Dynamic Federated Learning Model for Identifying Adversarial Clients

Nuria Rodríguez-Barroso[a], Eugenio Martínez-Cámara[a,*], M. Victoria Luzón[a], Gerardo González Seco[b], Miguel Ángel Veganzones[b], Francisco Herrera[a]

***Abstract*—Federated learning, as a distributed learning that conducts the training on the local devices without accessing to the training data, is vulnerable to dirty-label data poisoning adversarial attacks. We claim that the federated learning model has to avoid those kind of adversarial attacks through filtering out the clients that manipulate the local data. We propose a dynamic federated learning model that dynamically discards those adversarial clients, which allows to prevent the corruption of the global learning model. We evaluate the dynamic discarding of adversarial clients deploying a deep learning classification model in a federated learning setting, and using the EMNIST Digits and Fashion MNIST image classification datasets. Likewise, we analyse the capacity of detecting clients with poor data distribution and reducing the number of rounds of learning by selecting the clients to aggregate. The results show that the dynamic selection of the clients to aggregate enhances the performance of the global learning model, discards the adversarial and poor clients and reduces the rounds of learning.**

***Index Terms*—Federated Learning, Deep Learning, adversarial attacks, dynamic aggregation operator**

I. Introduction

THE standard machine learning approach is built upon an algorithm that learns from a centralised data source. Distributed machine learning proposes the distribution of the data and elements of a learning model among several nodes as a solution for the unceasing growing of learning model complexity and the size of training data [1, 2]. However, the distributed machine learning solution is neither valid for the data privacy challenge, nor for an scenario with a large number of clients and a non homogeneous data distribution [3].

Federated learning (FL) is a nascent machine learning approach in which the algorithms learn from sequestered data [4]. FL is mainly composed of two components: a global server that owns the global learning model and a set of clients storing the local learning models and the local training datasets. Likewise, FL consists in: (1) training the local learning models in each data source, (2) distilling the parameters of the local learning models into a central server, (3) aggregating the parameters of the local models in a federated learning model and (4) updating the local learning models with the aggregated federated model after the aggregation. This specific setting supports its main feature, which is the prevention of data leakage and the protection of data privacy, because the data do not abandon its local storage and they are not shared with any other client or third party.

We know that machine learning is vulnerable to malicious manipulations on the input data to cause incorrect classification [5]. This vulnerability to adversarial attacks is higher in FL because it does not have access to the training data [6]. Among the different kind of adversarial attacks in the literature [7], in this paper we focus on poisoning attacks [8], which are based on the arbitrary manipulation of the training data, and specifically on the data poisoning attack [9, 10]. These adversarial attacks in FL are conducted by the clients because they own the data, therefore the defence has to be performed by the FL model in the global server.

We claim in this paper that the FL model has to be able to dynamically avoid adversarial clients to preserve the learning model from data poisoning attacks. In the literature there are a number of federated aggregation operators, but they do not prevent the federated model from this kind of attack [11, 12, 13], or they do it following some assumptions about the nature of the adversarial clients [14].

We propose a dynamic FL model that dynamically selects the clients to be aggregated and discards the adversarial ones. The proposed model is agnostic about the number and nature of the adversarial clients. The dynamic FL model is built upon an Induced Ordered Weighted Averaging (IOWA) operator [15], which weights the contribution of each client in the aggregation, and it is guided by a dynamic linguistic quantifier. We call this new FL model FL-IOWA-DQ.

We deploy in a FL setting an image deep learning classification model for evaluating FL-IOWA-DQ. We leverage the benchmark image classification datasets EMNIST[1] Digits [16] and Fashion MNIST[2] [17], and we distribute the data over the clients following a non independent and identically distributed (non-IID) distribution. We show that the FL-IOWA-DQ model is able to identify the adversarial and poor clients, filter them out and enhance the performance of the global learning model.

In addition, one of the major handicaps of FL is the time spent in communications between the server and the clients [18, 19]. We show that the dynamic selection of clients of our dynamic aggregation operator also allows to improve the learning of the federated model, which results in (1) reducing

[a] Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[b] Sherpa.ai, Spain
* Corresponding author
Email addresses: rbnuria@ugr.es (Nuria Rodríguez-Barroso), emcamara@decsai.ugr.es (Eugenio Martínez-Cámara), luzon@ugr.es (M. Victoria Luzón), g.gonzalez@sherpa.ai (Gerardo González-Seco), ma.veganzones@sherpa.ai (Miguel Ángel Veganzones), herrera@decsai.ugr.es (Francisco Herrera)

[1] https://www.nist.gov/node/1298471/emnist-dataset
[2] https://github.com/zalandoresearch/fashion-mnist

the number of rounds of learning, (2) lessening the number of model updates among the central federated model and the clients, and hence (3) diminishing the time spent in communications between the server and the clients.

The rest of the work is organised as follows: the following section summarises the background related to FL, federated aggregation operators and adversarial attacks in FL. Section III is focused on the description of the dynamic FL model for identifying adversarial clients. We detail the experimental set-up in Section IV and evaluate and analyse the results of the FL models in Section V. Finally, conclusions are described in Section VI.

II. Background

We propose in this paper a novel FL model built upon a federated aggregation operator with the capacity of dynamically discarding adversarial clients. Accordingly, we introduce in this section some relevant concepts and related works. We define FL in Section II-A, we describe the main federated aggregation operators in the literature in Section II-B, and we introduce the main types of adversarial attacks in Section II-C.

A. Federated Learning

FL is a nascent learning approach pushed by the need of overcoming the limitations of distributed learning for preserving data privacy and for processing large number of clients following a non homogeneous data distribution [3]. FL proposes a new training approach of learning algorithms that consists in the iterative training of the model in the devices that own the data, the aggregation of those models in the federated model, and the updating of the local models with the federated model. Accordingly, FL prevents from data leakage and preserves data privacy, because the data do not leave the electronic device.

Formally, FL is a distributed machine learning paradigm consisting of a set of clients $\{C_1, \ldots, C_n\}$ with their respective local training data $\{D_1, \ldots, D_n\}$. Each of these clients C_i has a local learning model named as LLM_i represented by the parameters $\{\Theta_i, \ldots, \Theta_n\}$. FL aims at learning the global learning model represented by Θ, using the scattered data across clients through an iterative learning process known as *round of learning*. For that purpose, in each round of learning t, each client trains its local learning model over their local training data D_i^t which updates the local parameters Θ_i^t to $\hat{\Theta}_i^t$. Subsequently, the global parameters Θ are computed aggregating the trained local parameters $\{\hat{\Theta}_1^t, \ldots, \hat{\Theta}_n^t\}$ using an specific federated aggregation operator Δ:

$$\Theta^t = \Delta(\hat{\Theta}_1^t, \hat{\Theta}_2^t, \ldots, \hat{\Theta}_n^t) \tag{1}$$

After the aggregation of the parameters in the global learning model, the local learning models are updated with the aggregated parameters:

$$\Theta_i^{t+1} \leftarrow \Theta^t, \quad \forall i \in \{1, \ldots, n\} \tag{2}$$

The updates among the clients and the server are repeated as much as needed for the learning process. Thus, the final value of Θ will sum up the knowledge sequestered in the clients.

As a new learning paradigm, there are some works that attempt to adapt standard learning techniques to a FL environment. Multi-task learning is one of the possible applications of FL, where each electronic device can perform a different task, and its first adaptation to FL is described in [20]. Recommendation systems are also susceptible to be implemented in a FL environment as is described in [21]. Domain adaptation is related to concept of model generalisation, which is also consider in [22] for FL. Likewise, in [23] the authors state the security challenges that have to face up FL.

B. Related works about federated aggregation operators

One of the main elements of FL is the federated aggregation operator, which has to: (1) assure a right aggregation of the local models in order to optimise the learning process; (2) reduce the number of communication rounds among the clients and the federated server and (3) be robust against malicious clients and clients with poor data.

There are some federated aggregation operators in the literature:

- FedAvg [11] builds the federated model by averaging the parameters of the local models. FedAvg was used for improving the prediction of terms [24], recognising out-of-vocabulary words [25] and for generating personalised language models [26]. There is a weighted version which consists of weighting each client according to the proportion of data to the total data population it owns.
- Federated Stochastic Variance Reduced Gradient (FSVRG) [12] is a modification of FedAvg to work with sparse data. It consists on computing the gradient of each client step by step using a full gradient previously computed by averaging the full gradient of each client.
- CO-OP [13] is designed for asynchronous model updates in contrast to the previous two operators. It merges any received client model with the global model. Instead of directly averaging the models, the merging between a local model and the global model is carried out using a weighting scheme based on a measure of the difference in the age of the models. This is motivated by the fact that in an asynchronous framework, some clients will be trained on obsolete models while others will be trained on more up-to-date models.
- Bayesian learning may be also used for aggregating the local models in the global server [27].
- The adaptation of some machine learning algorithms requires some ad-hoc procedures. For instance, an ad-hoc federated aggregation operator for the federated version of the k-means algorithm is described in [28].

The performance of the FL model depends on the federated aggregation operator, because it aggregates the local learning models from the clients. Likewise, since it works with the parameters of the local learning models, it has to safeguard the FL from adversarial attacks.

C. Related works about adversarial attacks

FL is highly vulnerable to attacks against the learning models, because of its distributed scheme in different nodes.

The main difference with classical attacks on both centralised and distributed training models is that the FL is susceptible to receive attacks during training process due to the use of private and uninspected data. Therefore, FL emphasises the impact of some classic distributed learning attacks and makes it more challenging to prevent them.

Adversarial attacks can be carried out by the server and the clients, since they know the parameters of the learning process at some stage of the rounds of learning, or a third party that accesses the model parameters shared during the communication between server and clients. According to Kairouz et al. [4], there are three main types of attacks:

1 *Model update Poisoning Attacks* [29, 30, 31]: they are characterised by the corruption of some client model's update either by the client itself, a man-in-the-middle attack or the server. For generalisation, we assume that the adversary (or adversaries) directly controls a certain number of clients, and they can directly change the outputs of these clients to try to skew the global model towards their goal. There are two types of poissoning attacks:
   - *Untargeted attacks*: they can alter the outputs of the FL model. The Byzantine adversarial attacks represent the worst-case scenario, as they can make a FL model produce any arbitrary outputs [32]. One of the most widely used solutions proposed for this type of attack is to employ more robust aggregation operators such as median-based ones [33] or use data shuffling and redundancy [34]. However, this defensive techniques are insufficient against model update poisoning attacks in FL [35].
   - *Targeted attacks:* they focus on a specific target of the adversary such as introducing a backdoor into the model [36]. The major challenge in dealing with these attacks is that the poisoned models are often very similar to the rest of models. In addition, they usually preserve the overall measures as they affect only specific cases, which leaves the untargeted attack defences ineffective. Existing defences against backdoor attacks [37] can not be applied in federated environments as they require to access to the data, which is not possible in FL and hence and open challenge.
2 *Data Poisoning Attacks*: they are defined as a more restrictive attack in which the opponent can only modify the client's local data by manipulating either the labels or some features of the data. As model attacks, they are also subdivided in untargeted [**?**] and targeted [14, 38, 39] ones. Since a data poisoning attack causes a model update poisoning, defence mechanisms against Byzantine attacks can be applied.
3 *Evasion Attacks*: they consist of manipulating a deployed model by modifying the samples fed into it at test time [40]. Because of the definition of the attack, white-box model attacks are more natural and FL increases the demand for defences against them.

Additionally, differential privacy [41] tools are an important safeguard for the information shared during the communication between the server and the clients. Therefore, the defensive challenges of the FL should focus on client attacks.

We propose in this paper a defence mechanism against data poisoning attacks by means of a FL model, which dynamically selects the clients that are not adversarial and filters out the adversarial ones.

III. Dynamic FL model for identifying adversarial clients

FL is featured by its restriction to access to the training data, which is sequestered in the clients. Accordingly, data poisoning attacks, and more specifically dirty-label data poisoning attacks [9, 10], grounded in the malicious manipulation of the training data, can corrupt the FL model, which cannot inspect the training to defend itself against this kind of adversarial attacks.

The dirty-label poisoning adversarial attack is conducted by the clients in a FL setting, and it consists in making subtle modifications in the labels of the training data for corrupting the learning process. This attack can be simulated by arbitrarily assigning labels to all the training samples of a subset of the federated clients. We define a dirty-label poisoning adversarial client in Definition 3.1.

Definition 3.1 (Adversarial client): Let $C_i \in \{C_1, ..., C_n\}$ be an arbitrary client of a FL environment whose original training dataset is $D_i = \langle x_i^l; y_i^l \rangle$, where x_i^l is the sample data and y_i^l the label. We say that C_i is an **adversarial client** if it uses the altered dataset D_i' as training dataset with

$$D_i' = \langle x_i^l; y_i^{\sigma(l)} \rangle,$$

where σ is a random permutation.

Regarding the limitation of FL to inspect the training data for discovering adversarial clients, we propose a dynamic FL model that dynamically selects the clients to be aggregated, and filters out the adversarial ones. The dynamic FL model is built upon a federated aggregation operator based on a Induced Ordered Weighted Averaging (IOWA) operator [15], and we call it FL-IOWA-DQ.

The IOWA operators, and more generally the Ordered Weighted Averaging (OWA) ones [42], are functions for weighting the contribution of a set of clients in a aggregation process, as it is the aggregation of the parameters of the local learning models in FL. We mathematically introduce OWA and IOWA operators in Appendix A, and according to the definition the IOWA operator is composed of (1) an order-inducing function to set the weighting assignation order, and (2) a linguistic quantifier to calculate the weight contribution value. We define the induced-order function of the IOWA operator of the FL-IOWA-DQ model in Section III-A, and the linguistic quantifier that dynamically adapts the weighting value calculation during the FL training in Section III-B.

A. Accuracy-based induced ordering function for FL clients

The aim of dirty-label data poisoning adversarial attacks is hindering the performance of a FL model through altering the labels of the training data. Since FL is grounded in the aggregation of the LLM_i, those maliciously altered ones would perform lower than the non-altered ones. Hence, the validation

of the LLM_i before the aggregation may help to identify the suspicious adversarial clients.

We propose the Local Accuracy Function, f_{LA}, to measure the performance of each LLM_i before its aggregation. The f_{LA} function is based on the availability of a validation set shared among the clients. The creation of this validation set is justified by its reduced size compared to the size required for training, and the possibility of making it up through expert or prior knowledge. We define the f_{LA} function in Definition 3.2.

Definition 3.2 (Local Accuracy Function (f_{LA})): it measures the performance of a local learning model LLM_i using a fixed validation dataset named as VD. For that, it computes the *accuracy* of LLM_i over VD:

$$f_{LA}(LLM_i) = \text{accuracy}(LLM_i, VD) \tag{3}$$

where accuracy(LLM_i, VD) refers to the standard accuracy evaluation measure of the local learning model LLM_i in the dataset VD.

B. Dynamic linguistic quantifier for weighting FL clients

The non-IID data distribution of most of the FL settings make impossible to know beforehand the nature of the clients, and hence it is impossible to know the amount of adversarial clients. Therefore, the selection of the FL clients by its weighted contribution has to be dynamically calculated for adapting to the nature of the clients.

The dynamic selection of the FL-IOWA-DQ model is based on a IOWA linguistic quantifier that some of its parameters values depend on the value of f_{LA}. Before the definition of the linguistic quantifier of FL-IOWA-DQ, we first define the IOWA linguistic quantifier in Definition 3.3.

Definition 3.3 (Linguistic quantifier): It is a function $Q : [0,1] \to [0,1]$ verifying $Q(0) = 0$, $Q(1) = 1$ and $Q(x) \geq Q(y)$ for $x > y$. Equation 4 defines how the function Q computes the weighting values where w_i represents the weighting associated to the position i of a vector of dimension n, and Equation 5 defines the behaviour of the function Q.

$$w_i^{(a,b)} = Q_{a,b}\left(\frac{i}{n}\right) - Q_{a,b}\left(\frac{i-1}{n}\right) \tag{4}$$

$$Q_{a,b}(x) = \begin{cases} 0 & 0 \leq x \leq a \\ \dfrac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq 1 \end{cases} \tag{5}$$

where $a, b \in [0,1]$ satisfying $0 \leq a \leq b \leq 1$, and they set the intervals for calculating the contribution weight of each LLM_i. For the sake of clarification, those x values in the same interval will have the same weighting value.

We redefine the function $Q_{a,b}$ for providing it a dynamic behaviour and a higher weighting of top clients, which depends on the f_{LA} function. Accordingly, we propose Q_{a,b,c,y_b} that is defined in Equations 6, and incorporates two new parameters to the model:

1 Parameter y_b. The higher weighting of top clients using the pair (b, y_b) with $b, y_b \in [0,1]$ where b is the the portion of clients we want to weight higher and y_b the portion of the total weight assigned to these clients. For example, for the pair $(b = 0.2, y_b = 0.4)$ we distribute the 40% of the weight among the top 20% of the best clients.

2 The dynamic behaviour of the parameter c. This parameter represents the portion of clients that we do not discard. For example, a value of $c = 0.8$ means that the 20% of the clients will be discarded. With the aim of dynamically adapt it, we first calculate $\hat{c}$, which is the portion of clients which differs from the top-1 in more than $3/4$ of the maximum distance between clients according to f_{LA}, and set $c = 1 - \hat{c}$. This way, we dynamically discard the clients with the worst performance in terms of f_{LA}. The remaining weight not assigned to top clients is distributed among the clients not discarded.

$$Q_{a,b,c,y_b}(x) = \begin{cases} 0 & 0 \leq x \leq a \\ \dfrac{x-a}{b-a} \cdot y_b & a \leq x \leq b \\ \dfrac{x-b}{c-b} \cdot (1-y_b) + y_b & b \leq x \leq c \\ 1 & c \leq x \leq 1 \end{cases} \tag{6}$$

$$w_i^{(a,b,c,y_b)} = Q_{a,b,c,y_b}\left(\frac{i}{n}\right) - Q_{a,b,c,y_b}\left(\frac{i-1}{n}\right) \tag{7}$$

Based on the quantifier Q_{a,b,c,y_b} and the ordering function f_{LA}, we define the aggregation operator of FL-IOWA-DQ as:

Definition 3.4 (IOWA Dynamic Quantifier federated aggregation operator (IOWA-DQ)): The operator of dimension n is a mapping $\Psi : ([0,1] \times \Omega)^n \to \Omega$ with an associated set of weights $W = w_i^{(a,b,c,y_b)}$, and it is defined to aggregate the second arguments of a 2-tuple n list according to the following expression:

$$\Psi_{IOWA-DQ}(\langle u_1, \Theta_1\rangle, \ldots, \langle u_n, \Theta_n\rangle) = \sum_{i=1}^{n} w_i^{(a,b',c,y_b)} \Theta_{\sigma(i)} \tag{8}$$

where $0 \leq a \leq b \leq c \leq 1$, $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ is a permutation function, such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall i = \{1, \ldots, n-1\}$, $u_i = f_{LA}(LLM_i)$, b' is the dynamic adaptation of b ($b' = b \times c$), and c is the portion of clients verifying that the distance to the highest u_i obtained is less than $3/4$ the distance from the lowest u_i to the highest u_i. Formally, c is the portion of clients C_i that verify

$$dist(u_{\sigma(1)}, u_{\sigma(i)}) \leq \frac{3}{4} \max_{i,j} dist(u_{\sigma(i)}, u_{\sigma(j)}) \tag{9}$$

Since we apply it to the aggregation of FL models, Θ_i represents the model's parameters of the client i but it could be applied in other contexts.

IV. EXPERIMENTAL SET-UP

We evaluate the FL-IOWA-DQ model through setting up a FL environment with the Sherpa.ai FL framework[3] [43].

[3]https://github.com/sherpaai/Sherpa.ai-Federated-Learning-Framework

Likewise we compare it with the static versions of FL-IOWA-DQ and other federated aggregation operators from the state of the art such as FedAvg [11].

The evaluation of FL-IOWA-DQ is performed in two datasets arranged for FL, and we describe them in Section IV-A. Also, we deployed an image classification deep learning model in the FL setting, which is described in Section IV-B. Section IV-C specifies the different configurations of the dynamic FL model. Finally, the federated aggregation operators used as baselines are introduced in Section IV-D.

A. Evaluation datasets

We use two different datasets for the evaluation of the FL-IOWA-DQ model. Since the FL-IOWA-DQ model needs a validation set for dynamically discarding adversarial clients, we create it from the training subsets of the two datasets, which follows the same distribution of the training subsets and has the same size than the test subsets. The two datasets used in the evaluation are described as what follows:

1 The EMNIST (Extended Modified NIST) dataset, which was presented in 2017 in [16] as an extension of the MNIST dataset [44]. The EMNIST Digits class contains a balanced subset of the digits dataset containing 28,000 samples of each digit. The dataset consists of 280,000 samples, which 240,000 are training samples and 40,000 test samples.
2 The Fashion MNIST [17] aims to be a more challenging replacement for the original MNSIT dataset. It contains a balanced subset of the 10 different classes containing 7,000 samples of each class. Hence, the dataset consists of 70,000 samples, which 60,000 are training samples and 10,000 test samples.

In summary, the datasets, after appropriate modifications to prepare the validation sets, follow the data distributions shown in Table I.

TABLE I
SIZE OF THE TRAINING, VALIDATION AND TEST SETS OF EMNIST AND FASHION MNIST DATASETS.

	Training	Validation	Test
EMNIST	200,000	40,000	40,000
Fashion MNIST	50,000	10,000	10,000

With the aim of adapting the datasets to a federated environment, the training data is distributed among the clients following a non-IID distribution. Accordingly, we randomly assign instances of a reduced number of labels to each client simulating a scenario in which each client contains partial information.

B. Deep Learning model for image classification

We use a deep learning model as local learning models LLM_i and global learning model for FL with n clients. Accordingly, the matrix from the FL definition $\Theta \in \Omega = \mathbb{R}^n \times \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$ where d_1 is the number of layers of the neural network and d_2 the maximum dimension of layer weights. Since we evaluate our proposal on a image classification task, we developed an image classification deep learning model based on a convolutional neural network (CNN), which is similar to the proposed in [11].

The deep learning model is designed with two convolutional layers with kernel size 5×5 and output size of 32 and 64 units respectively. Both layers are followed by a max-pooling layer with a 2×2 filter and stride = 2. Then, we add a dense layer with 512 hidden units and ReLU as activation function. The output layer is a softmax layer. The training is driven by the cross-entropy function. Equation 10 sums up the deep learning classification model.

$$\begin{aligned}
\text{Softmax}(y_7) &= pred \\
\text{ReLU}(y^6_{7\times7\times64}) &= y_7 \\
\text{Dropout}(y^5_{7\times7\times64}, dr^2) &= y^6_{7\times7\times64} \\
\text{MaxPooling}(y^4_{14\times14\times64}) &= y^5_{7\times7\times64} \\
\text{CNN}_2(y^3_{14\times14\times32}) &= y^4_{14\times14\times64} \\
\text{Dropout}(y^2_{14\times14\times32}, dr^1) &= y^3_{14\times14\times32} \\
\text{MaxPooling}(y^1_{28\times28\times32}) &= y^2_{14\times14\times32} \\
\text{CNN}_1(D_{28\times28}) &= y^1_{28\times28\times32}
\end{aligned} \tag{10}$$

where dr^1 and dr^2 are the dropout rate of the first and second dropout layers respectively.

C. Dynamic FL models

The parametric and dynamic definition of the FL-IOWA-DQ model allows to use it with different configurations. We evaluate two different configurations according to the weighting of top clients by means of y_b. The parameters of FL-IOWA-DQ in both configuration are the following ones:

1 We set $a = 0$ in order to take into account all the best clients.
2 We fix $b = 0.2 \times c$ to assign the highest weight to the best 20% of the clients involved in the aggregation.
3 We use two different values for the top weighting parameter $y_b = \{0.4, 0.75\}$ resulting in two different configurations.

Based on these parameters, we define the two configurations of the FL-IOWA-DQ in Table II. Figure 1 shows the IOWA-DQ-0.4 linguistic quantifier as fuzzy sets.

TABLE II
VALUE OF THE PARAMETERS FOR EACH CONFIGURATION OF THE DYNAMIC FL MODELS BASED ON IOWA OPERATORS.

	a	b	c	y_b
FL-IOWA-DQ-0.4	0	0.2 × c	*dynamic*	0.4
FL-IOWA-DQ-0.75	0	0.2 × c	*dynamic*	0.75

D. FL baselines based on federated aggregation operators

We compare the FL models specified above with the following FL models as baselines. For all of them, we use the same deep learning models. Hence, the difference in behaviour

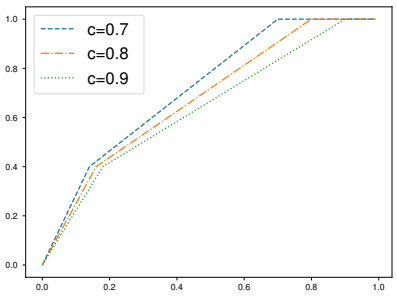

Fig. 1. Linguistic quantifier of IOWA-DQ-0.4.

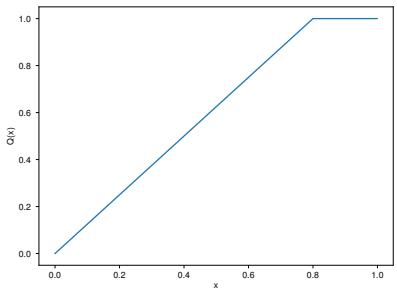

Fig. 2. Linguistic quantifier of AL-80.

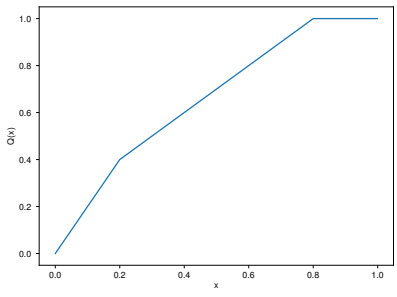

Fig. 3. Linguistic quantifier of IOWA-SQ-0.4.

will be due to the federated aggregation operator. Likewise, we define the aggregation operator and name the associated FL model as FL-[operator].

1) FL-FedAvg: The corresponding aggregation operator builds the federated model by means the average of local models. It is the most used federated aggregation operator in the literature. We formally define it in Definition 4.1.

Definition 4.1 (Federated Averaging (FedAvg) [11]): The FedAvg aggregation operator of dimension n is a mapping $FedAvg : \Omega^n \to \Omega$ defined to aggregate a list of parameters $\{\Theta_1, \ldots, \Theta_n\}$ according to the following expression:

$$FedAvg(\Theta_1, \ldots, \Theta_n) = \frac{\sum_i^n \Theta_i}{n} \tag{11}$$

2) FL-W-FedAvg: We also consider the weighted version of FL-FedAvg based on the amount of data for each client. We formally define it in Definition 4.2.

Definition 4.2 (Weighted Federated Averaging (W-FedAvg) [11]): The W-FedAvg aggregation operator of dimension n is a mapping $W - FedAvg : \Omega^n \to \Omega$ defined to aggregate a list of parameters $\{\Theta_1, \ldots, \Theta_n\}$ according to the following expression:

$$W - FedAvg(\Theta_1, \ldots, \Theta_n) = \sum_i^n \frac{\Theta_i}{n_i} \tag{12}$$

where n_i represents the amount of data of client i.

3) FL-AL-80: With the aim of defining this FL model, we extend the "at least half" IOWA operator [45] to "at least 80%" aggregation operator, which is defined in Definition 4.3.

Definition 4.3 (IOWA At Least 80% (AL-80)): The IOWA AL-80 operator of dimension n is a mapping $\Psi : ([0,1] \times \Omega)^n \to \Omega$ which has an associated set of weights $W = w_i^{(0,0.8)}$ and it is defined to aggregate the second arguments of a 2-tuple n list according to the following expression:

$$\Psi_{AL-80}(\langle u_1, \Theta_1 \rangle, \ldots, \langle u_n, \Theta_n \rangle) = \sum_{i=1}^{n} w_i^{(0,0.8)} \Theta_{\sigma(i)} \tag{13}$$

being $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ a permutation function such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall i = \{1, \ldots, n-1\}$ and $u_i = f_{LA}(LLM_i)$.

Figure 2 shows the AL-80 linguistic quantifier as a fuzzy set.

TABLE III
VALUE OF THE PARAMETERS FOR EACH CONFIGURATION OF THE STATIC FL MODELS BASED ON IOWA OPERATORS.

	a	b	c	y_b
IOWA-SQ-0.4	0	0.2	0.8	0.4
IOWA-SQ-0.75	0	0.2	0.8	0.75

4) FL-IOWA-SQ: For the sake of evaluating the dynamic behaviour of FL-IOWA-DQ, we also present its static version which differs from the dynamic version in a manual assignment of the value c. Let us mathematically define IOWA-SQ in Definition 4.4.

Definition 4.4 (IOWA Static Quantifier aggregation operator (IOWA-SQ)): The IOWA-SQ operator of dimension n is a mapping $\Psi : ([0,1] \times \Omega)^n \to \Omega$ which has an associated set of weights $W = w_i^{(a,b,c,y_b)}$ and it is defined to aggregate the second arguments of a 2-tuple n list according to the following expression:

$$\Psi_{IOWA-SQ}(\langle u_1, \Theta_1 \rangle, \ldots, \langle u_n, \Theta_n \rangle) = \sum_{i=1}^{n} w_i^{(a,b,c,y_b)} \Theta_{\sigma(i)} \tag{14}$$

where $0 \leq a \leq b \leq c \leq 1$, $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ is a permutation function, such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall i = \{1, \ldots, n-1\}$ and $u_i = f_{LA}(LLM_i)$.

Analogously to dynamic FL models, we define two different configurations of the static FL model in Table III depending on the top clients weighting parameter y_b.

Figure 3 shows the IOWA-SQ-0.4 linguistic quantifier as a fuzzy set.

V. EXPERIMENTAL RESULTS

We evaluate the performance of FL-IOWA-DQ in different scenarios, in order to compare its performance with other federated aggregation operators, and to evaluate its capacity of dynamically discarding adversarial clients and reducing the number of rounds of learning. The evaluation scenarios are described as what follows:

- AD Scenario: we asses its ability to detect adversarial clients and enhance the performance of the FL model in a scenario with a fixed percentage of them in Section V-A. We study if the filtered out clients are only the adversarial ones.

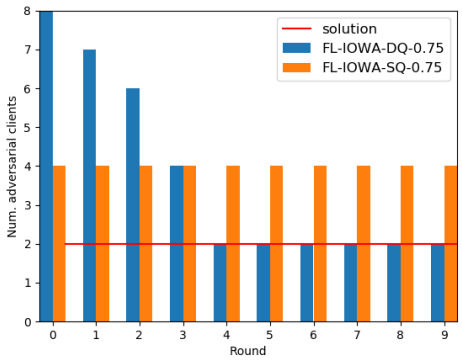

Fig. 4. Adversarial clients detected in AD Scenario with 20 clients.

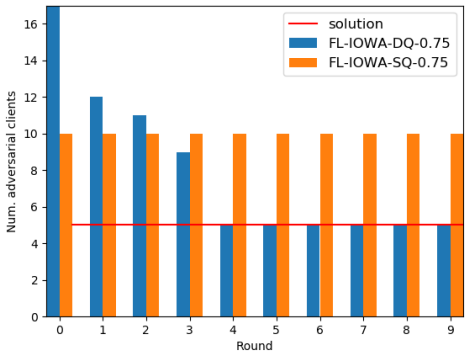

Fig. 5. Adversarial clients detected in AD Scenario with 50 clients.

- NON-AD Scenario: we also evaluate in Section V-B the FL-IOWA-DQ model in a scenario without adversarial clients, in order to study if it can discard clients with a low performance, and if this results in a enhancement of the performance of the FL model.
- High-AD Scenario: we show the significance of the dynamic character of the proposed FL model by increasing the number of adversarial clients in Section V-C, which should be more detrimental to the rest of the FL models.
- Rounds of learning: we analyse the reduction of the number of rounds of learning(see Section V-D) when using the dynamic FL model, which reduces the communication that is one of the main FL handicaps.

In every experimental scenario, the training of the FL models is performed in 10 rounds of learning of the clients. Each client runs 5 training epochs in each round following a mini-batch based training. The evaluation performance was conducted using the data from the EMNIST and Fashion MNIST test datasets and we use accuracy as evaluation measure. We performed 10 runs of each experiment and the result shown is the average of the results obtained in the 10 runs. In each scenario we carry out experiments with 20 and 50 clients to take into account different situations.

A. AD Scenario - With adversarial clients

In this scenario, we consider the non-IID partition of the training data and turn into adversarial clients 10% of the clients, i.e. 2 and 5 adversarial clients respectively.

Table IV shows the results obtained by each FL model. In this case, the superiority of the IOWA-based FL models is evident with respect to the baselines. We highlight its capacity of not considering in the aggregation those clients with a low performance, which are in this case adversarial clients with the intention of manipulating the global FL model.

Concerning the IOWA-based FL models, it is hereby confirmed that dynamic FL models are the ones that reached the highest results, because of the following two reasons:

- The scaled weighting in favour of those clients in the top 20% in accuracy. This factor is confirmed to be advantageous because static IOWA-based FL models get better results than FL-AL-80.
- The dynamic adaptation to the number of adversarial clients in each scenario.

The value y_b may depend on the problem, but a higher value seems to be the best option, because we assign more weight to the top clients.

Identificacion of adversarial clients: Figures 4 and 5 show the number of adversarial clients discarded by each FL-IOWA-based model in each round of federated training in the AD scenario with 20 and 50 clients of which 10% are adversarial clients, *i.e.* two and five clients. Since the parameter y_b does not influence this identification, we show the results with $y_b = 0.75$. The number of clients that are not considered by the static FL model (FL-IOWA-SQ-0.75) is constantly 20% of the clients, *i.e.* 4 and 10 clients. In contrast, dynamic FL model FL-IOWA-DQ-0.4 has a more flexible behaviour. It begins identifying too much adversarial clients (nearly 50%) because some clients with complex training data get poor performance in first rounds. Then, the number of adversarial clients progressively decreases because these clients with complex data improve its accuracy benefited by the rest of the clients. Finally, at the 4th round in both 20 and 50 clients scenario respectively, FL-IOWA-DQ-0.75 succeeds in identifying only and exclusively the adversarial clients. Therefore, the dynamic behaviour of FL-IOWA-DQ-0.75 is positive for improving the global performance of the model, and for only identifying adversarial clients.

TABLE IV
ACCURACY OF FL MODELS IN THE AD SCENARIO.

	EMNIST		**Fashion-MNIST**	
	20 clients	**50 clients**	**20 clients**	**50 clients**
FL-FedAvg	0.9826	0.9791	0.8661	0.8439
FL-W-FedAvg	0.9776	0.9774	0.8699	0.8321
FL-AL-80	0.9832	0.9803	0.8708	0.8469
FL-IOWA-SQ-0.4	0.9863	0.9824	0.8747	0.8541
FL-IOWA-SQ-0.75	0.9883	0.9869	0.8656	0.8671
FL-IOWA-DQ-0.4	0.9870	0.9886	**0.8782**	0.8694
FL-IOWA-DQ-0.75	**0.9900**	**0.9898**	0.8680	**0.8729**

B. NON-AD Scenario - Without adversarial clients

In this scenario, we consider the non-IID partition of the training data without any adversarial clients, in order to analyse the existence of clients with a poor distribution of the data. Table V shows the results reached by each FL model. According to the results, we conclude:

- Every proposed FL-IOWA-based model performs better than the ones presented in the literature.
- Among the FL-IOWA-based models, the dynamics FL models also show a subtle superiority.
- Comparing between dynamic FL models based on the value y_b, in most of the cases the dynamic FL model that assigns more weight to the top 20% clients (FL-IOWA-DQ-0.75) has achieved the best performance, which highlights the impact of this higher weighting on the best clients.

TABLE V
ACCURACY OF FL MODELS IN THE NON-AD SCENARIO.

	EMNIST		Fashion-MNIST	
	20 clients	**50 clients**	**20 clients**	**50 clients**
FL-FedAvg	0.9864	0.9801	0.8704	0.8452
FL-W-FedAvg	0.9857	0.9769	0.8721	0.8396
FL-AL-80	0.9861	0.9807	0.8772	0.8492
FL-IOWA-SQ-0.4	0.9882	0.9836	0.8793	0.8547
FL-IOWA-SQ-0.75	0.9890	0.9868	0.8726	0.8673
FL-IOWA-DQ-0.4	0.9891	0.9848	**0.8953**	0.8684
FL-IOWA-DQ-0.75	**0.9893**	**0.9873**	0.8923	**0.8728**

Although there are not adversarial clients, the results show the relevance of weighting the contribution of each client to the FL model according to their performance. That is because it is very likely that there would be clients with a poor data distribution.

We stress out that dynamic FL models reached better results in AD Scenario than in NON-AD Scenario when EMNIST dataset. This further highlights the benefit to the global learning model of discarding the clients that contribute the least in the aggregation, whether they are poor clients or adversarial ones.

C. High-AD Scenario - Increasing the number of adversarial clients

We evaluated the FL models with a predefined number of adversarial clients in the AD-Scenario, but what would be the performance of the IOWA-based FL models in a scenario with more adversarial clients?

Regarding the previous question, we repeated the evaluation of AD scenario with 30% of adversarial clients, *i.e.* 6 and 15 adversarial clients. Table VI shows the results obtained in terms of the accuracy obtained with 30% adversarial clients. We stress out the significance of the dynamic aspect of the proposed dynamic FL models. While static FL models significantly reduce their performance because it uses adversarial clients in the aggregation, dynamic FL models preserve their performance and even improve the results of the AD scenario with 10% of adversarial clients in some cases. This shows the ability of the FL-IOWA-DQ model to adapt to real scenarios where the number and nature of adversarial clients is unknown.

TABLE VI
ACCURACY OF FL MODELS IN THE SCENARIO WITH 30% OF ADVERSARIAL CLIENTS. WE ALSO SHOW THE DIFFERENCE WITH THE ACCURACY REACHED IN TABLE IV IN ORDER TO COMPARE THE GAIN OR LOSS OF ACCURACY WHEN THE NUMBER OF ADVERSARIAL CLIENTS INCREASED.

	EMNIST		Fashion-MNIST	
	20 clients	**50 clients**	**20 clients**	**50 clients**
FL-FedAvg	0.9788	0.9753	0.8451	0.8435
FL-W-FedAvg	0.9769	0.9758	0.8456	0.8228
FL-AL-80	0.9713	0.9781	0.8439	0.8212
FL-IOWA-SQ-0.4	0.9826	0.9820	0.8468	0.8539
FL-IOWA-SQ-0.75	0.9844	0.9861	0.8518	0.8604
FL-IOWA-DQ-0.4	**0.9876**	0.9860	0.8648	0.8610
FL-IOWA-DQ-0.75	0.9873	**0.9874**	**0.8722**	**0.8684**

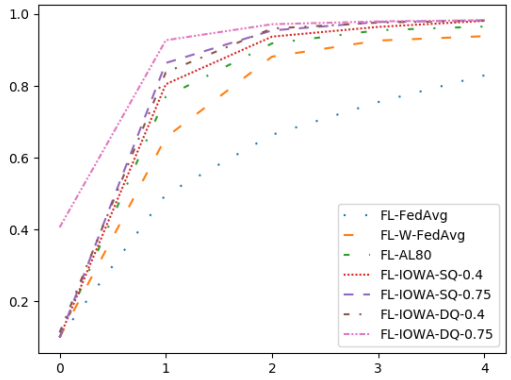

Fig. 6. Accuracy per round of FL models using 20 clients without adversarial clients (NON-AD Scenario) during the first 5 rounds.

D. Analysis of the number of updates

One of the major FL handicaps is the time spent in communications between server and client. For that reason, the reduction of the communication among the server and the clients is crucial in FL. The communication may be reduced by the limitation of the model updates or training rounds, or reducing the amount of information share among the server and the client. In this section we focus on the reduction of the number of training rounds, in order to lessen the number of model updates. In this section, we study the speed of convergence of the IOWA-based FL models, in the sense of the number or training rounds needed to reach good results.

We analyse the accuracy per round achieved in NON-AD Scenario with 20 agents during the first 5 rounds. Figure 6 shows the performance reached by each FL model.

Figure 6 shows that the FL-IOWA-based models allow to reach good results with less training rounds than the FL-FedAvg and FL-W-FedAvg. Specifically, FL-IOWA-DQ-0.75 is the FL model that reaches better results since the first training rounds. We stress out the leading of FL-IOWA-DQ-0.75 since the 1st round. Therefore, the dynamic selection of the clients to aggregate allows to reduce the number of updates among the server and the clients, and hence the number of communication rounds.

VI. Conclusion and future work

In this paper, we propose the dynamic FL model FL-IOWA-DQ, which is built upon a federated aggregation operator based on an IOWA operator and has the capacity of dynamically selecting clients according to its performance to filter out adversarial clients. We evaluated the FL-IOWA-DQ model on the EMNIST and Fashion MNIST image classification benchamrks. We designed different evaluation scenarios, the first type with adversarial clients (AD and High-AD Scenario) and the second type without adversarial ones (NON-AD Scenario). The results of the experiments and the subsequent qualitative analysis show:

1 The FL-IOWA-DQ model outperformed FL-FedAvg and FL-W-FedAvg, as well as the static versions of FL-IOWA-DQ in both scenarios.
2 It is able to only filter out adversarial clients.
3 It has the capacity of reaching a higher quality training model in less training rounds, hence it reduces the number of training rounds and communication updates among the federated server and the clients.
4 Its dynamic nature also allows to reach higher results when the number of adversarial clients is larger than the assumed by the static version of the FL-IOWA-DQ model.

Therefore, we show that the dynamic selection of the clients to be aggregated conducted by FL-IOWA-DQ enhances the performance of the FL model, allow to defend against data poisoning attacks and also reduce the number of rounds of learning needed to converge, which means that our claim holds.

As future work, we plan to deepen in the analysis of the adversarial attacks taxonomy with the purpose of proposing aggregation operators that defend the FL model against more complex attacks. In this way, we will focus on sybil attacks [46] which, our point of view, are the most challenging attacks due to the coordination between clients.

Appendix
Ordered weighted model averaging

Group decision making is the AI task focused on finding out a consensus decision from a set of experts by summing up their individual evaluations. Yager proposed in [42] the Ordered Weighted Averaging (OWA) operators with the aim of modelling the fuzzy opinion majority [45] in group decision making. Yager and Filev generalised the OWA operator definition in [15], where they defined the OWA operator with an order-induced vector for ordering the argument variable. They called this generalisation of OWA operators with a specific semantic in the aggregation process as Induced Ordered Weighted Averaging (IOWA). The OWA and IOWA operators are weighted aggregation functions that are mathematically defined as what follows:

Definition A.1 (OWA Operator [42]): An OWA operator of dimension n is a function $\Phi : \mathbb{R}^n \to \mathbb{R}$ that has an associated set of weights or weighting vector $W = (w_1, \ldots, w_n)$ so that $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$, and it is defined to aggregate a list of real values $\{c_1, \ldots, c_n\}$ according to the Equation 15:

$$\Phi(c_1, \ldots, c_n) = \sum_{i=1}^{n} w_i c_{\sigma(i)} \tag{15}$$

being $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ a permutation function such that $c_{\sigma(i)} \geq c_{\sigma(i+1)}$, $\forall i = \{1, \ldots, n-1\}$.

Definition A.2 (IOWA Operator [15]): An IOWA operator of dimension n is a mapping $\Psi : (\mathbb{R} \times \mathbb{R})^n \to \mathbb{R}$ which has an associated set of weights $W = (w_1, \ldots, w_n)$ so that $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$, and it is defined to aggregate the second arguments of a 2-tuple list $\{\langle u_1, c_1\rangle, \ldots, \langle u_n, c_n\rangle\}$ according to the following expression:

$$\Psi(\langle u_1, c_1\rangle, \ldots, \langle u_n, c_n\rangle) = \sum_{i=1}^{n} w_i c_{\sigma(i)} \tag{16}$$

being $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ a permutation function such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall i = \{1, \ldots, n-1\}$. The vector of values $U = (u_1, \ldots, u_n)$ is called the order-inducing vector and $(c_1, \ldots, c_n)$ the values of the argument variable.

The OWA and IOWA operators are functions for weighting the contribution of experts for the global decision in the case of group decision making, and the contribution of a set of clients in an aggregation process in a general scenario. However, they need an additional function to calculate the values of the parameters, which in the context of group decision making means the grade of membership to a fuzzy concept. The weight value calculation function is known as linguistic quantifier [47], which is defined as a function $Q : [0, 1] \to [0, 1]$ such as $Q(0) = 0$, $Q(1) = 1$ and $Q(x) \geq Q(y)$ for $x > y$. Equation 17 defines how the function Q computes the weight values and Equation 18 defines the behaviour of the function Q.

$$w_i^{(a,b)} = Q_{a,b}\left(\frac{i}{n}\right) - Q_{a,b}\left(\frac{i-1}{n}\right) \tag{17}$$

$$Q_{a,b}(x) = \begin{cases} 0 & 0 \leq x \leq a \\ \dfrac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq 1 \end{cases} \tag{18}$$

where $a, b \in [0, 1]$ satisfying $0 \leq a \leq b \leq 1$.

The function Q in Equation 18 can be redefined in order to model different linguistic quantifiers. Since the definition of the notion quantifier guided aggregation [42, 47], other definitions of the function Q has been proposed to model different linguistic quantifiers like "most" or "at least" [45].

Acknowledgments

This research work is partially supported by the SMART project (TIN2017-89517-P) from the Spanish Government, a grant from the Fondo Europeo de Desarrollo Regional (FEDER) and the contract OTRI-4137 with SHERPA Europe S.L. Nuria Rodrguez Barroso and Eugenio Martnez Cmara was supported by fellowship programmes Formacin de Profesorado Universitario (FPU18/04475) and Juan de la Cierva Incorporacin (IJC2018-036092-I) respectively.

References

[1] J. Dean *et al.*, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1223–1231.

[2] C. Ma *et al.*, "Distributed optimization with arbitrary local solvers," *Optimization Methods and Software*, vol. 32, no. 4, pp. 813–848, 2017.

[3] J. Konen *et al.*, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[4] P. Kairouz *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[5] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2004, p. 99108.

[6] A. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, Long Beach, California, USA, 2019, pp. 634–643.

[7] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, ser. AISec 11. Association for Computing Machinery, 2011, p. 4358.

[8] B. Biggio *et al.*, *Security Evaluation of Support Vector Machines in Adversarial Environments*. Cham: Springer International Publishing, 2014, pp. 105–153.

[9] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[10] M. Jagielski *et al.*, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 19–35.

[11] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1912.04977*, 2016.

[12] J. Konen, H. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1511.03575*, 2016.

[13] Y. Wang, "Co-op: Cooperative machine learning from mobile devices," Master's thesis, University of Alberta, 2017.

[14] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *ArXiv preprint arXiv:1808.04866*, 2018.

[15] R. Yager and D. Filev, "Induced ordered weighted averaging operators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 2, pp. 141–150, 1999.

[16] G. Cohen, S.A., J. Tapson, and A.S., "EMNIST: an extension of MNIST to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

[17] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*.

[18] F. Sattler, S. Wiedemann, K. Mller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. In press, pp. 1–14, 2019.

[19] H. Zhu and Y. Jin, "Multi-objective evolutionary federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1310–1322, 2020.

[20] V. Smith, C. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 4424–4434.

[21] F. Chen, Z. Dong, Z. Li, and X. He, "Federated meta-learning for recommendation," *arXiv preprint arXiv:1802.07876*, 2018.

[22] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 4615–4625.

[23] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[24] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," *ArXiv*, vol. abs/1811.03604, 2018.

[25] M. Chen, R. Mathews, T. Ouyang, and F. Beaufays, "Federated learning of out-of-vocabulary words," *arXiv preprint arXiv:1903.10635*, 2019.

[26] M. Chen *et al.*, "Federated learning of n-gram language models," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 121–130.

[27] M. Yurochkin *et al.*, "Bayesian nonparametric federated learning of neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 7252–7261.

[28] A. Soliman, S. Girdzijauskas, M.-R. Bouguelia, S. Pashami, and S. Nowaczyk, "Decentralized and adaptive k-means clustering for non-iid data using hyperloglog counters," in *Advances in Knowledge Discovery and Data Mining*, H. W. Lauw *et al.*, Eds. Springer International Publishing, 2020, pp. 343–355.

[29] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 634–643.

[30] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al.*, Eds. Curran Associates, Inc., 2017, pp. 119–129.

[31] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR,

2018, pp. 3521–3530.
[32] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, p. 382401, Jul. 1982.
[33] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, Dec. 2017.
[34] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "DRACO: Byzantine-resilient distributed training via redundant gradients," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 903–912.
[35] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020.
[36] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. Online: PMLR, 26–28 Aug 2020, pp. 2938–2948.
[37] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Research in Attacks, Intrusions, and Defenses*, M. Bailey, T. Holz, M. Stamatogiannakis, and S. Ioannidis, Eds. Cham: Springer International Publishing, 2018, pp. 273–294.
[38] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.
[39] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ser. ICML12. Madison, WI, USA: Omnipress, 2012, p. 14671474.
[40] B. Biggio *et al.*, "Evasion attacks against machine learning at test time," *Lecture Notes in Computer Science*, p. 387402, 2013.
[41] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 34, pp. 211–407, 2014.
[42] R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.
[43] N. Rodrguez-Barroso *et al.*, "Federated learning and differential privacy: Software tools analysis, the Sherpa.ai FL framework and methodological guidelines for preserving data privacy," *Information Fusion*, vol. In press, 2020.
[44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[45] G. Pasi and R. Yager, "Modeling the concept of majority opinion in group decision making," *Information Science*, vol. 176, no. 4, pp. 390–414, 2006.
[46] A. Mohaisen and J. Kim, "The sybil attacks and defenses: A survey," *The Smart Computing Review*, vol. 3, 12 2013.
[47] R. Yager, "Quantifier guided aggregation using owa operators," *International Journal of Intelligent Systems*, vol. 11, no. 1, pp. 49–73, 1996.

Nuria Rodríguez-Barroso studied Data Sciences M.Sc. at University of Granada and received two B.Sc. degrees in Computer Science and Mathematics from the same university. Currently, she studies her Ph.D in Computer Science at the University of Granada and she is specially interested on Natural Language Processing, Deep Learning and Federated Learning.

Eugenio Martínez-Cámara is a postdoctoral researcher at University of Granada, Spain. He received a B.Sc. degree in Computer Science and Management and M.Sc. degree in Computer Science from the University of Jaén, Spain, in 2008 and 2010, respectively. He received his Ph.D. in Computer Science in 2015 at the University of Jaén. Dr. Martínez-Cámara also worked as postdoctoral researcher at Technische Universität Darmstadt, Germany. He serves as a member of the editorial board of the journal *Procesamiento del Lenguaje Natural*. His current research interest are related to different Natural Language Processing (NLP) tasks, the application of deep learning in NLP, and the study of novel computing paradigms as Federated Learning.

M. Victoria Luzón is an associate professor in the Software Engineering Department at University of Granada. Her current research interests include sentiment analysis, artificial intelligence and federated learning. Luzóm has a Ph.D. in Industrial Engineering from the University of Vigo, Spain.

Gerardo González Seco received the Ms.C. degree in Computer Science from the University of Cantabria (Spain) in 2012. He has vast experience in the development of artificial intelligence services and products, and he is currently the AI Technology Director in Sherpa.ai. His interests include machine learning operations, parallel and distributed computation, and artificial intelligence based software engineering.

Miguel Ángel Veganzones received the Ms.C. and Ph.D. degrees in Computer Science and Artificial intelligence from the Basque Country University (EHU/UPV), Donostia-San Sebastian, Spain, in 2008 and 2012, respectively. In October 2012, he joined the Images-Signal Department, GIPSA-Lab, Grenoble, France, as a Postdoctoral Researcher. He is currently the AI Director in Sherpa.ai. His research interests include multi-modality analysis, natural language processing, statistical and computational machine learning and explainable artificial intelligence.

Francisco Herrera (SM'15) received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain. He is a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada and Director of the Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI). He is an academician in the Royal Academy of Engineering (Spain). He has been the supervisor of 51 Ph.D. students. He has published more than 500 journal papers, receiving more than 85000 citations (Scholar Google, H-index 143). He has been nominated as a Highly Cited Researcher (in the fields of Computer Science and Engineering, respectively, 2014 to present, Clarivate Analytics). He currently acts as Editor in Chief of the international journal "Information Fusion" (Elsevier). He acts as editorial member of a dozen of journals. His current research interests include among others, Computational Intelligence (including fuzzy modeling, computing with words, evolutionary algorithms and deep learning), information fusion and decision making, and data science (including data preprocessing, prediction, non-standard classification problems, and big data).