Towards Effective Device-Aware Federated Learning

Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, and Antonio Ferrara(✉)

Polytechnic University of Bari, Bari, Italy
{vito.anelli,yashar.deldjoo,tommaso.noia,antonio.ferrara}@poliba.it

Abstract. With the wealth of information produced by social networks, smartphones, medical or financial applications, speculations have been raised about the sensitivity of such data in terms of users' personal privacy and data security. To address the above issues, Federated Learning (FL) has been recently proposed as a means to leave data and computational resources distributed over a large number of nodes (clients) where a central coordinating server aggregates only locally computed updates without knowing the original data. In this work, we extend the FL framework by pushing forward the state the art in the field on several dimensions: (i) unlike the original FedAvg approach relying solely on single criteria (i.e., local dataset size), a suite of *domain-* and *client-specific criteria* constitute the basis to compute each local client's contribution, (ii) the multi-criteria contribution of each device is computed in a prioritized fashion by leveraging a *priority-aware aggregation operator* used in the field of information retrieval, and (iii) a mechanism is proposed for *online-adjustment* of the aggregation operator parameters via a local search strategy with backtracking. Extensive experiments on a publicly available dataset indicate the merits of the proposed approach compared to standard FedAvg baseline.

Keywords: Federated learning · Aggregation · Data distribution

1 Introduction and Context

The vast amount of data generated by billions of mobile and online IoT devices worldwide holds the promise of significantly improved usability and user experience in intelligent applications. This large-scale quantity of rich data has created an opportunity to greatly advance the intelligence of machine learning models by catering powerful deep neural network models. Despite this opportunity, nowadays such pervasive devices can capture a lot of data about the user, information such as what she does, what she sees and even where she goes [15]. Actually, most of these data contain sensitive information that a user may deem private. To respond to concerns about sensitivity of user data in terms of data privacy and security, in the last few years, initiatives have been made by governments

M. Alviano et al. (Eds.): AI*IA 2019, LNAI 11946, pp. 477–491, 2019.
https://doi.org/10.1007/978-3-030-35166-3_34

to prioritize and improve the security and privacy of user data. For instance, in 2018, General Data Protection Regulation (GDPR) was enforced by the European Union to protect users' personal privacy and data security. These issues and regulations pose a new challenge to traditional AI models where one party is involved in collecting, processing and transferring all data to other parties. As a matter of fact, it is easy to foresee the risks and responsibilities involved in storing/processing such sensitive data in the traditional *centralized* AI fashion.

Federated learning is an approach recently proposed by Google [9,10,14] with the goal to train a global machine learning model from a massive amount of data, which is *distributed* on the client devices such as personal mobile phones and/or IoT devices. It is noteworthy that FL differs from traditional distributed learning since we assume that training data (which is supposed to be sensitive) is kept on the very large set of users' private devices they were generated on (e.g., data generated from users' interaction with mobile applications). Therefore, we have to deal with data that is quantitatively unbalanced and differently distributed over devices, i.e. each device data is not a representative sample of the overall distribution. Instead, in a traditional distributed setting, data has to be collected in a centralized location and then evenly distributed over proprietary compute nodes. As a matter of fact, with FL we leverage users' computing power for training a shared ML model while preserving privacy, by actually decoupling the ability to learn a ML model from the need to centrally store private data.

In principle, a FL model is able to deal with fundamental issues related to privacy, ownership and locality of data [2]. In [14], authors introduced the *FederatedAveraging* (FedAvg) algorithm, which combines local stochastic gradient descent on each client via a central server that performs model aggregation by averaging the values of local hyperparameters. To ensure that the developments made in FL scenarios uphold to real-world assumptions, in [3] the authors introduced LEAF, a modular benchmarking framework supplying developers/researchers with a rich number of resources including open-source federated datasets, an evaluation framework, and a number of reference implementations.

Despite its potentially disruptive contribution, we argue that FedAvg exposes some major shortcomings. First, the aggregation operation in FedAvg sets the contribution of each agent proportional to each individual client's local dataset size. A wealth of qualitative measures such as the number of sample classes held by each agent, the divergence of each computed local model from the global model — which may be critical for convergence [16] —, some estimations about the agent computing and connection capabilities or about their honesty and trustworthiness are ignored. While FedAvg only uses limited knowledge about local data, we argue that the integration of the above-mentioned qualitative measures and the expert's domain knowledge is indispensable for increasing the quality of the global model.

The work at hand considerably extends the FedAvg approach [14] by building on three main assumptions:

- we can substantially improve the quality of the global model by incorporating *a set of criteria* about domain and clients, and properly assigning the contribution of individual update in the final model based on these criteria;
- the introduced criteria can be combined by using different aggregation operators; toward this goal, we assert about the potential benefits of using a *prioritized multi-criteria aggregation operator* over the identified set of criteria to define each individual's local update contribution to the federation process;
- computation of parameters for the aggregation operator (the priority order of the above-mentioned criteria) via an *online monitoring and adjustment* is an important factor for improving the quality of global model.

The remainder of the paper is structured as follows. Section 2 is devoted to introducing the proposed FL system, it first describes the standard FL model and then provides a formal description of the proposed FL approach and the key concepts behind integration of local criteria and prioritized multi-criteria aggregation operator in the proposed system. Section 3 details the experimental setup of the entire system by relying on LEAF, an open-source benchmarking framework for federated settings, which comes with a suite of datasets realistically pre-processed for FL scenarios. Section 4 presents results and discussion. Finally, Sect. 5 concludes the paper and discusses future perspectives.

2 Federated Learning and Aggregation Operator

In the following, we introduce the main elements behind the proposed approach. We start by presenting a formal description to the standard FL approach (cf. Sect. 2.1) and then we describe our proposed FL approach (cf. Sect. 2.2).

2.1 Background: Standard FL

In a FL setup, a set $\mathcal{A} = \{A_1, \ldots, A_K\}$ of agents (clients) participate to the training federation with a server S coordinating them. Each agent A_k stores its local data $\mathcal{D}_k = \{(x_1^k, y_1^k), (x_2^k, y_2^k), \ldots, (x_{|\mathcal{D}_k|}^k, y_{|\mathcal{D}_k|}^k)\}$, and never shares them with S. In our setting, x_i^k represents the data sample i of agent k and y_i^k is the corresponding label. The motivation behind a FL setup is mainly efficiency — K can be very large — and privacy [1,14]. As local training data $\mathcal{D}_k$ never leaves federating agent machines, FL models can be trained on user private (and sensitive) data, e.g., the history of her typed messages, which can be considerably different from publicly accessible datasets.

The final objective in FL is to learn a global model characterized by a parameter vector $\mathbf{w}^G \in \mathbb{R}^d$, with d being the number of parameters for the model, such that a global loss is minimized without a direct access to data across clients. The basic idea is to train the global model separately for each agent k on $\mathcal{D}_k$, such that a local loss is minimized and the agents have to share with S only the computed model parameters $\mathbf{w}^k$, which will be aggregated at the server level.

By means of a communication protocol, the agents and the global server exchange information about the parameters of the local and global model. At the t-th round of communication, the central server S broadcasts the current global model $\mathbf{w}_t^G$ to a fraction of agents $\mathcal{A}^- \subset \mathcal{A}$. Then, every agent k in $\mathcal{A}^-$ carries out some optimization steps over its local data $\mathcal{D}_k$ in order to optimize a local loss. Finally, the computed local parameter vector $\mathbf{w}_{t+1}^k$ is sent back to the central server. The central server S computes a weighted mean of the resulting local models in order to obtain an updated global model $\mathbf{w}_{t+1}^G$

$$\mathbf{w}_{t+1}^G = \sum_{k=1}^{|\mathcal{A}^-|} p_{t+1}^k \mathbf{w}_{t+1}^k. \tag{1}$$

For the sake of simplicity of discussion, throughout this work, we do not consider the time dimension and focus our attention on one time instance as given by Eq. (2)

$$\mathbf{w}^G = \sum_{k=1}^{|\mathcal{A}^-|} p^k \mathbf{w}^k, \tag{2}$$

in which $p^k \in [0, 1]$ is the weight associated with agent k and $\sum_{k=1}^{|\mathcal{A}^-|} p^k = 1$.

We argue that collecting information about clients and incorporating that knowledge to compute the appropriate agent-dependent value p^k is important for computing an effective and efficient federated model. Moreover, it is worth noticing that p^k may encode and carry out some useful knowledge in the optimization of the global model with respect to relevant domain-specific dimensions.

2.2 Proposed Federated Learning Approach

As discussed at the end of the previous section, we may have different factors and/or criteria influencing the computation of p^k. Given a set of properly identified criteria about clients, it could be then possible to enhance the global model update procedure by using this information.

To connect it to the formalism presented before, let us assume $C = \{C_1, \ldots, C_m\}$ be a set of measurable properties (criteria) characterizing local agent k or local data $\mathcal{D}_k$. We use the term $c_i^k \in [0, 1]$ to denote, for each agent k, the degree of satisfaction of criterion C_i in a specific round of communication. Hence, in the proposed FL aggregation protocol, the central server computes p^k as

$$p^k = \frac{f(c_1^k, \ldots, c_m^k)}{Z} = \frac{s^k}{Z}, \tag{3}$$

where f is a *local aggregation operation* over the set of properties (criteria), which represent agent k, $s^k \in \mathbb{R}$ is a numerical score evaluating the k-th agent contribution based on the m identified properties and, finally, Z is a normalization factor. In order to ensure that $\sum_{k=1}^{|\mathcal{A}^-|} p^k = 1$ where $p^k \in [0, 1]$, we compute $Z = \sum_{k=1}^{|\mathcal{A}^-|} s^k$.

Example 1. Let us consider three criteria C_1, C_2, C_3 describing, e.g., three specific qualities of the local devices, their produced models or their data. Let us suppose that we have just two clients, and client 1 obtained evaluations $c_1^1 = 0.5, c_2^1 = 0.8, c_3^1 = 0.9$, while client 2 obtained $c_1^2 = 0.2, c_2^1 = 0.9, c_3^1 = 0.7$. Based on Eq. 3, overall evaluation of client 1 and 2 will be proportional and equal to $\frac{f(0.5,0.8,0.9)}{Z}$ and $\frac{f(0.2,0.9,0.7)}{Z}$ in which $Z = f(0.5, 0.8, 0.9) + f(0.2, 0.9, 0.7)$. □

In the following, we briefly discuss the identified set of criteria (together with a motivation for the selection), the selected aggregation operator f, and the online adjustment procedure.

Identification of Local Criteria. In FedAvg, the server performs aggregation to compute p^k, without knowing any information about participating clients, except for a pure quantitative measure about local dataset size. Our approach relies on the assumption that it might be much better to use multiple criteria encoding different useful knowledge about clients to obtain a more informative global model during training. This makes it possible for a domain expert to build the federated model by leveraging different any additional *domain-* and *client-specific* knowledge.

For instance, one may want to choose the criteria in such a way that the rounds of communication needed to reach a desired target accuracy are minimized. Moreover, a domain expert could ask users/clients to measure their adherence to some other target properties (e.g. their nationality, gender, age, job, behavioral characteristics, etc.), in order to build a global model emphasizing the contribution of some classes of users; in this way, the domain expert may, in principle, build a model favoring some targeted commercial purposes.

All in all, we may have a suite of criteria to reach the final global goal (in Sect. 3 we will see the example adopted in our experimental setup).

Prioritized Multi-criteria Aggregation Operator. Once local criteria evaluations have been collected, the central server aggregates them for each device in order to obtain a final score associated to that device. Over the years, a wide range of aggregation operators have been proposed in the field of information retrieval (IR) [13]. We selected some prominent ones and exploited them in our FL setup. In particular, we focused on the weighted averaging operator, the ordered weighted averaging (OWA) models [17,18], which extend the binary logic of *AND* and *OR* operators by allowing representation of intermediate quantifiers, the Choquet-based models [4,7,8], which are able to interpret positive and negative interactions between criteria, and finally the priority-based models [6]. Due to the lack of space, here we report only the approach and the experimental evaluation related to the last one, modeled in terms of a MCDM problem, because of its better performance.

The core idea of the *prioritized multi-criteria aggregation operator* proposed in [6] is to assign a priority order to the involved criteria. The main rationale behind the idea is to allow a domain expert to model circumstances where the

lack of fulfillment of a higher priority criterion cannot be compensated with the fulfillment of a lower priority one [13]. As an example, we may consider the case where the domain expert may want to consider extremely important the age of an agent's user rather than its dataset size, so that even a large local dataset would be penalized if the user age criteria is not satisfied.

Formally, the prioritized multi-criteria aggregation operator $f : [0,1]^m \rightarrow [0,m]$ measures an overall *score* from a prioritized set of criteria evaluations on the local model $\mathbf{w}^k$ as in the following [6]:

$$\begin{aligned} s^k = f(c_1^k, ..., c_m^k) = \sum_{i=1}^{m} \lambda_i \cdot c_{(i)}^k \\ \lambda_1 = 1, \quad \lambda_i = \lambda_{i-1} \cdot c_{(i-1)}^k, \ \ i \in [2,m] \end{aligned} \tag{4}$$

where $c_{(i)}^k$ is the evaluation of $C_{(i)}$ for device k and the $\cdot_{(i)}$ notation indicates the indices of a sorted priority order for criteria, as specified by the domain expert, from the most important to the least important one. For each score $c_{(i)}^k$, an importance weight λ_i is computed, depending both on the specified priority order over the criteria and on the fulfillment and the weight of the immediately preceding criterion.

Example 2. Let us suppose that we are interested in evaluating device k based on three criteria C_1, C_2, C_3 and their respective evaluations are $c_1^k = 0.5, c_2^k = 0.8, c_3^k = 0.9$. Let the priority order of criteria be $C_{(1)} = C_1, C_{(2)} = C_2, C_{(3)} = C_3$, from the most important to the least important; then, $\lambda_1 = 1, \lambda_2 = \lambda_1 \cdot c_{(1)}^k = 0.5, \lambda_3 = \lambda_2 \cdot c_{(2)}^k = 0.4$. Hence, the final device score will be $s^k = (1 \cdot 0.5) + (0.5 \cdot 0.8) + (0.4 \cdot 0.9) = 1.26$. If we change the priority order to be $C_{(1)} = C_3, C_{(2)} = C_2, C_{(3)} = C_1$, we would then obtain $\lambda_1 = 1, \lambda_2 = \lambda_1 \cdot c_{(1)}^k = 0.9, \lambda_3 = \lambda_2 \cdot c_{(2)}^k = 0.72$ with a final device score of $s^k = (1 \cdot 0.9) + (0.9 \cdot 0.8) + (0.4 \cdot 0.5) = 1.82$. We see that this latter value is higher than the previous one since the most important criterion here is better fulfilled. □

Online Adjustment. The aggregation operator we are using takes as parameter the priority order of the involved criteria and, as a consequence, one of the problem is to identify the best ordering for Eq. 4 which takes benefit of the gathered information. Although by definition this priority order could be defined by a domain expert, here we propose to choose the best one in an online fashion such that we can maximize the performances of the model at each round of communication.

Let $(C_{(1),t}, ..., C_{(m),t})$ be the last priority ordering of the criteria used to compute the local scores p_t^k (see Eqs. (3) and (4)) at time t. The sequence of steps needed to compute the updates to the global model is formalized in Algorithm 1 and commented in the following.

Lines 1–7 On each device, we locally train the last broadcasted global model $\mathbf{w}_t^G$ with the local training data, in order to compute $\mathbf{w}_{t+1}^k$; then, we measure the local scores for each of the identified criteria.

Algorithm 1. Sequence of steps executed by the server to compute the new global model with online adjustment of aggregation operator parameters. Functions *ModelUpdate*, *PropertyMeasure*, and *LocalTestAccuracy* are executed locally on the k-th device. Variable acc_t is an estimation of the global accuracy.

Require: $\mathbf{w}_t^G$, acc_t, $(C_{(1),t}, ..., C_{(m),t})$
Ensure: $\mathbf{w}_{t+1}^G$, acc_{t+1}, $(C_{(1),t+1}, ..., C_{(m),t+1})$
1: broadcast $\mathbf{w}_t^G$ to clients in $\mathcal{A}^-$
2: **for** each client $k \in \mathcal{A}^-$ **in parallel do**
3: $\quad\mathbf{w}_{t+1}^k \leftarrow \text{ModelUpdate}(k, \mathbf{w}_t^G)$
4: $\quad$**for** each criterion $C_i \in C$ **do**
5: $\quad\quad c_{i,t+1}^k \leftarrow \text{PropertyMeasure}(k, \mathbf{w}_{t+1}^k, C_i)$
6: $\quad$**end for**
7: **end for**
8: $P \leftarrow (C_{(1),t}, ..., C_{(m),t})$
9: **for** each client $k \in \mathcal{A}^-$ **do**
10: $\quad p_{t+1}^k \leftarrow f(c_{(1),t+1}^k, ..., c_{(m),t+1}^k)/Z$
11: **end for**
12: $\overline{\mathbf{w}}_{t+1}^G \leftarrow \sum_{k=1}^{|\mathcal{A}^-|} p_{t+1}^k \mathbf{w}_{t+1}^k$
13: **for** each client $k \in \mathcal{A}$ **in parallel do**
14: $\quad\text{acc}_{t+1}^k \leftarrow \text{LocalTestAccuracy}(k, \overline{\mathbf{w}}_{t+1}^G)$
15: **end for**
16: $\text{acc}_{t+1} \leftarrow$ weighted average of acc_{t+1}^k w.r.t. local test set size, $\forall k \in \mathcal{A}$
17: **while** $\text{acc}_{t+1} < \text{acc}_t$ **do**
18: $\quad$**if** other priority orderings are available **then**
19: $\quad\quad P \leftarrow$ another priority ordering of criteria $(C_{(1)}, ..., C_{(m)})\star$
20: $\quad\quad$repeat steps 9—16
21: $\quad$**else**
22: $\quad\quad P \leftarrow$ priority ordering for which we get the maximum value for acc_{t+1}
23: $\quad\quad\text{acc}_{t+1}^k \leftarrow$ accuracy of the model which performed best
24: $\quad\quad$repeat steps 9—12
25: $\quad\quad$**break**
26: $\quad$**end if**
27: **end while**
28: $(C_{(1),t+1}, ..., C_{(m),t+1}) \leftarrow P$
29: $\mathbf{w}_{t+1}^G \leftarrow \overline{\mathbf{w}}_{t+1}^G$

Lines 9–11 For each device, we use the priority ordering of criteria already used in the previous round of communication to compute the local score p_{t+1}^k.

Line 12 A new *candidate* global model $\overline{\mathbf{w}}_{t+1}^G$ is built by computing a weighted averaging of the local models w.r.t. the computed p_{t+1}^k.

Lines 13–15 On each device, $\overline{\mathbf{w}}_{t+1}^G$ is locally tested using the local test set.

Lines 16–29 An estimation of a global accuracy is computed weighting local accuracies w.r.t. local test set size; then, if the obtained accuracy is higher on average than the accuracy obtained with $\mathbf{w}_t^G$, then we update the global value $\mathbf{w}_{t+1}^G \leftarrow \overline{\mathbf{w}}_{t+1}^G$ and we proceed with the next round of communication; otherwise, another permutation is considered and, once a new p_{t+1}^k is computed for each device, we go back to step 3; if no other permutations are available,

the candidate global model which produced the least worst test accuracy is assigned to $\mathbf{w}_{t+1}^{G}$.

The above-mentioned steps are also graphically illustrated by means of a plot in Fig. 1, where an exemplification with dummy values is presented. Training steps proceed with the same parametrization until a lower accuracy is obtained (blue point in round of communication 8); then, the previous model is restored and the other configurations are tested, until a higher accuracy is found (e.g., orange point in round 8). When a higher accuracy cannot be found, the least worst option is selected (e.g., green point in round 10).[1]

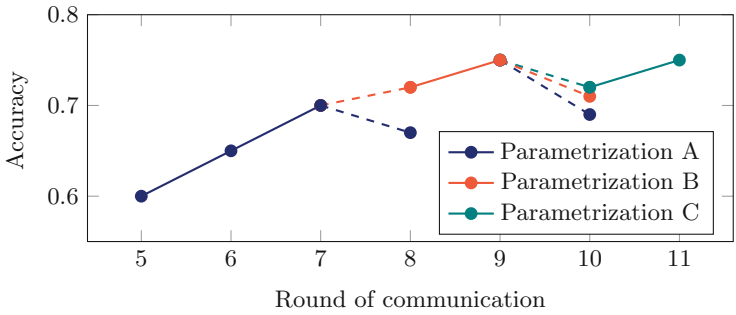

Fig. 1. An illustration of the online parameter adjustment for the aggregation operator. (Color figure online)

3 Experimental Setup

In this section we describe the experimental setup used to validate the performance of the proposed FL system.

Experimental Evaluation Framework. In order to perform the experimental validation and performance evaluation, an extensive set of experiments has been carried out by relying on LEAF [3], a modular open-source benchmarking framework for federated settings, which comes with a suite of datasets appropriately preprocessed for FL scenarios. LEAF also provides reproducible reference

[1] We should be reminded that the proposed adjustment algorithm may involve some communication and computational overhead due to the need of evaluating each of the candidate global models on local test data. We have not included this overhead in the count of rounds, since in the literature of FL a round of communication is defined as the entire process of model exchanging between clients and server and local model training [11]. Alternatively, we could define these extra rounds as *testing rounds*, which imply the same communication cost as a round of communication, but a significantly lower computational power. In the worst case, we would need $m!$ testing rounds for each round of communication, where m is the number of criteria.

implementations and introduces both system and statistical rigorous metrics for understanding the quality of the FL approach.

As for the metrics computation, the global model is tested on each device over the local test sets. The objective of LEAF is to capture the distribution of performance across devices by considering the 10th and 90th percentiles of the local accuracy values and by estimating a global accuracy (local accuracy values are averaged weighting them based on local test set size).

In this work, we improve the validation of the FL setting by using an approach which offers an overview of the whole training performances, instead of metrics describing a single round of communication. More specifically, *we measure the number of round of communication required to allow a certain percentage of devices, which participate to the federation process, to reach a target accuracy (e.g., 75% or 80%)*, since this measurement is able to fairly show how effective and efficient is the model across the devices.

Federated Dataset. We run our experiments using the FEMNIST dataset [3], which contains handwritten characters and digits from various writers and their true labels. Unlike the original FedAvg algorithm [14], which uses the MNIST dataset [12] artificially split by labels, the FEMNIST dataset [3], is larger and more realistically distributed. The dataset contains 805,263 examples of 62 classes of handwritten characters and digits from 3,550 writers and it is built by partitioning data in ExtendedMNIST [5] — an extended version of MNIST with letters and digits — based on writers of digits/characters. It is important to note that data in FEMNIST is inherently non-IID distributed, as the local training data can vary between clients; therefore, it is not representative of the whole population distribution. We use the described dataset to perform a digit/character classification task, although for computational limits we use a subsampled version (10% of total, 371 clients involved). Even though this training data is quite simple, in our view FEMNIST is sufficiently appropriate for our purposes, since one of the most motivating example of FL is when the training data comes from personal users' interaction with mobile applications. Actually, one could find interesting to eventually experiment our approach with different datasets, for example with a less marked user-dependence.

Convolutional Model. Similar to [14], the classification task is performed by using a convolutional neural network (CNN). The network has two convolutional layers with 5×5 filters — the first with 32 channels, the second with 64, each followed by 2×2 max pooling —, a fully connected layer with 2048 units and ReLu activation, and a final softmax output layer, with a total of 6,603,710 parameters.

Hyperparameter Settings. We set the hyperparameters for the whole set of our experiments as follows, also guided by the results obtained in [14]. As for the FedAvg client fraction parameter, in each round of communication only 10% of clients are selected to perform the computation. For what concerns the parameters of stochastic gradient decent (SGD), we set the local batch size to 10 and the number of local epochs equal to 5. This is the configuration that in the baseline makes it possible to reach the target accuracy in less rounds of communication. Moreover, we set the learning rate to $\eta = 0.01$. Finally, we set the maximum number of rounds of communication per each experiment to 1000.

Identified Local Criteria. In our experimental setting, the proposed FL system extends pure quantitative criteria in FedAvg [14] — dataset size — and leverages two new criteria. Please note that we are not stating that the proposed ones are the only possible criteria. We present them just to show how the introduction of new information may lead to a better final model. More specifically, in our experimental evaluation, we aim at both reducing the number of rounds of communication necessary to reach a target accuracy and making the global model not diverging towards local specializations and overfittings.

The criteria have been defined so that $c_i^k \in [0, 1]$ with 0 meaning bad performance and 1 good performance. Moreover, in order to make each criterion lying in the same interval scale, we normalized them such that $\sum_{k=1}^{|\mathcal{A}^-|} c_i^k = 1$.

Local Dataset Size(***base DS***)**.** The first criterion we considered is the one already used by FedAvg [14] namely the local dataset size given by $c_1^k = |\mathcal{D}_k|/|\cup_{i\in\mathcal{A}^-}\mathcal{D}_i|$. This criterion is a *pure quantitative measure* about the local data, which will serve both as baseline in empirical validation of the results (i.e., when used in isolation) and as part of the entire identified set of criteria in the developed FL system (i.e., when used in a group).

Local Label Diversity(***Ld***)**.** The second considered criterion is the *diversity of labels* in each local dataset, measuring the diversity of each local dataset in terms of class labels. We assert this criterion to be important since it can provide a clue on how much each device can be useful for learning to predict different labels. To quantify this criterion we use $c_2^k = \delta(\mathcal{D}_k)/\sum_{i\in\mathcal{A}^-}\delta(\mathcal{D}_i)$ where δ measures the number of different labels (classes) present over the samples of that dataset.

Local Model Divergence(***Md***)**.** With non-IID distributions — and this is the case of our dataset — model performance dramatically gets worse [19]. Moreover, a large number of local training epochs may lead each device to move further away from the initial global model, towards the opposite of the global objective [16]. Therefore, a possible solution inspired by [16] is to limit these negative effects, by penalizing higher divergences and highlighting local models that are not very far from the received global model. We evaluate the local model divergence as $c_3^k = \varphi^k/\sum_{i\in\mathcal{A}^-}\varphi^i$ where $\varphi^i = \frac{1}{\sqrt{||\mathbf{w}^G - \mathbf{w}^i||_2 + 1}}$.

4 Results and Discussion

In order to validate the empirical performance of the proposed FL system, an extensive set of experiments has been carried out with respect to three understudy exploration dimensions in agreement with the assumptions presented in Sect. 1. The final results are shown in Table 1. Note that they are presented for reaching two distinctive desired target global accuracy of 75% and 80%.[2] Each column indicates the percentage of devices participating to the federation that is able to reach a desired target accuracy[3]. In addition, we present the results in three groups of (**Low**, **Mid**, **High**) for percentage of participating devices.

Study A: Effect of Individual Criteria. Study A contemplates answering the question: "*Are we able to introduce a set of device- and data- dependent criteria through the help of which we can train a better global model?*". The results for this study are summarized in the row **Ind** of Table 1. To answer the previous question, we considered the effect of each of the three identified criteria **base Ds**, **Md**, **Ld** *in isolation*, i.e., alternatively using only one of them. The results with respect to both desired accuracies of 75% and 80% show that the new identified criteria (**Md** and **Ld**) have an impact in the final quality of the global model, which is *comparable* (in **Low** and **Mid** cases) or *superior* with respect to the conventional **base Ds** criteria (in the case of **High)**. For example, when comparing **Md** and **Ld**, one can notice the results are equal to 25.5 v.s. 27 with a marginal difference of only 6%. This is while, if we desire to satisfy a higher number of devices (**High** case) to reach a certain accuracy, the proposed criteria show a quality substantially better than the **base Ds** criteria. For example, **Ld** has a mean performance of 405 compared with 552.5 obtained **base Ds**. This is equal to an improvement of 36% with respect to existing baseline. These initial results already show how the global model can benefit from considering other criteria than just the dataset size.

Study B: Impact of Priority Order in Multi-criteria Aggregation. Study B focuses on the question: "*Are we able to exploit the potential benefits of a prioritized multi-criteria aggregation operator to build a more informative global model based on the identified criteria?*". The results for this study are summarized in row **MCA** of Table 1. To answer this research question, we performed one experiment for each individual permutation of criteria in the prioritized multi-criteria aggregation setting. Since there are 3 identified criteria, we have in total 6 permutations of criteria. For a fine-grained analysis, we provide the results obtained for *all the permutation runs*, denoted, e.g., by $\mathbf{Ds} \succ \mathbf{Ld} \succ \mathbf{Md}$,

[2] We chose these values since they represent reasonable accuracy values and higher were not reached in the 1,000 allowed rounds of communication.

[3] The total number of participating devices in the federation is 371, thus 20%, as an example, indicates the round of communication required for $0.2 \times 317 = 75$ devices to reach the desired target accuracy.

Ds $\succ$ **Md** $\succ$ **Ld**. By looking at the results, we can notice that in **Low** and **Mid** categories, the best results are obtained for **Ds** $\succ$ **Ld** $\succ$ **Md** and **Ds** $\succ$ **Md** $\succ$ **Ld**. These results share a similar characteristic, which involves the fact that by considering **Ds** as the first important criterion, we can grant a smaller subset of devices the chance to reach to a desired target accuracy in faster pace/rate. This result is in agreement with individual results (see **Ind** in Table 1) in the sense that the *criterion **Ds** provides the best quality in **Low** and **Mid** study cases for both desired target accuracy of 75% and 80%*. However, when concentrating on the **High** category, one can notice **Md** $\succ$ **Ds** $\succ$ **Ld** provides the best performance. This result is a bit surprising and shows that to satisfy a higher number of devices, the criterion **Md** plays the most important role. This result is surprising from the sense that in the individual results (see **Ind** in Table 1), **Ld** has the most important performance, while in the obtained result it has the lowest priority. Interestingly, we may notice that in all these best cases, the pattern **Ds** $\succ$ **Ld** always occurs[4].

Study C: Impact of Online Adjustment of the Priority-Order in Multi-criteria Aggregation. Finally, study C answers the question: "*Is it possible to update parameters for the aggregation operator (the priority order of the above-mentioned criteria) via an online monitoring and adjustment or improving the quality of global model?*". The results for this study are summarized in row **Final** of Table 1. This study in fact is concerned with the *dynamic* behavior of our proposed FL approach, by letting the server choose at each round of communication the priority ordering maximizing the accuracy (i.e, obtain the best sub-optimal accuracy). Similar to the previous study, here we also run six experiments, related to the six possible *initializations* for the priority combinations. In Table 1 we show results related to the best run and to their mean. In this final experimental setting, we see an overall improvement in the performances of the proposes approach when we initialize the priority ordering with **Md** $\succ$ **Ds** $\succ$ **Ld**. Also in this case, the pattern **Ds** $\succ$ **Ld** occurs. In this final stage of our proposed approach we can notice it outperforms FL original algorithm, although we take into account the increased communication and computational requirements already discussed in Sect. 2.2.

[4] We remember here that a preference relation $\succ$ is transitive. Hence **Ds** $\succ$ **Md** $\succ$ **Ld** implies **Ds** $\succ$ **Ld**.

Table 1. Final results of the empirical evaluation. Each cell provides the number of rounds of communication necessary to make the percentage of devices specified in the columns reach a desired target accuracy (either 75% or 80% in our case). Runs that did not reach the target accuracy for the specified percentage of devices in the 1,000 allowed rounds are marked with —. The best results obtained in study MCA are shown in **underlined bold**, while the best results in study Final are shown in **bold**.

Study/% devices		Low			Mid			High		
		20%	30%	Mean	40%	50%	Mean	70%	75%	mean
Target accuracy 75%										
Ind	*Dataset size (base)*	*22*	*29*	*25.5*	*39*	*62*	*50.5*	*304*	*801*	*552.5*
	Model divergence	24	30	27	41	67	54	274	768	521
	Label diversity	25	32	28.5	43	70	56.5	278	532	405
MCA	Ds ≻ Ld ≻ Md	**20**	**29**	**24.5**	**39**	**60**	**49.5**	300	823	561.5
	Ds ≻ Md ≻ Ld	**20**	**29**	**24.5**	**39**	**60**	**49.5**	300	669	484.5
	Ld ≻ Ds ≻ Md	24	31	27.5	41	68	54.5	259	768	513.5
	Md ≻ Ds ≻ Ld	24	32	28	45	70	57.5	**255**	**532**	**393.5**
	Ld ≻ Md ≻ Ds	23	30	26.5	41	68	54.5	270	729	499.5
	Md ≻ Ld ≻ Ds	24	32	28	46	70	58	255	620	437.5
	Mean	22.5	30.5	26.5	41.8	66	53.9	273.17	690.1	481.6
Final	Md ≻ Ds ≻ Ld	**12**	**19**	**15.5**	**26**	**57**	**41.5**	**164**	**494**	**329**
	Mean	20.5	27.5	24	38.6	61.8	50.2	223	611.8	417.4
Target accuracy 80%										
Ind	*Dataset size (base)*	*31*	*45*	*38*	*72*	*136*	*104*	—	—	—
	Model divergence	31	46	38.5	82	151	116.5	—	—	—
	Label diversity	36	53	44.5	90	161	125.5	—	—	—
MCA	Ds ≻ Ld ≻ Md	**30**	**45**	**37.5**	**72**	**135**	**103.5**	—	—	—
	Ds ≻ Md ≻ Ld	**30**	**45**	**37.5**	**72**	**135**	**103.5**	—	—	—
	Ld ≻ Ds ≻ Md	31	46	38.5	82	149	115.5	—	—	—
	Md ≻ Ds ≻ Ld	36	53	44.5	84	161	122.5	—	—	—
	Ld ≻ Md ≻ Ds	31	46	38.5	82	151	116.5	—	—	—
	Md ≻ Ld ≻ Ds	36	53	44.5	90	161	125.5	—	—	—
	Mean	32.3	48	40.1	80.3	148.6	114.5	—	—	—
Final	Md ≻ Ds ≻ Ld	**21**	**36**	**28.5**	**61**	**133**	**97**	—	—	—
	Mean	30	43.5	36.7	78.1	142.6	110.4	—	—	—

5 Conclusions and Future Perspectives

In this work, we presented a practical protocol for effectively aggregating data by proposing a set of *device-* and *data-aware* properties (criteria) that are exploited by a central server in order to obtain a more qualitative/informative global model. Our experiments show that the standard federated learning standard, FedAvg can be substantially improved by training high-quality models using relatively few rounds of communication, by using a properly defined set of local criteria and using aggregation strategy that can exploit the information from such criteria. We want to stress here that devising such criteria is not a trivial task, and we deem necessary the knowledge of experts in the specific field or

domain. Moreover, it would be arduous to find a general criterion that would meet the needs of all domains. Future perspectives for this work concern with the identification of other local criteria — both general purpose and domain-specific —, the experimentation with other aggregation operators and with other interesting datasets, as well as the extension of this federated approach to other machine learning systems, such as those in recommendation domain.

Acknowledgements. The authors wish to thank Angelo Schiavone for fruitful discussions and for helping with the implementation of the framework.

References

1. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. arXiv preprint arXiv:1807.00459 (2018)
2. Bonawitz, K., et al.: Towards federated learning at scale: system design. CoRR abs/1902.01046 (2019). http://arxiv.org/abs/1902.01046
3. Caldas, S., et al.: Leaf: a benchmark for federated settings. arXiv preprint arXiv:1812.01097 (2018)
4. Choquet, G.: Theory of capacities. Annales de l'Institut Fourier **5**, 131–295 (1954). https://doi.org/10.5802/aif.53
5. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: EMNIST: extending MNIST to handwritten letters. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2921–2926. IEEE (2017)
6. da Costa Pereira, C., Dragoni, M., Pasi, G.: Multidimensional relevance: prioritized aggregation in a personalized information retrieval setting. Inf. Process. Manag. **48**(2), 340–357 (2012). https://doi.org/10.1016/j.ipm.2011.07.001
7. Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. Eur. J. Oper. Res. **89**(3), 445–456 (1996). https://doi.org/10.1016/0377-2217(95)00176-X. http://www.sciencedirect.com/science/article/pii/037722179500176X
8. Grabisch, M., Roubens, M.: Application of the Choquet integral in multicriteria decision making. In: Fuzzy Measures and Integrals, pp. 348–374 (2000)
9. Konecný, J., McMahan, B., Ramage, D.: Federated optimization: distributed optimization beyond the datacenter. CoRR abs/1511.03575 (2015). http://arxiv.org/abs/1511.03575
10. Konecný, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated optimization: distributed machine learning for on-device intelligence. CoRR abs/1610.02527 (2016). http://arxiv.org/abs/1610.02527
11. Konecný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency. CoRR abs/1610.05492 (2016). http://arxiv.org/abs/1610.05492
12. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
13. Marrara, S., Pasi, G., Viviani, M.: Aggregation operators in information retrieval. Fuzzy Sets Syst. **324**, 3–19 (2017). https://doi.org/10.1016/j.fss.2016.12.018
14. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS

2017, Fort Lauderdale, FL, USA, 20–22 April 2017, pp. 1273–1282 (2017). http://proceedings.mlr.press/v54/mcmahan17a.html
15. Miller, K.W., Voas, J.M., Hurlburt, G.F.: BYOD: security and privacyconsiderations. IT Prof. **14**(5), 53–55 (2012). https://doi.org/10.1109/MITP.2012.93
16. Sahu, A.K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., Smith, V.: On the convergence of federated optimization in heterogeneous networks. arXiv preprint arXiv:1812.06127 (2018)
17. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Trans. Syst. Man Cybern. **18**(1), 183–190 (1988). https://doi.org/10.1109/21.87068
18. Yager, R.R.: Quantifier guided aggregation using OWA operators. Int. J. Intell. Syst. **11**(1), 49–73 (1996)
19. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. CoRR abs/1806.00582 (2018). http://arxiv.org/abs/1806.00582