variable

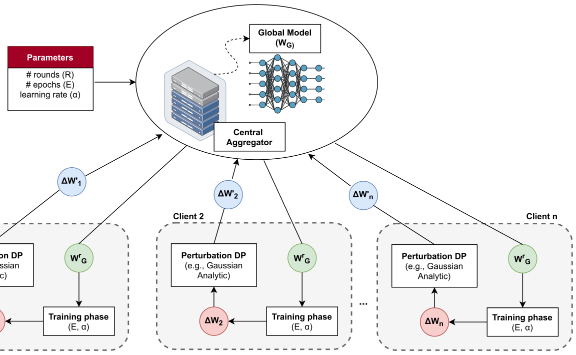

Fig. 1. Proposed FL with DP architecture.

privacy guarantees than (ϵ, δ)-DP. For this reason, we have chosen for our approach $\delta = 0$.

As described in Section V, our work evaluates different DP techniques based on Gaussian and Laplacian distributions, which are further detailed in Section III.

III. Proposed Privacy-Preserving FL-Enabled IDS for IoT/IIoT

Based on the aforementioned aspects of privacy-preserving approaches for FL, in the following we describe the proposed architecture and algorithm enabling the FL training process. Furthermore, we provide a formal description of the specific DP techniques being considered in our work. For the sake of clarity, Table I provides the meaning of the main variables and terms, which are used in such description.

A. DP-Enabled FL

The overall architecture of our DP-enabled FL approach for intrusion detection in IoT is shown in Fig. 1. A *client* represents the end device where the local training is performed. Each client is in charge of training the global model sent by the aggregator with its local data and generating a model update. Furthermore, clients are endowed with the logic needed to apply the corresponding DP algorithm. Moreover, the *aggregator* is the central service that receives the model updates coming from the clients and generates an *aggregated model*, which is sent back to the clients for each training round.

In particular, considering the use of DP in the federated training, the process in each training round is as follows.

1) The aggregator selects a set of clients to participate in the training process by considering a certain *client selection* approach. In the context of IoT, operational conditions of the device (e.g., battery consumption) could be considered.
2) In the case of the initial training round, the aggregator creates a new general model W^G, whose weights (w_i^G) are sent to the selected clients. Otherwise, the aggregator creates an aggregated model by using a certain aggregation function based on the updates provided by the clients. For this work, we compare the use of *FedAvg* [4] and Fed+ [10] aggregation methods, which are further explained in Section IV. The resulting aggregated model is sent to the clients.
3) Each client takes the shared model and trains it with its local data, generating a particular model update with the weights $\triangle W_i^k$ that results from local training. It should be noted that the number of epochs executed by a client is determined by the number of epochs E.
4) The clients anonymize the calculated weights leveraging a DP mechanism. The particular DP perturbation mechanism applied in this step could be one of the approaches defined in the following section. Then, the resultant anonymized weights $\triangle W_i$ are sent back to the aggregator entity.
5) The aggregator combines all the received model updates using an aggregation algorithm, such as FedAvg, generating a new global model.

These steps are repeated until a certain number of rounds R is reached, or another condition is met, such as achieving a certain target accuracy.

Based on the previous description, Algorithm 1 provides a detailed description of the required steps for the DP-enabled federated training process, showing the integration and relationship between the FL aggregation method and DP mechanism applied in each round. It should be noted that our work is focused on evaluating several DP techniques under different privacy requirements to come up with a potential tradeoff between privacy and accuracy. These techniques are further described in the following.

B. Perturbation DP Mechanisms

As depicted in Definition 1, an output perturbation mechanism takes an input D and returns a random variable $\kappa(D)$. Such a random variable is computed using the addition of the transformation of the input data by means of a function $f : X \to \mathbb{R}^d$ to some random noise that follows a certain distribution rn; therefore, we could express the $(\epsilon, \delta) - DP$ as follows: $\kappa(D) = f(D) + rn$.

The perturbation methods analyzed in this article are summarized as follows.

1) *Truncated Laplacian mechanism [11]:* Given the privacy parameters $0 < \delta < 1/2$ and $\epsilon > 0$ and query sensitivity $\Delta > 0$, the density probability of the truncated Laplacian

distribution function ρ_{TLap} is defined as

$$f_{\text{TLap}}(x)\begin{cases}\mathrm{B}e^{-\frac{x}{\lambda}}, & \text{for x} \in [-A, A]\\ 0, & \text{otherwise}\end{cases}$$

where

$$\lambda = \frac{\Delta}{\epsilon} \tag{1}$$

$$A = \frac{\Delta}{\epsilon}\log\left(1 + \frac{e^{\epsilon} - 1}{2\delta}\right)$$

$$B = \frac{1}{2\lambda\left(1 - e^{-\frac{A}{\lambda}}\right)} = \frac{1}{2\frac{\Delta}{\epsilon}\left(1 - \frac{1}{1+\frac{e^{\epsilon}-1}{2\delta}}\right)}.$$

2) *Uniform:* A special case of the truncated Laplacian mechanism [11] when $\epsilon = 0$.
3) *Gaussian [12]:* For any $\epsilon, \delta \in (0, 1)$, Gaussian output perturbation mechanism is

$$\sigma = \Delta(\sqrt{2\log(1.25/\delta)})/\epsilon. \tag{2}$$

4) *Analytic Gaussian mechanism [12]:* Let $f : \mathbb{X} \rightarrow \mathbb{R}^d$ be a function with global L_2 sensitivity Δ. For any $\epsilon \geq 0$ and $\delta \in [0, 1]$, the Gaussian output perturbation mechanism $M(x) = f(x) + Z$ with $Z \sim N(0, \sigma^2 I)$ is (ϵ, δ)-DP if and only if

$$\phi\left(\frac{\Delta}{2\sigma} - \frac{\epsilon\sigma}{\Delta}\right) - e^{\epsilon}\phi\left(-\frac{\Delta}{2\sigma} - \frac{\epsilon\sigma}{\Delta}\right) \leq \delta. \tag{3}$$

5) *Bounded domain Laplace mechanism [13]:* Given $b > 0$ and $D \subset \mathbb{R}$, the bounded Laplace mechanism $W_q : \Omega \rightarrow D$, for each $q \in D$, is given by its probability density function f_{W_q}

$$f_{W_q(x)} = \begin{cases}0 & \text{if } x \notin D\\ \frac{1}{C_q}\frac{1}{2b}e^{-\frac{|x-q|}{b}}, & \text{if } x \in D\end{cases} \tag{4}$$

where $C_q = \int_D \frac{1}{2b}e^{-\frac{|x-q|}{b}}dx$ is a normalization constant. Let $\epsilon \geq 0$, $\delta \in [0, 1]$, and query sensitivity $\Delta > 0$, then the mechanism W_q is (ϵ, δ)-DP whenever

$$b \geq \frac{\Delta}{e - \log\Delta C(b) - \log(1 - \delta)}. \tag{5}$$

It is needed to note that C_q is a function of b, and that the domain $D = [l, u]$ having lu. Then, the following holds:

$$\max_{\substack{q,q'\in D\\ |q'-q|\leq\Delta}} \frac{C_{q'}}{C_q}e^{\frac{|q-q'|}{b}} = C(b)e^{\frac{\Delta}{b}}$$

when we define $\Delta C(b)$ as follows:

$$\Delta C(b) = \frac{C_{l+\Delta}(b)}{C_l(b)}. \tag{6}$$

6) *Bounded Laplace noise mechanism [14]:* This algorithm adds independent noise in each of the k coordinates, drawn from the distribution $\mu_{DE,R}$ that is supported in $(-R, R)$

$$M_{DE,R}(\vec{x}^{(1)}, \ldots, \vec{x}^{(n)}) = \frac{1}{n}\sum_{i=1}^{n}\vec{x}^{(i)} + \vec{\eta} \tag{7}$$

where $\vec{x}^{(i)}$ and each η_i is drawn independently from the distribution $\mu_{DE,R}$. Let $e \in (0, 1), k \in \mathbb{N}$, and $\delta \geq e^{\frac{-k}{\log^{8}\tilde{}\, k}}$. Define $R = \frac{C}{\epsilon n}\sqrt{\log(1/\delta)}$, where $C > 0$ is a universal constant, then $M_{DE,R}$ is $(\epsilon, \delta) - DP$.

As we can understand from the definitions of the DP methods, the level of perturbation will highly depend on the chosen parameter ϵ. In that sense, we wanted to analyze how the similarity between the perturbed weights and the original ones is affected by ϵ, and we used the Pearson correlation to do so. This coefficient is a measure of the linear association of two variables, and it is measured from -1 to +1. A value of +1 indicates that the data objects are perfectly correlated, and a score of -1 means that the data objects are not correlated.

Essentially, the Pearson correlation coefficient (PCC) is the ratio between the covariance and the standard deviation of two variables. In mathematical form, the coefficient can be described as

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2\sum(y_i - \overline{y})^2}} \tag{8}$$

where r is the correlation coefficient, x_i represents the values of the *x*-variable in a sample, $\bar{x}$ is the mean of the values of the *x*-variable, y_i represents the values of the *y*-variable in a sample, and $\bar{y}$ is the mean of the values of the *y*-variable.

In our approach, the PCC, defined in (8), is calculated over the model updates for every training round, before and after applying the DP mechanism, as defined in (1)–(7). This means that this metric indicates the similarity between the original weights and the modified ones for every client.

IV. Methodology

Before describing our evaluation results, this section provides an overview of different aspects of our approach, including the dataset, machine learning classification technique, and aggregation functions being considered.

A. Dataset Description

The dataset used in this article is based on the CIC-ToN_IoT dataset,[1] which was generated through the CICFlowMeter tool from the original PCAP files of the ToN_ IoT dataset [9] to extract 83 features. As a first step, we remove the nonnumeric features (e.g., flow ID). Then, we separate the samples of the whole dataset based on the destination IP addresses, i.e., victims' addresses, and then remove the samples that do not correspond to the top ten IP addresses, sorted by number of samples. The resulting dataset contains 4.404.084 samples, which correspond to the 82.29% of the original CIC-ToN-IoT.[2]

[1] [Online]. Available: https://staff.itee.uq.edu.au/marius/NIDS_datasets/

[2] The used dataset is available and can be downloaded from the following link: [Online]. Available: https://mega.nz/folder/FIoClAKT#ueQbjk5iU7J8XW12lLuvaw

Algorithm 1: Algorithm of Our DP Framework:

```
Input: Number of rounds $R$, number of parties $P$,
number of epochs $E$ and learning rate $\alpha$
Output: Global model $W^G$
[party[i]]
LocalUpdate($i$, $W_r^G$):
Let x be the input and y the labels of the local data
Normalize local inputs
Replace the parameters in the local model in order to get
$W_i^r$
Initialize $w = 0^{m-1}, b = 0$
for each epoch from 1 to E do
    Compute prediction $\hat{y}_k = h(x_k)$
    Compute loss $\mathcal{L}_k = L(\hat{y}_k, y_k)$
    Compute the gradients $\triangle w = -\nabla_{\mathcal{L}_k} w, \triangle b = \frac{-\partial \mathcal{L}}{-\partial b}$
    Update parameters $w = w + \triangle w, b = +\triangle b$
end for
Apply the $(\epsilon, \delta) - DP$ mechanism to the weights $w$ to
get $\kappa(w)$.
return $\kappa(w)$ to the server
[server]
initialize $W_0^G$
for each iteration r from 0 to R do
      $S_r$ = Choose p parties out of P
      for each party i $\in S_r$ do
      $\triangle w_r^i$ = LocalUpdate($i$, $W_r^G$)
    end for
    Calculate new weights using Fed+:
    $w_{r+1}^k = w_r^k - \alpha^k[\nabla f_k(w_r^k) + \gamma_k \nabla B(w_r^k, C(w))]$
end for
```

Afterward, as the resulting datasets are highly imbalanced, we use Shannon entropy to measure the imbalance of each one of them. The main reason to do so is that the use of imbalanced and non-i.i.d. data has a significant impact on FL scenarios [15]. In particular, given a dataset of length n, and k classes of size c_i, the balance between the classes is given by the formula

$$\text{Entropy} = \frac{-\Sigma_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n}}{\log k} \tag{9}$$

where the function is equal to 0 if all classes are 0 except one, and is equal to 1 if all $c_i = \frac{n}{k}$.

Furthermore, it should be noted that we consider that one FL client is represented by a single victim IP address. In this context, given that each instance represents a network flow, n is the number of network flows, k is the number of the attack classes, and c_i is the number of instances of the class i.

Table II describes the data distribution of the different parties, including the distribution of classes and the entropy values for each party. Based on such values, we select the parties with an entropy value higher than 0.2, so parties 0, 2, 4, and 5 are selected as the FL clients for our scenario.

Then, after this initial step, due to the fact that the classes of each local dataset are not well balanced, we use a simple instance selection mechanism based on undersampling, which consists in removing random samples from the predominant classes until we reach an entropy level higher than 0.6. First, from these classes, we select the instances that satisfy the entropy requirement, and then we randomly remove these selected instances. Table III summarizes the resulting data distribution of parties that are selected for the DP-enabled federated training process.

B. Multiclass Classification

In our approach, we use supervised learning considering a multiclass approach to classify the dataset instances into benign or a specific attack, namely, DoS, DDoS, Backdoor, Injection, MITM, Scanning, Password, and XSS. Specifically, we apply the multinomial logistic regression [16], also called softmax regression, due to its training efficiency. The algorithm was provided by the sklearn library[3]. It can also interpret model coefficients as indicators of feature importance. As with most classifiers, the input variables need to be independent for the correct use of the algorithm. Given the input x, the objective is to know the probability of y (the label) in each potential class $p(y = c|x)$. The softmax function takes a vector z of k arbitrary values and maps them to a probability distribution as follows:

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)}.$$

In our case, the input to the softmax will be the dot product between a weight vector w and the input vector x plus a bias for each of the k classes

$$p(y = c|x) = \frac{\exp(w_c \dot{x} + b_c)}{\sum_{j=1}^{k} \exp(w_j \dot{x} + b_j)}.$$

The loss function for multinomial logistic regression generalizes the loss function for binary logistic regression and is known as the cross-entropy loss or log loss. While other supervised techniques could be employed, it should be noted that our work is focused on analyzing the impact of DP techniques considering different privacy requirements and aggregation functions in an FL setting.

C. Aggregation Functions

As described in Section II, the local updates generated by each client in FL are combined through an aggregation function in each training round. The most basic aggregation function is represented by FedAvg [4], which generates the global model based on the average of the weights generated by the FL clients. In particular, let $W^G = (w_i^G)$ be the weights of the general model and $W^k = (w_i^k)$ the weights of the party k, then

$$w_i^G = \sum \frac{D_i}{D} w_i^k$$

where D and D_i are the total data size and data size of each party, respectively.

However, the performance of FedAvg may be degraded in scenarios with non-i.i.d. and highly skewed data, as this case. In this work, we also consider a recent approach called Fed+ [10], which unifies several functions to cope with scenarios

[3] [Online]. Available: https://scikit-learn.org/

TABLE II
DESCRIPTION OF THE PARTIES CONSIDERED IN OUR DATASET

Party	Total samples	Benign	XSS	Injection	Password	Scanning	MITM	DDoS	Dos	Backdoor	Entropy
0	811504	42527	474520	140519	140519	13419	-	-	-	-	0.52
1	763518	763516	2	-	-	-	-	-	-	-	0
2	740117	116540	594627	16271	1138	10923	253	202	145	18	0.287
3	519806	519804	2	-	-	-	-	-	-	-	0
4	424531	2794	307962	66812	38009	8954	-	-	-	-	0.389
5	330956	10537	206036	44043	67431	2909	-	-	-	-	0.473
6	223092	3587	209637	9868	-	-	-	-	-	-	0.12
7	217737	217737	-	-	-	-	-	-	-	-	0
8	186891	8981	177910	-	-	-	-	-	-	-	0.088
9	185932	8551	177381	-	-	-	-	-	-	-	0.085

TABLE III
DESCRIPTION OF THE PARTIES PARTICIPATING IN THE PROPOSED DP-ENABLED FL SCENARIO

Party	Total samples	Benign	XSS	Injection	Password	Scanning	MITM	DDoS	Dos	Backdoor	Entropy
0	205946	42527	50000	50000	50000	13419	-	-	-	-	0.69858
2	42679	10000	10000	10000	1138	10923	253	202	145	18	0.70266
4	71748	2794	20000	20000	20000	8954	-	-	-	-	0.66218
5	73446	10537	20000	20000	20000	2909	-	-	-	-	0.66888

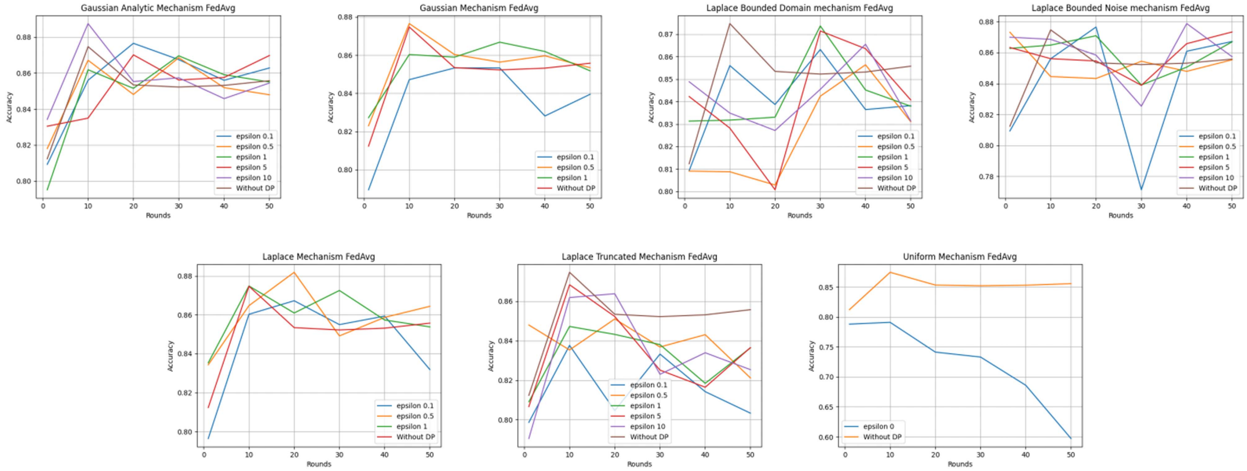

Fig. 2. FedAvg accuracy results for all perturbation mechanisms.

composed by heterogeneous data distributions. For this purpose, Fed+ relaxes the requirement of forcing all parties to converge on a single model. In particular, let the main objective in FedAvg be

$$\min F(W) = \frac{1}{D}\sum f_k(W^k)$$

where f_k is the local loss function of the party k. In the case of Fed+, the main objective is

$$\min F(W) = \frac{1}{D}\sum f_k(W^k) + \alpha_k B(W^k, C(W))$$

where W are the global model weights, W^k are the weights of the party k model, $B(\cdot,\cdot)$ is a distance function, $\alpha_i > 0$ are penalty constants, and C is an aggregate function that computes a central point of W.

Then, to calculate the weights in each round, parties generate their respective W^k, and send it to the aggregator. Afterward, the aggregator calculates the value of $C(W)$ and then sends it to the parties. Finally, the parties calculate the new model weights through

$$W^k_{r+1} = W^k_r - \gamma^k[\nabla f_k(W^k_r) + \alpha_k \nabla B(W^k_r, C(W))]$$

where γ^k are the learning rates, and W^k_r represents the weights of party k at round r.

V. EVALUATION RESULTS

This section describes the evaluation results achieved by applying each one of the previously described DP mechanisms to the FL training scenario with a logistic regression classifier over the ToN_IoT dataset. The main performance evaluation parameters are ϵ, the perturbation DP mechanism, and the aggregation algorithm to be applied. The evolution in terms of accuracy for each mechanism and different ϵ values throughout the rounds are shown in Figs. 2 and 3. For the first one, FedAvg is configured as the aggregation algorithm to be used in every round, while Fed+ is used in the second one. As a reminder, a smaller ϵ provides a better privacy-preserving scenario. The evaluation also compares the achieved accuracy when using FL without applying DP techniques.

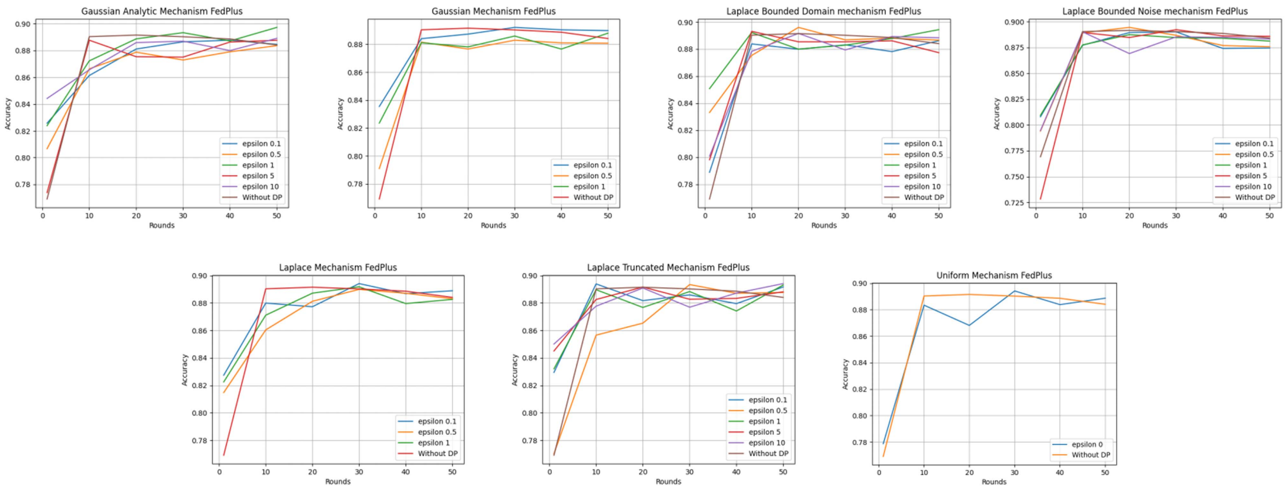

Fig. 3. Fed+ accuracy results for all perturbation mechanisms.

For the Gaussian analytic mechanism, ϵ can take values higher than 1. Fig. 2 shows that the accuracy levels for all ϵ values are higher than 0.8, and that we achieve similar results to the configuration without DP, meaning that our framework almost does not impact the accuracy while providing more privacy than classical FL approaches. Moreover, Gaussian mechanism, as shown in Fig. 2, reaches close accuracy values for every ϵ value, even with the ones using the non-DP configuration. In general, in both cases, it is noticeable an increase in the first ten rounds in all cases, and then the accuracy stabilizes.

Furthermore, Fig. 2 shows the same information for Laplace bounded domain and Laplace bounded noise mechanisms. In this case, as in the previous mechanism, the accuracy values for different ϵ values are very close, even if the accuracy level is reduced at round 30 for Laplace bounded noise and round 20 for Laplace bounded domain for every ϵ and the non-DP configuration. In the case of the uniform mechanism, it should be noted that the accuracy value is clearly reduced throughout the rounds.

Moreover, Fig. 3 shows the results by using Fed+ as the aggregation algorithm. For this case, the evolution of the accuracy value is clearly ascending until round ten for every ϵ in all mechanisms. Compared with FedAvg, the final accuracy levels at the last round are slightly better, except for the uniform mechanism, which has a clearly higher accuracy level compared with FedAvg, since Fed+ behaves better with non-i.i.d. data, which is our case. According to Fig. 3, in most of the cases, the accuracy using our DP strategy outperforms the accuracy of the scenario without any perturbation. This could be explained by the fact that running stochastic gradient descent with noisy gradient estimates can help in the performance of the model [17], while the noisy data are above some threshold. Also, this phenomenon has previously been observed in other state-of-the-art studies, which apply DP for FL in a more limited way compared to us but yet obtain better accuracy when obfuscating data in certain cases [18].

Theoretically, the lower the ϵ value, since the privacy factor increases, the final model should reach lower accuracy values as the weights are more obfuscated. However, as can be seen in Figs. 2 and 3, for almost all perturbation mechanisms and both aggregation algorithms, there is no big difference between the accuracy reached by a mechanism using the most restrictive ϵ value (lower) and using the most relaxed one (higher). Nevertheless, using FedAvg and some mechanisms, such as Laplace truncated or Gaussian, it is clear that the more restrictive the ϵ is, the lower the accuracy throughout the rounds.

Moreover, it should also be noted that the privacy enhancement achieved by each mechanism is given by the distance to a PCC value of 1, which would represent the classical FL scenario where no perturbation mechanism is applied to data, and therefore, there is no obfuscation at all. Table IV gives the average PCC values for each perturbation mechanism and different ϵ values. It should be noted that, for all mechanisms, the lower the ϵ value, the lower the PCC. This indicates that, actually, lower ϵ values achieve a more obfuscated set of weights and, therefore, a higher privacy factor. As it can be seen, uniform is the mechanism that provides a more obfuscated set of weights, and consequently, the lowest PCC. Therefore, this mechanism provides the highest privacy factor compared with the other DP techniques analyzed in this article. Furthermore, the uniform mechanism also provides a similar accuracy level compared with the other DP techniques when using Fed+ as shown in Fig. 3. Consequently, it provides the best results considering the values of PCC and accuracy for the proposed scenario.

By last, Fig. 4 shows a box–plot graphic of execution times for every mechanism. This graphic is the result of ten different executions per mechanism for the same epsilon ($\epsilon = 1$), except for the uniform mechanism since it only accepts $\epsilon = 0$. For each execution, the time spent in the perturbation process is measured, and then the maximum, minimum, and average times are shown in each box in the graphic.

VI. Related Work

As discussed previously, despite the advantages provided by FL about privacy aspects, recent works demonstrated that different attacks (e.g., inference) are still possible during the federated training process through the access to the gradients/weights,

TABLE IV
PEARSON CORRELATION COEFFICIENT

ϵ	Mechanism	Pearson Correlation	Mechanism	Pearson Correlation
0.1	Laplace Truncated	0.6560	Laplace Bounded Domain	0.4552
0.5	Laplace Truncated	0.9663	Laplace Bounded Domain	0.9028
1	Laplace Truncated	0.9906	Laplace Bounded Domain	0.9762
5	Laplace Truncated	0.9996	Laplace Bounded Domain	0.9994
10	Laplace Truncated	0.9999	Laplace Bounded Domain	0.9998
0.1	Laplace	0.6572	Gaussian	0.3549
0.5	Laplace	0.9655	Gaussian	0.8256
1	Laplace	0.9907	Gaussian	0.9406
0.1	Gaussian Analytic	0.5964	Laplace Bounded Noise	0.6951
0.5	Gaussian Analytic	0.9156	Laplace Bounded Noise	0.9666
1	Gaussian Analytic	0.9704	Laplace Bounded Noise	0.9908
5	Gaussian Analytic	0.9977	Laplace Bounded Noise	0.9996
10	Gaussian Analytic	0.9992	Laplace Bounded Noise	0.9999
0	Uniform	0.0472		

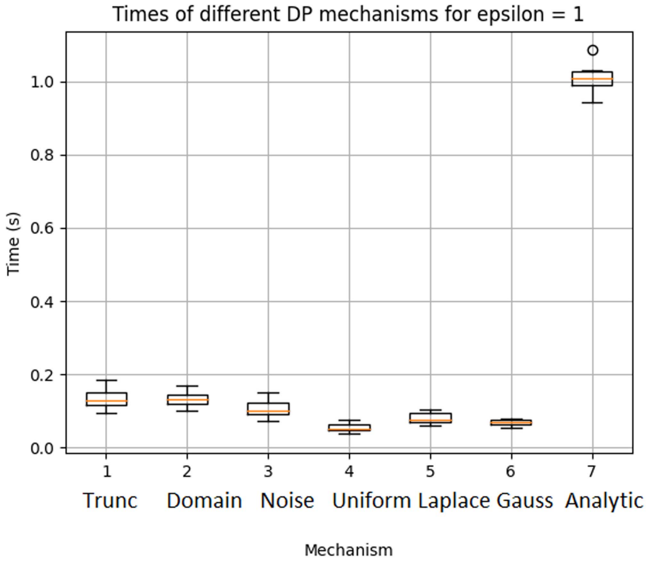

Fig. 4. DP execution times per mechanism.

which are uploaded by the FL clients to the coordinator [5]. Therefore, as mentioned in Section II, recent works have proposed the application of different privacy-preserving techniques, such as SMC and DP, to FL scenarios. In particular, a partial evaluation of DP techniques was carried out in [7], which implements the PrivacyFL simulator. Furthermore, Truex *et al.* [19] integrated SMC and DP to tackle inference attacks while maintaining an acceptable level of accuracy. The resulting approach is applied to several ML models, namely, decision trees, convolutional neural networks (CNN), and support-vector machine. However, these works do not compare the impact of different DP techniques considering their application to IDS approaches for IoT scenarios.

In the context of IoT, Briggs *et al.* [8] described the use of privacy-preserving techniques for FL, which also provides a set of challenges for resource-constrained scenarios. Hu *et al.* [20] addressed the application of DP for FL in IoT, which uses activity recognition data from smartphones and personalized models on each device. Also based on the use of DP, Lu *et al.* [21] made use of blockchain so that the computation required for the consensus mechanism is also used for the federated training process. However, they did not provide evaluation results related to the DP technique being employed. Furthermore, Zhao *et al.* [22] assessed the impact on the accuracy of applying Gaussian noise as a DP technique in an FL environment. However, the evaluation is based on the well-known MNIST dataset, and they did not compare these results with other DP techniques. In addition, Hu *et al.* [23] considered an IoT scenario with resource constraints, where a relaxed version of DP is applied and evaluated considering different datasets.

While previous approaches demonstrate the interest on the application of privacy-preserving techniques for FL, these aspects are not typically considered in the context of anomaly/intrusion detection. For example, Li *et al.* [24] proposed a fog architecture for DDoS attack detection and mitigation using FL. However, only DoS attacks are considered, and privacy techniques are not integrated. Furthermore, Liu *et al.* [25] used CNN for anomaly detection in IIoT based on a gradient compression mechanism to reduce the communication overhead in FL. In addition, the use of FL was also considered by Khoa *et al.* [26] to implement an intrusion detection approach in Industry 4.0 scenarios. They compared their solution using different ML techniques and datasets, but privacy aspects were not considered. Furthermore, Attota *et al.* [27] used an ensemble approach and an optimization algorithm for feature extraction to come up with an IDS approach for IoT. For validation purposes, they used a dataset containing traffic of a single IoT protocol. Moreover, recently a federated version of CNN was used by Man *et al.* [28] for intrusion detection in IoT that is intended to reduce communication overhead. Other works, such as [29], are based on old industrial datasets that do not consider recent attacks from IIoT scenarios. However, like in the previous cases, privacy-preserving techniques are not integrated. More related to our proposal, Chathoth *et al.* [30] recently proposed two DP-based continuous learning methods that consider heterogeneous privacy requirements for different FL clients in an IDS system. However, the approach is based on a non-IoT-specific dataset (CSE-CIC-IDS2018), and different DP techniques are not compared. In contrast to this approach, our work evaluates different DP techniques that are applied to the recent Ton_IoT dataset, which contains different types of attacks from IIoT environments, including network and sensor data manipulation attacks. To the best of our knowledge, this is the first effort that provides a comprehensive evaluation on the application of DP techniques for FL considering different aggregation techniques, in order to foster the development of a privacy-preserving FL-enabled IDS for IIoT.

VII. Conclusion

The development of ML-enabled IDS approaches was based on the processing of devices' network traffic to detect potential attacks and threats. While FL was coined to avoid parties to share their data, it still suffers from privacy issues associated with the communication of gradients/weights in each training round. To address such issue, this work provided an exhaustive evaluation on the use of DP techniques based on additive noise mechanisms, which are applied during the federated training process of the ToN_IoT dataset to come up with a privacy-preserving IDS for IIoT scenarios. We compared different noise addition techniques based on Gaussian and Laplacian distributions, and assessed the accuracy obtained using Fed+ as an alternative aggregation function to FedAvg that has recently been proposed to deal with non-i.i.d. data distributions, which are prevalent in the real world. According to our evaluation results, the use of such DP techniques maintains an acceptable level of accuracy, which is even close to a non-DP scenario in the case of low privacy requirements (i.e., with a high ϵ value). In the case of Fed+, the impact of DP techniques on the accuracy is not perceptible. To the best of our knowledge, this work represents the first effort to provide a comprehensive evaluation of an FL-enabled IDS in IIoT considering different aggregation functions. As future work, we will analyze the development of a personalized FL approach where each device has different privacy requirements, as well as the use of gradient compression techniques to be considered on IIoT scenarios with network constraints.

References

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[2] N. Garcia, T. Alcaniz, A. González-Vidal, J. B. Bernabe, D. Rivera, and A. Skarmeta, "Distributed real-time SlowDoS attacks detection over encrypted traffic using artificial intelligence," *J. Netw. Comput. Appl.*, vol. 173, 2021, Art. no. 102871.

[3] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: A survey and review," 2014. [Online]. Available: https://arxiv.org/abs/1412.7584

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[5] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, 2021.

[6] O. Choudhury *et al.*, "Anonymizing data for privacy-preserving federated learning," 2020. [Online]. Available: https://arxiv.org/abs/2002.09096

[7] V. Mugunthan, A. Peraire-Bueno, and L. Kagal, "PrivacyFL: A simulator for privacy-preserving and secure federated learning," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 3085–3092.

[8] C. Briggs, Z. Fan, and P. Andras, "A review of privacy-preserving federated learning for the Internet-of-Things," in *Federated Learning Systems*. Cham, Switzerland: Springer, 2021, pp. 21–50.

[9] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. den Hartog, "Ton_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion datasets," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2021.3085194.

[10] P. Yu, L. Wynter, and S. H. Lim, "Fed+: A family of fusion algorithms for federated learning," 2020. [Online]. Available: https://arxiv.org/abs/2009.06303

[11] Q. Geng, W. Ding, R. Guo, and S. Kumar, "Privacy and utility tradeoff in approximate differential privacy," 2018. [Online]. Available: https://arxiv.org/abs/1810.00877

[12] B. Balle and Y.-X. Wang, "Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 394–403.

[13] N. Holohan, S. Antonatos, S. Braghin, and P. M. Aonghusa, "The bounded Laplace mechanism in differential privacy," 2018. [Online]. Available: https://arxiv.org/abs/1808.10410

[14] Y. Dagan and G. Kur, "A bounded-noise mechanism for differential privacy," 2020. [Online]. Available: https://arxiv.org/abs/2012.03817

[15] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," 2019. [Online]. Available: https://arxiv.org/abs/1907.02189

[16] D. Böhning, "Multinomial logistic regression algorithm," *Ann. Inst. Stat. Math.*, vol. 44, no. 1, pp. 197–200, 1992.

[17] S. Song, K. Chaudhuri, and A. Sarwate, "Learning from data with heterogeneous noise using SGD," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, 2015, pp. 894–902.

[18] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, Apr. 2020.

[19] S. Truex *et al.*, "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, 2019, pp. 1–11.

[20] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9530–9539, Oct. 2020.

[21] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[22] B. Zhao, K. Fan, K. Yang, Z. Wang, H. Li, and Y. Yang, "Anonymous and privacy-preserving federated learning with industrial Big Data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6314–6323, Sep. 2021.

[23] R. Hu, Y. Guo, E. P. Ratazzi, and Y. Gong, "Differentially private federated learning for resource-constrained Internet of Things," 2020. [Online]. Available: https://arxiv.org/abs/2003.12705

[24] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lv, "FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2021.3088938.

[25] Y. Liu *et al.*, "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.

[26] T. V. Khoa *et al.*, "Collaborative learning model for cyberattack detection systems in IoT Industry 4.0," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2020, pp. 1–6.

[27] D. C. Attota, V. Mothukuri, R. M. Parizi, and S. Pouriyeh, "An ensemble multi-view federated learning intrusion detection for IoT," *IEEE Access*, vol. 9, pp. 117734–117745, 2021.

[28] D. Man, F. Zeng, W. Yang, M. Yu, J. Lv, and Y. Wang, "Intelligent intrusion detection based on federated learning for edge-assisted Internet of Things," *Secur. Commun. Netw.*, vol. 2021, 2021, Art. no. 9361348.

[29] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2021.3077803.

[30] A. K. Chathoth, A. Jagannatha, and S. Lee, "Federated intrusion detection for IoT with heterogeneous cohort privacy," 2021. [Online]. Available: https://arxiv.org/abs/2101.09878

Pedro Ruzafa-Alcázar received the degree in computer science from the University of Murcia, Murcia, Spain, in 2021.

He became a scholarship Researcher in 2020. His research interests include cyber security, network architecture, and telematics services deployment.

Pablo Fernández-Saura received the B.Eng. and M.Sc. degrees in computer science from the University of Murcia, Murcia, Spain, in 2020 and 2021 respectively.

He was a Researcher in several European projects, such as H2020 CyberSec4Europe and H2020 Inspire-5Gplus, with the University of Murcia. His research interests include cybersecurity, artificial intelligence, and 5G networks.

Enrique Mármol-Campos received the graduate degree in mathematics and the M.S. degree in advanced math, with the specialty of operative research and statistics, in 2018 and 2019, respectively, from the University of Murcia, Murcia, Spain, where he is currently working toward the Ph.D. degree in federated learning, cybersecurity, IoT, and deep learning.

His research interests include the application of federated learning in the cybersecurity of Internet of Things and Internet of Vehicles devices, finding new ways to optimize the resources, securing the privacy of all parts involved, and the search of its use in real-time scenarios where these devices presents lots of restrictions.

Aurora González-Vidal received the graduate degree in mathematics and the Ph.D. degree in computer science from the University of Murcia, Murcia, Spain, in 2014 and 2019 respectively.

In 2015, she received a fellowship to work with the Statistical Division of the Research Support Service, where she specialized in statistics and data analysis. Afterward, she studied a Big Data Master. She is currently a Postdoctoral Researcher with the University of Murcia. She has collaborated in several national and European projects, such as ENTROPY, IoTCrawler, and DEMETER. Her research interests include machine learning in IoT-based environments, missing values imputation, and time-series segmentation.

Dr. Vidal is the President of the R Users Association UMUR.

José L. Hernández-Ramos received the Ph.D. degree in computer science from the University of Murcia, Murcia, Spain, in 2016.

He is currently a Scientific Project Officer with Joint Research Centre, European Commission, Ispra, Italy. He has participated in different European research projects, such as SocIoTal, SMARTIE, and SerIoT. He was a Scientific Expert for the Agence Nationale de la Recherche, a TPC Member/Co-Chair of several conferences, and the Guest Editor and Reviewer of different journals. He has authored or coauthored more than 60 peer-reviewed papers. His research interests include application of security and privacy mechanisms in the Internet of Things and transport systems scenarios, including blockchain and machine learning.

Jorge Bernal-Bernabe received the B.S. degree in computer science, in 2007, M.S. degree in telematics, in 2008, M.B.A. degree in 2009 and Ph.D. degree in computer science, from the University of Murcia, Murcia, Spain, in 2015.

He was a Visiting Researcher with the Hewlett-Packard Laboratories Palo Alto, CA, USA and University of the West of Scotland, Glasgow, U.K. He has authored more than 40 high-impact indexed journal papers and numerous conferences. During the last years, he was working in several European research projects, such as SocIoTal, ARIES, OLYMPUS, ANASTACIA, INSPIRE-5G, and CyberSec4EU. He is currently an Associate Professor with the University of Murcia. His research interests include security, trust, and privacy management in distributed systems and IoT.

Antonio F. Skarmeta (Member, IEEE) received the Ph.D. degree in computer science from the University of Murcia, Murcia, Spain, in 1995.

He is currently a Full Professor with the Department of Information and Communications Engineering, University of Murcia. He has authored or coauthored more than 200 international papers. He was a Member of several program committees. His research interests include integration of security services, identity, the Internet of Things, and smart cities.