Future Generation Computer Systems 166 (2025) 107639

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Contribution prediction in federated learning via client behavior evaluation

Ahmed A. Al-Saedi [a,*], Veselka Boeva [a], Emiliano Casalicchio [a,b]

[a] Department of Computer Science, Blekinge Institute of Technology, Karlskrona, SE-371 79, Sweden
[b] Department of Computer Science, Sapienza University of Rome, Rome, 00185, Italy

ARTICLE INFO

Keywords:
Federated learning
Contribution evaluation
Clustering analysis
Eccentricity analysis
Behavior monitoring

ABSTRACT

Federated learning (FL), a decentralized machine learning framework that allows edge devices (i.e., clients) to train a global model while preserving data/client privacy, has become increasingly popular recently. In FL, a shared global model is built by aggregating the updated parameters in a distributed manner. To incentivize data owners to participate in FL, it is essential for service providers to fairly evaluate the contribution of each data owner to the shared model during the learning process. To the best of our knowledge, most existing solutions are resource-demanding and usually run as an additional evaluation procedure. The latter produces an expensive computational cost for large data owners. In this paper, we present simple and effective FL solutions that show how the clients' behavior can be evaluated during the training process with respect to reliability, and this is demonstrated for two existing FL models, Cluster Analysis-based Federated Learning (CA-FL) and Group-Personalized FL (GP-FL), respectively. In the former model, CA-FL, the frequency of each client to be selected as a cluster representative and in that way to be involved in the building of the shared model is assessed. This can eventually be considered as a measure of the respective client data reliability. In the latter model, GP-FL, we calculate how many times each client changes a cluster it belongs to during FL training, which can be interpreted as a measure of the client's unstable behavior, i.e., it can be considered as not very reliable. We validate our FL approaches on three LEAF datasets and benchmark their performance to two baseline contribution evaluation approaches. The experimental results demonstrate that by applying the two FL models we are able to get robust evaluations of clients' behavior during the training process. These evaluations can be used for further studying, comparing, understanding, and eventually predicting clients' contributions to the shared global model.

1. Introduction

Federated Learning (FL) [1], a novel decentralized machine learning paradigm, enables cooperative model training through multi-party collaboration by transferring model parameters instead of transferring local private data during the training process. From a high-level point of view, FL allows multiple data holders (clients hereafter) to train a shared global model with a central server through model parameter exchange, without disclosing their own data [2,3]. In this context, successfully evaluating the quality of each client's data to the federated global model becomes an essential challenge in FL applications [4,5]. Therefore, a key question in FL is how to fairly and inexpensively evaluate clients' behavior, using different data sources, for building an accurate FL model [6]. To this end, two methods have been applied in the literature for contribution measurement. The Deletion approach (based on the conventional Leave-one-out (LOO) analyses [7]), and FL-Cohort [8] have provided solutions to the contribution evaluation problem. The Deletion approach [7] works by retraining the federated model without the contribution of a given client's data and measuring the importance of retrained model changes to evaluate this client's contribution to the overall federated model. Instead of removing a single client at a time, FL-Cohort [8] eliminates multiple, similar clients at each FL global training round. Note that these approaches evaluate the clients' contribution in a separate procedure in addition to the training process, producing the shared global model. Thus, on the one hand, these techniques perform an independent evaluation of the client's contribution, increasing the use of the computational resources and the time to discover weak contributors. On the other hand, FL-Cohort measures the global imbalance only once prior to training the FL model and does not take into account the data dynamic during the training process. Evidently, we need new effective FL frameworks, that can alleviate the resource usage and time required to evaluate the behavior of the clients involved in the training process.

The Deletion and FL-Cohort approaches only focus on iteratively removing a single client or group of clients from global training at

* Corresponding author.
E-mail address: abs@bth.se (A.A. Al-Saedi).

https://doi.org/10.1016/j.future.2024.107639
Received 10 December 2023; Received in revised form 30 September 2024; Accepted 23 November 2024
Available online 30 November 2024

each global round to measure the contribution of the parties involved in the federation. Therefore, the resource consumption and the time to evaluate the use behavior for these approaches is proportional to the number of clients, making them impractical in real scenarios. In addition, those FL solutions assume that the data does not change significantly at each client (e.g., no concept drift, new data is assumed to be from the same distribution, etc.), which is unrealistic.

Provoked by the above challenges and the limitations of previous research efforts, we present an efficient contribution evaluation approach inspired by the ideas of CA-FL [9] and GP-FL [10]. We have found that the CA-FL mechanism can effectively handle the problem of clients' behavior assessment. Specifically, the CA-FL can evaluate the frequency of each client to be a cluster representative during the training process. The calculated score can be considered as an indirect measure of the data quality (reliability) of a particular client. In general, the CA-FL selects the top-performing (the most reliable) clients at each training round for the global training of the shared model in FL scenario. Furthermore, the GP-FL provides with another technique that can evaluate the reliability of each client. In detail, the GP-FL evaluates how many times the client has changed its cluster during the FL training process. In practice, if the client changes the cluster it belongs to very often, this means this client has varying behavior, because it does not have a stable position in the clustering structure determined on the base of clients' class distributions. This can also be interpreted as an indirect evaluation of the stability of the data provided by the client.

The novelty of our proposed approach is to predict the contribution of clients in FL by evaluating their behavior during the distributed training process without using an additional evaluation procedure. Most of the existing and widely used contribution evaluation approaches in FL, such as the Deletion approach [7] and FL-Cohort [8], are a separate resource expensive procedure that is relied on the evaluation how the model performance is affected when the client's data is excluded from the training. Our work's contribution is summarized as follows:

1. We propose two approaches, inspired by CA-FL and GP-FL federated learning algorithms, to be used to evaluate each client's behavior during the training process, i.e., without requiring any extra computational cost.
2. The CA-FL method is capable of recognizing the clients that demonstrate reliable behavior during the training. The CA-FL selects the local updates of the top-performing clients at each training round to be aggregated into the global model. This additionally contributes to achieving high performance in terms of accuracy and computational overhead in comparison with the conventional FL approaches.
3. The GP-FL method is capable of identifying clients that demonstrate unreliable behavior during the training. It counts how many time each client changes its cluster, and in that way the clients that have highly unstable behavior can be drop out the federation, which will contribute to the increase of performance of the overall model.
4. Our approaches can provide with realistic evaluations of the clients' behavior during the training process. These allow to discriminate between the clients that can eventually contribute to the performance of the FL model and ones that are not capable to do this.
5. We evaluate our CA-FL and GP-FL methods in different experimental scenarios on three LEAF datasets [11–13], which specifically emulate federated environments to show their robustness and effectiveness. The performance of our CA-FL and GP-FL methods is benchmarked to that of two baseline approaches, namely the Deletion approach [7] and FL-Cohort [8]. The obtained experimental results demonstrate the robustness of our approaches in comparison to the two baseline techniques in monitoring and evaluating clients' behavior during the training process.

The remainder of this paper is organized as follows. The related works are discussed in Section 2. The motivation, preliminaries and the study details are described in Sections 3 and 4, respectively. Section 5 explains the experimental setup, while Section 6 describes the experimental results and provides a discussion. The paper is concluded in Section 7.

2. Related work

Our discussion in this section is focused on three topics related to our work: (1) federated learning, (2) cluster-based federated learning solutions and (3) contribution evaluation in FL.

2.1. Federated learning

Federated learning has received much attention as a distributed training of machine learning techniques due to its high performance and ability to mitigate many privacy concerns and costs. Subsequently, several excellent research achievements have been introduced to face different issues encountered in FL, including heterogeneity challenges [14–17], privacy and security threats [18–20], communication overhead [9,21–23], and convergence analysis [24–27]. Caldas et al. [28] have proposed LEAF – a benchmark in the exploration of experiments on federated settings, which are more practical than the simulated datasets. However, the research status on current FL challenges is still in continuous improvement.

2.2. Cluster-based federated learning solutions

The formulation of clustered FL has significant attention in many recent works to design efficient client selection strategies and overcome challenges caused by Non-IID data in a federated setting, where the clients are assumed to be in various groups. Sattler et al. [29] have proposed a Clustered Federated Learning (CFL), clustering framework that deals with federated multi-task learning settings. The CFL recursively separates clients into clusters of similar clients based on the geometric properties of their local data distributions. Authors in [30] propose a similar algorithm named FedGroup, based on a similarity-based clustering approach. Likewise, GP-FL [10] trains a set of global models, one per cluster of clients. It splits clients into groups based on local data distribution with respect to the classes. Ghosh et al. have proposed in [31] an iterative federated clustering approach (IFCA). It alternates between identifying the group membership of each client and optimizing the group models. The work in [32] has presented federated SEM (FeSEM) by solving a joint optimization and assigning the center for each client. More recently, soft clustering FL like FedEM [33] and FedSoft [34] have been introduced which allow clients to follow a mixture of multiple distributions. In HyperCluster [35], the authors propose to assign each client to the group whose model produces the lowest loss on its local data. CA-FL [9] proposes that an FL model can reduce communication overheads via clustering analysis of the client updates. It identifies the most representative clients to communicate with the server to reduce communication overheads.

2.3. Contribution evaluation

A recent study of privacy and fairness concerns in the context of FL [36] discusses three different categories of fairness: performance, collaboration and model fairness. Performance fairness has as aim to achieve equal accuracy distribution across participants. The focus of collaboration fairness is on providing with greater rewards/incentives to participants with greater contributions. Model fairness is directed to maintain a balance between accuracy and privacy preservation using FL by protecting some specific model characteristics. This study is related to collaboration fairness by proposing a novel approach for evaluating participants' behavior during the training that can be applied to predict their contribution.

Contribution evaluation in FL has been widely studied in recent years to indicate performance improvement (e.g., client selection, incentive allocation, etc.) [37]. However, it is challenging to measure participants' contribution under FL settings since participants keep their original data — local and private. The majority of the existing works in the domain can be distributed into two main categories: a reduction in the number of sub-model evaluations required and an acceleration of a single round of evaluation. The authors in [38] developed DNN partitioning to minimize the FL training latency under a device-specific participation rate to involve the devices with better imbalanced data distribution over more communication rounds. The Shapley Value [39] and Leave-one-out (LOO), namely the Deletion approach [7], are applicable valuation methods that use local gradients as a tool for client contribution evaluation. [6] utilizes data quality, data volume, and data collection cost to determine each participant's contribution levels. Yu et al. [40] develop an incentive approach that compensates participants for their contributions to the federation by measuring according to self-reported data quantity and quality. Zhao et al. [41] measure the participant's contributions according to the similarity between local updates and the global model. They assume that local updates similar to the global model are more valuable. In [42], the authors use self-reported local data size and bandwidth for measuring utility with the Cobb–Douglas function. Lyu et al. [43] proposed an approach that compares a pairwise similarity to let participants perform a mutual evaluation on each others' generated samples. In [44] a concept of the contribution index to quantify participants' contributions by considering local datasets and machine learning models is proposed. In [45], the authors study the problem of data valuation using Shapley value to assess each participant's contribution. Shyn et al. [37] introduces an empirical approach called Federated Client Contribution Evaluation through Accuracy Approximation (FedCCEA) to evaluate client contribution using data size in a feasible time. FL-Cohort [8] removes a group of similar clients in each FL training and calculates the contribution of each cluster separately. However, the aforementioned studies require FL participants to incur significant computation and communication costs, making such approaches difficult to apply in practice. In contrast, we evaluate each client's behavior without incurring any additional computation costs. We compare our work with the Deletion approach and FL-Cohort, mentioned above.

3. Preliminaries

This section introduces preliminary and essential background concerning FL and clustering approaches. First, we briefly present the theoretical basis of FL (Section 3.1). Secondly, we introduce the clustering techniques used in our methods, i.e., k-medoids and Markov clustering. Third, the Silhouette Index cluster validation method and the Eccentricity Analysis are explained (see Section 3.2). Finally, we describe the Kendall's Tau Rank correlation used to compare the performance of the FL frameworks investigated.

3.1. Federated learning basis

In the FL setup, a group of distributed clients cooperatively train a global model $\mathcal{M}$ under the supervision of a central server, without releasing their private data. Suppose W is a set of all clients, and there is a subset of clients W_t, (i.e., $W_t \subset W$) participating in the FL system. Each client $w_i \in W_t$ has a private training dataset

$\mathcal{D}_i = \{(x_1^{w_i}, y_1^{w_i}), \dots, (x_{n_i}^{w_i}, y_{n_i}^{w_i})\}$,

where $x_j^{w_i} \in \mathbb{R}^P$ is the P-dimensional vector representation of the jth data sample, $y_j^{w_i} \in \{1, 2, \dots, k\}$ is the corresponding label of $x_j^{w_i}$, and n_i is the size of dataset $\mathcal{D}_i$ (i.e., $|\mathcal{D}_i| = n_i$).

The FL framework is depicted in Fig. 1. The main definition of FL given by the naive FedAvg algorithm [1] can be summarized as follows:

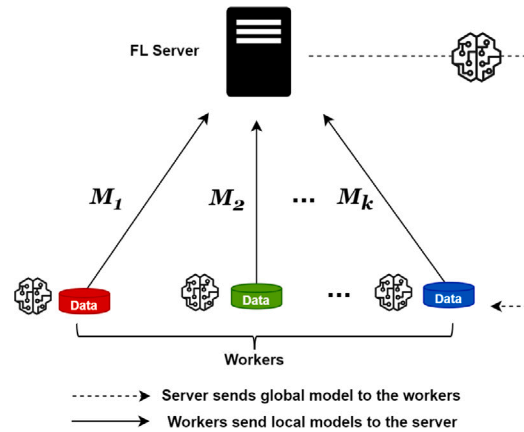


Fig. 1. Framework of the federated learning.

1. The central server randomly selects a set of several clients W_t (e.g., from a set W) to obtain the appropriate accuracy level of the learned global model $\mathcal{M}$ for several communication rounds.
2. The server sends a shared machine learning model $\mathcal{M}_t$ to the selected clients (initialized to $\mathcal{M}_0$ at the beginning).
3. Each client $w_i \in W_t$, for $t = 0, 1, \dots$, updates the model based on the local data $\mathcal{D}_i$ according to Eq. (1)

$$\mathcal{M}_{t+1}^i = \mathcal{M}_t^i - \eta g(\mathcal{M}_t^i), \tag{1}$$

   where the superscript i indexes the client, η is the learning rate and $g(\mathcal{M}_t^i)$ refers to the stochastic gradient computed based on the local dataset $\mathcal{D}_i$ of the w_ith client, calculated as

$$g(\mathcal{M}_t^i) = \frac{1}{N_{w_i}} \sum_{(x_j^{w_i}, y_j^{w_i}) \in \mathcal{D}_i} \nabla \ell(x_j^{w_i}, y_j^{w_i}; \mathcal{M}_t^i). \tag{2}$$

4. Each client $w_i \in W_t$ sends the local model $\mathcal{M}_t^i$ to the central server.
5. The central server aggregates these model parameters to update the global model according to Eq. (3)

$$\mathcal{M}_{t+1} = \mathcal{M}_t + \frac{\sum_{w_i \in W_t} p_i \mathcal{M}_{t+1}^i}{\sum_{w_i \in W_t} p_i}, \tag{3}$$

   where p_i is the relative weight of client w_i.
6. The central server S sends the aggregated model $\mathcal{M}_{t+1}$ to each client $W_t \subset W$.
7. The above steps are iterated within a predetermined number of multiple global rounds T.

3.2. Clustering approaches

Clustering techniques allow to group of data points into k disjoint groups without sufficient prior information [46]. Many clustering methods compute the similarity between all pairs of data points. In specific, the clustering algorithms work by minimizing within-group variances while maximizing them between groups relying on a clustering criterion.

3.2.1. k-Medoids

k-medoids method [47] is a variant of k-means that involves actual data points as the cluster centroids instead of derived ones. It is summarized as follows: given an integer k and an initial finite set containing n clients $W_t = \{w_1, \dots, w_n\}$, $1 \leq k \leq n$.

The problem of the k-medoids algorithm is to find a set of objects (here clients) as medoids $O = \{o_1, \dots, o_k\}, |W_t| = k$ as the center of a cluster [48] from within the set W_t, such that W_t can be partitioned into k non-empty disjoint clusters $c_1, \dots, c_k$ such that the overall sum of distances from every client w_i in the set to its nearest medoid o_j associated with its cluster is minimized.

$$\min_{O \subset W_t} \left\{ \sum_{i=1}^{n} \min_{j \in \{1,\dots,k\}} d(w_i, o_j), \mid O \mid = k \right\}. \tag{4}$$

In our CA-FL algorithm we use k-medoids for partitioning the available clients into groups of similar clients w.r.t. their local model updates. The Euclidean distance as a dissimilarity measure is adopted in the CA-FL algorithm.

3.2.2. Markov cluster

Markov Cluster (MCL) is an iterative process consisting of two steps called expansion and inflation, which are applied to a stochastic “Markov” matrix [49]. The expansion step is responsible for making the flow to connect different regions of the graph and the inflation step is responsible for both strengthening and weakening of current region. Suppose we have a matrix $M(n \times n)$ of the smoothed vectors $\hat{p}_t(w_i)$ which represents a distance matrix of each pair of clients $w_i \in W_t$ and generated by rescaling each column of the adjacent matrix $A(n \times n)$ from a graph. This matrix is then passed as an input parameter to the predicted function of a Markov clustering method as given in Eq. (5).

$$M(i,j) = \frac{Adj(i,j)}{\sum_{k=1}^{n} Adj(k,j)}. \tag{5}$$

where Adj is the adjacent matrix with self-loops. The expansion and the inflation steps are repeated until the matrix M is convergent [50]. In our GP-FL algorithm, a Markov cluster algorithm (MCL) is used to group a set of clients based on their empirical probability vectors. An initial clustering $C_t = \{C_{t1}, C_{t2}, \dots, C_{tk}\}$ of the clients is generated. The Wasserstein distance will be used to measure the distance between empirical probability distribution vectors.

3.3. Silhouette index

Silhouette score $s(i)$ is an unsupervised method for interpretation and validation of consistency within clusters of data points [51]. It is a combination of two ideas: cohesion (how close a data point is to other points in its own cluster) and separation (how far a data point is from points in other clusters) of the data point.

For data point $i \in C_j$ (data point i in the cluster C_j), we compute

$$a(i) = \frac{1}{\mid C_j \mid -1} \sum_{\substack{j \in C_j \\ i \neq j}} d(i,j), \tag{6}$$

Here $a(i)$ represents the average dissimilarity of data point i from all other data points in the same cluster C_j.

Then we define the average dissimilarity of data point i to all data points in some cluster C_r (where $r \neq j$). For each data point $i \in C_j$, we define

$$b(i) = \min_{r \neq j} \frac{1}{\mid C_r \mid} \sum_{j \in C_r} d(i,j), \tag{7}$$

to be the average dissimilarity of data point i from all data points in any other cluster (i.e., in any cluster of which i is not a member). The silhouette value $s(i)$ of one data point i, is calculated as:

$$s(i) = \begin{cases} 1 - a(i)/b(i) & \text{if } a(i) < b(i), \\ 0 & \text{if } a(i) = b(i), \\ b(i)/a(i) - 1 & \text{if } a(i) > b(i), \end{cases}$$

which can be rewritten as

$$s(i) = (b_i - a_i)/\max\{a_i, b_i\}. \tag{8}$$

3.4. Eccentricity analysis

AutoCloud [52] is an evolving clustering algorithm based on eccentricity data analytics principles [53] to autonomously create groups and merge them. Alike to AutoCloud model, eccentricity analysis was applied in our proposed GP-FL solution to update a dynamic clustering of clients. On this basis, the eccentricity ξ^j of a client w_i with respect to cluster C_j is [52]:

$$\xi^j(w_i) = \frac{1}{n_j} + \frac{(\mu_i^j - \hat{p}_i)^T(\mu_i^j - \hat{p}_i)}{n_j \sigma_i^j}, \tag{9}$$

where n_j is the size of C_j, $w_i \in C_j$, $\hat{p}_i$ is empirical probability vector held by the client w_i, and μ_i^j and σ_i^j are the mean and variance, respectively. In Eq. (10), the condition is verified to determine whether a particular client is considered to belong to a cluster C_j using Chebyshev inequality [54] as follows

$$\xi^j(w_i) \leq v_j \text{ and } v_j = (m^2+1)/2n_j, \tag{10}$$

where m is a pre-defined parameter that straightly impacts the clustering evaluation, and v_j is the threshold associated with the corresponding cluster C_j. $m = 3$ is broadly set as a standard value, thus achieving appropriate results for various datasets [55].

3.5. Kendall’s Tau Rank correlation

Kendall’s Tau Rank Correlation, also commonly known as Kendall correlation or Kendall’s tau, is one of the standard correlation methods used to measure the consistency among ranked attributes. The Kendall’s tau correlation coefficient can take values from −1.0 to 1.0. The positive values close to 1.0 represent the positive correlation between ranked variables, the negative values close to −1.0 indicate a negative relation, and the values close to zero imply no correlation [56]. If two rankings x and y are produced on the given clients, then Kendall’s tau $\mathcal{T}$ can be calculated using Eq. (11) [57].

$$\mathcal{T} = \frac{A - D}{\sqrt{(A + D + T_x)(A + D + T_y)}}, \tag{11}$$

where, A is the number of pairs in agreement, D is the number of pairs in disagreement, T_x is the number of pairs tied w.r.t. x, and T_y is the number of pairs tied w.r.t. y.

We evaluate and compare the FL baseline approaches by computing Kendall’s tau correlations between the ranking scores of the CA-FL, GP-FL, Deletion approach, and FL-Cohort.

4. Methodology

4.1. Problem statement

Assume a set of clients, W, forms a federation for training a global model. Each client w_i, $w_i \in W$, trains at each training round t, $t = 0, 1, \dots$, an ML model $\mathcal{M}_t^i$ locally on its data set and sends the model parameters to the central server, where a unified global model $\mathcal{M}_t$ is created by averaging the model parameters received from all the clients. We assume that all clients participate in all rounds of the federation. In this context, we want to evaluate the clients’ behavior during the distributed model training process. The aim is to quantify each client’s reliability, defined as the client’s attitude/capacity to provide/train high-performing local models for the majority of the training rounds. Those evaluations should provide an opportunity for comparing, understanding, and eventually predicting clients’ contributions to the shared global model. However, the existing approaches are resource and time-consuming and unsuitable for a real FL scenario where resource consumption is the main concern [58,59]. Meanwhile, the evaluation of the client’s influence on the federated model has attracted a lot of attention in recent years due to the need to provide an incentive mechanism. Most existing approaches usually assess participants’ contribution to the global model when a single or a group of clients are not involved in the training. For example, the Deletion approach [7] works by retraining the federated model without the contribution of a given client's data and measuring the retrained model performance in comparison to that of the model trained on the data of all the clients to evaluate this client's contribution to the overall federated model. The FL-Cohort [8] conducts the same procedure, but instead of removing a single client at a time, it eliminates a group of similar clients at each FL global training round. However, these approaches still face several issues. First, an additional evaluation procedure is needed to evaluate the client's contribution, leading to increased time costs. Second, the evaluation process for each client or group of clients significantly increases resource wastage [7,60]. On the other hand, our proposed approaches introduce effective FL solutions to predict clients' contributions by evaluating the clients' behavior during the training process. The latter is evaluated by using the FL algorithm working mechanism, i.e., client's behavior can be interpreted as reliable or unstable due to this.

4.2. Deletion approach

Naively, the Deletion approach [7] aims to assign the contribution to a participant client by calculating the model performance change when the client is erased from the set of participants selected in FL learning. Thus, this approach can exclude the contribution of certain clients to the overall global model. Suppose we evaluate the effect of the client w_i on the model predictions, the influence measure [7] can be formulated as:

$$influence^{-i} = \frac{1}{n}\sum_{j=1}^{n} |\hat{y}_j - \hat{y}_j^{-i}|, \tag{12}$$

where n is the size of the dataset, $\hat{y}_j$ is the prediction on jth instance made by the model trained on all data, and $\hat{y}_j^{-i}$ is the prediction on jth instance made by a model trained with the ith client omitted.

Given the initial global model $\mathcal{M}_0$ and a target client u, whose private data is required to be erased from the FL model, Deletion approach steps are summarized into Algorithm 1. In line 2, the server sends the global model $\mathcal{M}_t$ to all selected clients except the target client $w_i \in W_t \backslash u$ from global training. The global model $\mathcal{M}_t$ is sent to a set of participants, and each client $w_i \in W_t \setminus w_u$ optimizes the model locally according to Eq. (1). Then, each client uploads the initial update to the central server (line 4). Later, the server averages the model parameters uploaded to update the global model $\mathcal{M}_t$ according to Eq. (3). In line 7, an unlearned model $\hat{\mathcal{M}}_{t+1}$ is returned for the client w_u during t round of training. Finally, in line 8, we calculate the influence of the client w_u on the global model according to Eq. (12).

4.3. FL-cohort

In contrast to the Deletion approach, the FL-Cohort [8] removes multiple, similar clients on each round to quantify the contribution of a client to the FL setting. Algorithm 2 presents the FL-Cohort process. In FL-Cohort, firstly, there are a bunch of clients $W_t \in W$. Each client $w_i \in W$ sends a vector $\vec{v}_i = [N_i^1, \dots, N_i^Q]$, where Q is the number of classes and N_i^q indicates the number of samples for class q held by client w_i to measure the imbalance (line 2 in the Algorithm 2). On the server side, $\vec{V} = [\sum_i N_i^1, \dots, \sum_i N_i^Q]$ and the overall number of samples N are calculated using SA protocol. Then, the server announces the global measurement to all clients (lines 3 ∼ 4 in the Algorithm 2). Subsequently, $\vec{V}$ and N are distributed to each of the clients for calculating data imbalance within the cohort.

In FL-Cohort, data imbalance metrics, namely Label Imbalance (LI), Label Distribution Imbalance (LDI), and Quantity Imbalance (QI) are computed at each local party. In brief, these metrics are defined and computed as follows:

Label Imbalance (LI): LI of each client w_i is defined as follows:

$$LI_i = \frac{max_p\{N_i^p\}}{min_p\{N_i^p\}} \tag{13}$$

Algorithm 1 Deletion Approach

Output: Deletion Approach procedure unlearned model $\hat{\mathcal{M}}_{w_i}$ and Influence ^{i}

1: **procedure** DeletionApproach($\mathcal{M}_0$, $W_t \subseteq W$, Target client with index u, T number of iterations)
2: $\forall\ w_i \in W_t$ except client w_u, Send(w_i, $\mathcal{M}_t$)
3: **for** each client $w_i \in W_t \setminus w_u$ *in parallel* **do**
4: $\mathcal{M}^i_{t+1} \leftarrow$ clientUpdate($i, \mathcal{M}_t$)
5: **end for**
6: $\hat{\mathcal{M}}_{t+1} = \sum_{w_i \in W_t \backslash u} \frac{n_i}{n} \mathcal{M}^i_{t+1}$ following Eq. (3) ▷ *Unlearned model*
7: Influence $^{-u} = \frac{1}{n}\sum_{j=1}^{n} |\hat{\mathcal{M}}_j - \hat{\mathcal{M}}_j^{-u}|$ ▷ *According to Eq. (12)*
8: **return** Influence ^{i} for $\forall\ w_i \in W_t$
9: **end procedure**
10: **function** clientUpdate(($w_i, \mathcal{M}_t$)) ▷ *Local update*
11: **while** True **do**
12: Receive(w_i, $\mathcal{M}_t$) ▷ Receive the global model
13: LocalTraining($w_i, \mathcal{M}_t$)
14: $\mathcal{M}^i_{t+1} \leftarrow \mathcal{M}^i_t - \eta g^i_t$ ▷ *Following Eq. (1)*
15: Send(i, $\mathcal{M}^i_{t+1}$) ▷ Send the local model to the server
16: **end while**
17: **end function**

Therefore, LI_i is the ratio of the majority class size and minority class size for a local client i.

Label Distribution Imbalance (LDI): $\vec{V}$ and $\vec{v}_i$ are used to quantify imbalance in terms of label distribution. So, LDI is given as follows:

$$LDI_i = 1 - \frac{\vec{v}_i \cdot \vec{V}}{\|\vec{v}_i\| \cdot \|\vec{V}\|} \tag{14}$$

The cosine distance is utilized to measure of similarity between two vectors. As a result, LDI_i= 0 refers to perfect balance, whereas LDI_i= 1 confirms a very high imbalance.

Quantity Imbalance (QI): QI is defined as the ratio of the number of samples at client i and the mean number of samples overall i clients as:

$$QI_i = N_i / \frac{\sum_{l=1}^{I} N_l}{I} \tag{15}$$

Each client is represented by a vector $\vec{l}_i = [LI_i, LDI_i, QI_i]$. Then, the updated vector $\vec{l}_i$ of each client is sent back to the server, line 6 in Algorithm 2. Eventually, the server applies the k-means algorithm on the set of representations. As a result, k_t clusters of clients with similar imbalances are obtained (line 8 in Algorithm 2). In order to quantify the contribution of each client, FL-Cohort approach applies a leave-x-out (LXO) analysis, which iteratively deletes multiple and similar clients on each global training. Specifically, during LXO analysis, a model $\mathcal{M}_c^{LXO}$ leaves out the respective cluster c_j is trained, line 10. Lastly, the contribution of each cluster $c_j \in C_t$ is defined using a weighted F1 scores as follows [60]:

$$Cr_{c_j} = F1(\mathcal{M} - D^{Test}) - F1(\mathcal{M}_c^{LXO} - D^{Test}) \tag{16}$$

To assign a contribution cr_i for each client $w_i \in c_j$, simply the contribution of their respective cluster c_j is divided by the number of clients W_{c_j} belonging to c_j, line 12.

Steps (lines 2∼12 in the Algorithm 2) occur prior to the starting of FL training.

4.4. Proposed approach

In this section, we present a simple and effective FL framework for clients behavior evaluation based on two FL algorithms: CA-FL [9] and

Algorithm 2 FL-Cohort framework

Output: The FL-Cohort procedure Contribution $Cont_{w_i}$ of each $w_i \in c_j$ $(j = 1, 2, \dots, k)$

procedure FL-Cohort($\mathcal{M}_0$, $W_t \subseteq W$, Target cohort c_i, T)
  $\forall\ w_i \in W_t$, Send(w_i, $\vec{v}_i$)
  $\vec{V} \leftarrow [\sum_i N_i^1, \dots, N_i^Q]$
  $\forall\ w_i \in W_t$, Send(w_i, $\vec{V}$, N)
  for each $w_i \in W_t$ **do**
    $\vec{l_{i+1}} \leftarrow$ clientUpdate(i) ▷ Defined in Alg. 1
  end for
  $C_t \leftarrow$ k-means(k_t, $\{\vec{l_i} \mid w_i \in W_t\}$, W_t) prior to FL training
  $\bar{C}_t \leftarrow \forall\ c_{tj} \in C_t, \{C_t \setminus c_{tj}\}$
  $\mathcal{M}_{c_j}^{LXO} = \sum_{w_i \in \bar{C}_t} \frac{n_i}{n} \mathcal{M}_{t+1}^i$
  $Cr_{c_j} \leftarrow (\mathcal{M}, D^{Test}, \mathcal{M}_c^{LXO})$
  $Cr_{w_i} \leftarrow (Cr_{c_j}, W_{c_j})$
  $t \leftarrow 0$ ▷ *FL training*
  while $t \leq T$ **do**
    $t \leftarrow t + 1$
    $\forall\ w_i \in W_t$, Send(w_i, $\vec{V}$, N)
    for each client $w_i \in W_t$ *in parallel* **do**
      $(\mathcal{M}_{t+1}^i, \vec{l_i}) \leftarrow$ clientUpdate($i, \mathcal{M}_t$)
    end for
  end while
end procedure
return cohort model $\mathcal{M}_{c_j}^{LXO}$

GP-FL [10]. Namely in this paper, we utilize the advantages of the CA-FL and GP-FL approaches to predict FL clients' contributions to the built global model by evaluating their behavior.

The CA-FL is primarily focused on reducing communication overhead in FL but can also be used to identify clients with reliable behavior throughout the global training. In the CA-FL, a client is considered more reliable when providing a high-performance model. If a client is selected frequently as a cluster representative, i.e., it provides a highly performing local model, this client is interpreted by our approach as reliable. On the other hand, the GP-FL algorithm proposes group-personalized federated models that can detect clients with unstable behavior. If a client very often changes a cluster it belongs to during training, this client has unstable behavior. The strong side of these approaches is performance. First, they involve training a global model with a set of clients while evaluating the clients' behavior simultaneously during the training. Second, they do not incur large computation resources and time compared to existing approaches. The computation costs of the four studied approaches are discussed in Section 6.1.

The CA-FL [9] algorithm calculates the client frequency of being selected as a cluster representative. In specific, at each training round, a client with a top-performing model is selected from each cluster. Based on this, each client will be assigned a score that can be interpreted as a measure of the client's reliability. Certainly, the higher score implies a higher influence (contribution) to the built global model. These scores can be used for ranking the clients with respect to their reliability, i.e., contribution to the global model. This ranking can be used in different ways, e.g., for selecting a given percentage of the top-ranked clients and training a new global model on these clients' data that ensures higher performance. In that way, we can minimize the impact of non-reliable clients on the shared global model. Based on the produced ranking, it is also possible to split the clients into three disjoint groups representing respectively top contributing clients, ones with comparatively good contributions, and finally ones with very modest, almost no contributions. Three separate FL models can be built on these three groups of clients. This will contribute to the fairness of the FL scheme by providing each client with a model that best reflects its contribution to the federation in general. In addition, the availability of three different FL (global) models will also facilitate the scenarios when a client wants to withdraw its participation from the federation: namely, only the affected model should be retrained in this case.

Opposite to the CA-FL, the GP-FL [10] method provides an opportunity for identifying clients exposing unstable behavior during the training process. The GP-FL initially splits the clients into a few clusters based on the similarity between their class distributions. This clustering is dynamically maintained during the training process by evaluating at each training round the clients' assignment among the clusters, and if necessary some are re-assigned to the new clusters. To evaluate the clients' behavior, we calculate how many times each client changes a cluster it belongs to during the distributed training process, which can be interpreted as a measure of the instability of the client's data, i.e., it can be translated to the client's unreliability.

Note that different criteria can be applied to split the clients into three (or even more) groups (highly reliable clients, clients with good level of reliability, and unreliable clients), with respect to the scores produced by the CA-FL or the GP-FL, e.g., splitting thresholds can be defined based on the analysis of the scores or binning can be applied.

Binning is a technique used to discretize a continuous variable's values into bins (groups) [61]. In our experiments, we use this method to split clients into groups, both for the CA-FL and GP-FL. In addition, the calculated scores can be normalized, e.g., by applying min–max normalization, to facilitate in this way the interpretability of the scores.

The performance of the two algorithms, CA-FL and GP-FL, is studied under different experimental scenarios and they are additionally compared with two existing approaches (Deletion [7] and FL-cohort [8]) that are specially devoted to the clients' contribution evaluation in FL context. The correlations between the rankings produced by our algorithms and those based on the Deletion and FL-cohort approaches are studied by Kendall's tau, see Section 5.

The two FL methods, discussed above, are described in Section 4.4.1 and Section 4.4.2, respectively.

4.4.1. CA-FL

Algorithm 3 contains the pseudo-code of the proposed CA-FL phases [9]. In the Initialization phase (lines 2–6 in Algorithm 3) the central server sends the global model $\mathcal{M}_t$ to a subset of clients W_t $(W_t \subset W)$ (line 3).

Each client $w \in W_t$ receives the global model $\mathcal{M}_t$ and updates its local model to the shared model (line 4). At line 5, clients upload their trained local models $\{m_w^t \mid w \in W_t\}$ to the server, which then generates the new global model $\mathcal{M}_{t+1}$ through the FedAVG algorithm following Eq. (3). of all received local models.

In the Iteration phase (lines 7–14 in Algorithm 3), firstly, the server ranks the clients in each cluster $C_i^t, i = 1, 2, \dots, k$ and selects the top-ranked client, i.e., the representatives (line 9). In line 10, the frequency for each client being of a cluster representative is calculated. The selected representatives form a new set of clients $W^{t+1} = \{w_1^{t+1}, w_2^{t+1}, \dots, w_k^{t+1}\}$, where $k \ll \mid W^0 \mid$. Then, the server sends the global model M_t to each representative $w \in W^{t+1}$ for new global training. Each representative $w \in W^{t+1}$ receives the global model $\mathcal{M}_t$ and sent back m_w^{t+1} local update according to Eq. (1) to the server after train its local model locally. The Server averages the local models $\{m_w^{t+1} \mid w \in W^{t+1}\}$ uploaded by the representatives to update the global model $\mathcal{M}_{t+1}$. Finally, the server applies the Silhouette score (SI) to assess whether each representative is still well tied to its current cluster (Eq. (8)), i.e. the updated clustering C^{t+1} of the clients in W^0 is produced.

Steps at lines 8–14 are iterated until the predetermined number of communication rounds T is completed.

Algorithm 3 CA-FL framework

Output: The CA-FL procedure the frequencies of representatives $freq_t$ for T iterations

procedure CA-FL($\mathcal{M}_0$, $W_t \subseteq W$, k_t, T)
Initialization Phase
$t \leftarrow 0$
$\forall\ w_i \in W_t$, the server exec Send(w_i, $\mathcal{M}_t$)
Each $w_i \in W_t$ exec clientUpdate($i, \mathcal{M}_t$) ▷ Defined in Alg. 1
$\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$ ▷ According to Eq. (3)
$C_t \leftarrow$ Kmedoids(k_t, $\{\mathcal{M}^i_{t+1} \mid w_i \in W_t\}$, W_t)
Iteration Phase
while $t \leq T$ **do**
$t \leftarrow t+1$
$W_t \leftarrow$ SelectTopRanked(p, C_t)
$freq_t \leftarrow$ frequency(W_t) ▷ *Calculate frequency for each client of being of a cluster representative*
$\forall\ w_i \in W_t$, the server exec Send(w_i, $\mathcal{M}_t$)
Each $w_i \in W_t$ exec clientUpdate($w_i, \mathcal{M}_t$) ▷ Defined in Alg. 1
$\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$
$C_{t+1} \leftarrow$ Silhouette(k_t, C_t, W_t)
end while
end procedure
function Silhouette((k_t, C_t, W_t)) ▷ *Check whether each $w_i \in W_t$ still belongs to its cluster*
for $w_i \in W_t$ **do**
for $j = 1, 2, \ldots, k$ **do**
compute $s(w_i)$ ▷ *According to Eq.* (8)
end for
if $s(w_i) < 0, \forall j \in \{1, 2, \ldots, k\}$ **then**
$k_t \leftarrow k_t + 1$
$C_{tk_t} \leftarrow w_i$
else
Assign w_i to the nearest cluster C_{tj}
end if
end for
$\forall\ C_{tj}$ $(j = 1, 2, \ldots, k)$ recompute the cluster center
return C_{t+1} ▷ *The new set of clusters*
end function

4.4.2. GP-FL

The details of our GP-FL scheme [10] are shown in the Algorithm 4. In the Initialization phase (lines 2–9 in the Algorithm 4) the server initializes model $\mathcal{M}_t$ and the set of clients $W_t \subset W$. Then (line 3), the server sends the new global model $\mathcal{M}_t$ to each client $w_i \in W_t$. Each client receives the global model and optimizes model parameters locally, executing the clientUpdate function (line 4). After the training process is complete, the client produces the $\mathcal{M}^i_t$ local model alongside a vector $\hat{p}_t(w_i)$ that represents the empirical probabilities of the class distribution and is sent back to the main server. To avoid issues with extreme values, such as 0 or 1, each empirical probability vector $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k)$ can be smoothed by applying Laplace correlation (line 5) which is expressed as:

$$\hat{p}_i(D) = \frac{n_i + 1}{\mid D \mid + k}, \tag{17}$$

where D is a set of labeled examples, n_i is the number of examples in class C_i, and k (the number of classes) is added to ensure that the posterior probabilities are never zero [62]. The smoothed vectors $\hat{p}_t(w_i)$, for $w_i \in W_t$, are used to create a distance matrix (line 6) used by a Markov clustering algorithm executed by the server. To compute the distance matrix [63,64], we used Wasserstein distance. That distance allows us to measure the similarity between class probability estimations of each pair of clients. In this way, the clients who exhibit similar activity patterns will be grouped together by applying Markov Cluster, as described in Section 3.2.2. The Markov Cluster algorithm produces groups of clients with similar empirical probability vectors, i.e. an initial clustering $C_t = \{C_{t1}, C_{t2}, \ldots, C_{tk}\}$ of the clients is created. For each cluster $C_{tj} \in C_t$, $(j = 1, 2, \ldots, k)$, mean data vector μ^j_i and aggregated variance σ^j_i are calculated (line 7). Moreover, a global group model $\mathcal{M}^j_t$, is built by averaging over the model models of the clients assigned to C_{tj} (line 8).

Algorithm 4 GP-FL framework

Output: The GP-FL procedure the number of recurrences of clients $recur_t$ that changes his/her cluster

1: **procedure** GP-FL($\mathcal{M}_0$, $W_t \subseteq W$, T)
Initialization Phase
2: $t \leftarrow 0$
3: $\forall\ w_i \in W_t$ the server exec Send(w_i, $\mathcal{M}_t$)
4: Each client $w_i \in W_t$ exec clientUpdate($i, \mathcal{M}_t$)▷ Defined in Alg. 1
5: $\hat{p}_{t+1}(w_i) \leftarrow$ LaplaceSmoothingCorrelation($\hat{p}_{t+1}(w_i)$)
6: $C_t \leftarrow$ Markov($\{\hat{p}^i_{t+1} \mid w_i \in W_t\}, W_t$)
7: $\forall\ C_{tj}$ $(1 \leq j \leq k)$ compute μ^j_i and σ^j_i
8: $\forall\ C_{tj}$ $(1 \leq j \leq k)$ compute $\mathcal{M}^{C_{tj}}_{t+1} = \sum_{w \in C_{tj}} \frac{n_w}{n} \mathcal{M}^w_{t+1}$
9: $\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$
Iteration Phase
10: **while** $t \leq T$ **do**
11: $t \leftarrow t+1$
12: $\forall\ w_i \in C_{tj}$ the server exec Send(w_i, $\mathcal{M}^j_t$)
13: Each client $w_i \in C_{tj}$ exec clientUpdate($w_i, \mathcal{M}^j_t$) ▷ Defined in Alg. 1
14: $\forall \hat{p}_{t+1}$, the server update the existing $\hat{p}_{t+1}(w_i)$
15: $\hat{p}_{t+1}(w_i) \leftarrow$ LaplaceSmoothingCorrelation($\hat{p}_{t+1}(w_i)$)
16: $C_{t+1} \leftarrow$ EccentricityScore(k_t, C_t, W_t)
17: $recur_t \leftarrow$ Recurrence(W_t) ▷ *Calculate times that each client changes its cluster*
18: $\forall\ C_{t+1j}$ $(1 \leq j \leq k)$ compute μ^j_i and σ^j_i
19: $\mathcal{M}^{C_{tj}}_{t+1} = \sum_{w \in C_{tj}} \frac{n_w}{n} \mathcal{M}^w_{t+1}$ following Eq. (3)
20: $\mathcal{M}_{t+1} = \sum_{w_i \in W_t} \frac{n_i}{n} \mathcal{M}^i_{t+1}$
21: **return** (C_{t+1}, k_{t+1})
22: **end while**
23: **end procedure**

The collected local models $\{\mathcal{M}^i_t \mid w_i \in W_t\}$ are then aggregated to update the global model $\mathcal{M}_t$ through the FedAvg algorithm according to Eq. (3).

In the Iteration phase (lines 10–23) the server sends each group global model $\mathcal{M}^j_t$, $(j = 1, 2, \ldots, k)$ to its group of clients C_{tj} (line 13). Each client $w_i \in C_{tj}$, executing the clientUpdate function (line 14), receives the group global model $\mathcal{M}^j_t$ and optimizes its parameters locally, sending back to the server the local model $\mathcal{M}^i_{t+1}$ and the empirical probability vector $\hat{p}_{t+1}(w_i)$. Then the server, at line 14, updates the existing empirical probability vector $\hat{p}_{t+1}(w_i)$ by taking the average of it with the information provided by the previous data batch, i.e. $\hat{p}_t(w_i)$, and it applies Laplace smoothing to each vector $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, \mid W_t \mid$ (line 15). At line 16, the Server adapts the clients' grouping C_t to the current empirical probability vectors $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, \mid W_t \mid$, by invoking eccentricity score $\xi^j(w_i)$ following Eq. (9), $(j = 1, 2, \ldots, k)$ which assesses whether each client $w_i \in C_{tj}$ is still adequately tight with its current cluster, the one it was assigned at the previous round (t) (Cluster Optimization). The EccentricityScore pseudocode is in Algorithm 5 and is described in Section 4.4.3.

Algorithm 5 EccentricityScore

Output: updated k_t and C_t

1: **procedure** EccentricityScore(k_t, C_t)
2: **for** $j = 1, 2, \ldots, k_t$ **do**
3: **for** $w_i \in C_{tj}$ **do**
4: compute eccentricity score $\xi^j(w_i)$ ▷ *According to Eq. (9)*
5: **if** $\xi^j(w_i) < \upsilon_j(t)$ **then**
6: $w_i \in C_{tj}$ ▷ *According to Eq. (10)*
7: **else if** $\xi^j(w_i) > \upsilon_j(t)$ **then**
8: calculate $\xi^l(w_i)$ for each $C_{tl} \in C_t \setminus C_{tj}$
9: Assign w_i to C_{tl} for which $\xi^l(w_i) < \upsilon_l(t)$ and $\xi^l(w_i)$ is lowest ▷ *In case there is more than one cluster*
10: **else if** $\xi(w_i) > \upsilon_l(t)$ for each $C_{tl} \in C_t \setminus C_{tj}$ **then**
11: w_i represents a new singleton cluster
12: **end if**
13: **end for**
14: **end for**
15: **end procedure**
16: **return** (C_t, k_t)

For each cluster $C_{t+1j} \in C_{t+1}$, mean data vector μ_i^j and aggregated variance σ_i^j are calculated, considering the current grouping of the clients and also using the current empirical probability vectors $\hat{p}_{t+1}(w_i)$, for $i = 1, 2, \ldots, | W_{t+1} |$. These values of μ_i^j and σ_i^j will be needed at the next round to calculate the clients' eccentricity scores w.r.t. the current clusters.

The updated clustering C_{t+1} is produced, and the clusters in C_{t+1} may contain different clients from the clusters in C_t.

These steps of the *iteration* phase are repeated until a certain number of training rounds T is reached.

4.4.3. Eccentricity score function

The function firstly computes the Eccentricity score $\xi^j(w_i)$, using Eq. (9), for each client in a cluster (line 4). If $\xi^j(w_i)$ is below the threshold $\upsilon_j(t)$ following Eq. (10), the client does not change its cluster C_{tj} (lines 5 and 6). Otherwise, if $\xi^j(w_i) > \upsilon_j(t)$ then we calculate $\xi(w_i)$ for the other clusters, i.e. for each $C_{tl} \in C_t \setminus C_{tj}$, and will assign the client w_i to the cluster for each $\xi(w_i) < \upsilon_l(t)$. In case this is true for more than one cluster we will assign the client to the cluster for which the score is lowest. Finally (line 11), If $\xi(w_i) > \upsilon_l(t)$ for each cluster $C_{tl} \in C_t \setminus C_{tj}$, then this client w_i will give the start of a new singleton cluster, which means that this client w_i cannot be assigned to any existing cluster in C_t. Note that $k_{(t+1)} \geq k_t$, where $k_{(t+1)} =| C_{t+1} |$, since new singleton clusters may appear due to the updating operation.

5. Experimental setup

This section presents the experimental settings.

5.1. Dataset

We conduct experiments on three LEAF datasets [28] which are tailored to federated learning evaluation: Synthetic Dataset [11], Federated Extended MNIST (FEMNIST for short) [12] and CelebA [13].

- Synthetic: This dataset modifies the synthetic dataset presented in [11] to make it more challenging for current meta-learning methods. For this dataset, we use logistic regression as a basic ML model.
- FEMNIST [12]: This dataset is built by partitioning the data on the Extend MNIST based on the writer of the digit/character. For this dataset, we use a convolutional neural network (CNN) containing two convolutional layers and three fully connected layers.
- CelebA [13]: This dataset is a large-scale face attributes' dataset with celebrity pictures. For this dataset, we have used the same CNN network as for the FEMNIST dataset.

Table 1
Statistics of the used LEAF datasets.

Dataset	Number of devices	Samples per devices	
		Mean	Stdev
Synthetic	1000	107,55	213,22
FEMNIST	3500	226.26	89.12
CelebA	9343	21.44	7.63

Table 2
Default notation, definitions, and corresponding values of different hyper-parameters.

Notation	Definition	Value
W	Total number of clients	100
W_t	Fraction of clients randomly selected	0.2
η	Learning rate	0.05
T	Number of communication rounds	—
$\mathcal{D}_i$	The local data in client w_i	—
n_i	The size of data in client w_i	—
n	Total size of data	—
$\mathcal{M}_t$	The global model at tth round	—
$\mathcal{M}_t^i$	The local model of client w_i at round t	—
g_t^i	The gradients computed	—
$s(.)$	Silhouette Index score	—

Table 1 summarizes the statistics of the three used datasets.

For all datasets, we use the default division in Pytorch. The entire test dataset is used as a common test dataset for model performance evaluation, and half of the test dataset is used as unlabeled data on the central server. To reproduce the experimental setup of federated learning, we need to partition the dataset to simulate individual clients. Each client may have fewer (or even no) data samples in some classes.

The baseline algorithms used in this paper to assess the robustness of our proposed methods are explained in Section 4.2 and Section 4.3, respectively.

Accuracy and F1 score are the classification metrics used to evaluate the performance of our methods. The reliability/ unreliability of the clients is evaluated by ranking them based on the scores calculated by our FL algorithms. This ranking is also used to correlate the performance of our approaches with the baselines applying Kendall's tau.

5.2. Experimental details

We have implemented all the code using PyTorch with various settings, including client local data training with SGD. Two ML models are used in our experiments: logistic regression (LR) and Convolutional Neural Network (CNN). We have considered two different ML tasks on the three used federated benchmark datasets: logistic regression on a synthetic dataset and image classification for FEMNIST and CelebA datasets. The image classifier in all experiments is a standard CNN, which consists of two convolutional layers and one fully connected (FC) layer. Furthermore, for synthetic dataset, we use LR. Table 2 contains some necessary notations and values of different hyper-parameters used in this paper. In our experiments, we have evaluated the proposed methods and compared their performance to that of the two baseline solutions introduced in Sections 4.2 and 4.3 on IID and Non-IID data with 20 clients simulated in the three used datasets described above.

6. Evaluation and results

In this section, we elaborate and discuss the results of experiments. We compare the performance of our two methods, CA-FL and

GP-FL, with that of the two baselines, Deletion approach [7] and FL-Cohort [8], in different experimental settings in terms of their ability to evaluate the client's contributions to the shared global model.

The section is organized as follows. The contribution evaluation costs of the four compared approaches are discussed in Section 6.1. The results of the evaluation of the methods' performance are analyzed in Section 6.2. Section 6.3 is devoted to the analysis of the ranking correlations identified among the approaches. The performance of the CA-FL and GP-FL are further evaluated in Sections 4.4.1 and 4.4.2, respectively.

6.1. Contribution evaluation costs

We define $\mathcal{M}_t$ as a global model obtained after several communication rounds T, where W_t represents the set of clients selected to participate in FL training. In addition, the measure for assessing the client contribution $w_i \in W_t$ is the effect on the global model performance when the client participates in the FL. The influence of each client can be specified as the relative performance of a global model that includes all participants but erases only client w_i. In particular, the client influence is measured by the formula in Eq. (12), which is intuitive and calculable. However, calculating the client i contribution/influence demands extra cost to train a global model with all selected clients except the target client i. The cost $cost^{comp}$ of calculating the clients' contribution measures can be formulated as [65]

$$cost^{comp} = \sum_{w_i \in W_t} T(server^{comp} + \sum_{u \in W_t \setminus w_i} u^{comp}), \tag{18}$$

where T is the total number of communication rounds at the end of the FL process, $server^{comp}$ is the computation cost at the server for model aggregation, and u^{comp} is the number of computation operations for model updating at client u. The traditional FL process includes the sum of computations of model aggregation and model updates at each client $w_i \in W_t$ for T times. To calculate the measure of each client w_i, the FL process with $W_t \setminus w_i$ must be performed one by one. Thus, extra computation costs are required. The computation and communication costs can be huge, especially when the number of clients and the models transferred increase. We predict the client's contribution based on the behavior evaluation of each client. This is based on the intuition that the best-performance clients will also improve the final overall model performance and vice versa. Thus, the contribution/influence can be calculated in our proposed approaches during T rounds without requiring additional FL rounds. The computation cost caused by our proposed FL approaches is calculated as

$$cost^{comp} = \sum_{w_i \in W_t} T(server^{comp} + freq_{w_i}), \tag{19}$$

where $freq_{w_i}$ is the cost needed to calculate the frequency for each client w_i of being a cluster representative in the case of CA-FL. In the case of GP-FL, the recurrence $recur_{w_i}$ for each client that changes its cluster is calculated instead of $freq_{w_i}$. Such calculations are done on the server side, during the training phase, and their computational cost is much less than computing the $\sum_{u \in W_t \setminus w_i} u^{comp}$ term of Eq. (18). Hence, the cost of computing the contribution/influence in the CA-FL and GP-FL is much lower than in the traditional approach.

6.2. Comparison of methods' performance

Each of the four studied approaches produces a ranking of the clients scoring their influence on the global model by applying its own evaluation mechanism. To compare the performance of the four approaches we have initially applied each approach to each of the three datasets used to calculate a single evaluation for each client. Notice that the meaning/interpretation implemented in the calculated clients' evaluations differs for different approaches. For example, the two baseline approaches consider those as measures of clients' contribution (influence) to the global model. While the CA-FL (refer to Algorithm 3 (line 10)) and GP-FL (see Algorithm 4 (line 17)) interpret the calculated scores as evaluations of clients' behavior (reliable versus unreliable), i.e., their ability to have a positive/negative impact to the global model eventually.

Table 3
The clients' contributions calculated by the Deletion approach on Synthetic, FEMNIST and CelebA datasets in Non-IID setting. For each dataset the scores of the five top contributing clients are highlighted in red, while the lowest scores are printed in blue.

Client	Synthetic	FEMNIST	CelebA
1	0.031	0.031	0.031
2	0.035	0.037	0.038
3	**0.053**	0.041	**0.049**
4	0.035	0.033	0.031
5	0.039	0.039	0.033
6	0.039	0.036	**0.055**
7	**0.059**	**0.061**	0.040
8	0.042	**0.055**	0.036
9	0.037	0.038	0.035
10	**0.055**	0.037	**0.053**
11	0.039	0.041	0.037
12	0.036	**0.057**	0.032
13	**0.056**	**0.060**	0.035
14	0.038	0.032	**0.050**
15	0.039	0.037	0.033
16	0.037	0.034	0.034
17	0.035	**0.059**	0.034
18	0.040	0.033	0.034
19	0.036	0.041	0.031
20	**0.056**	0.037	**0.054**

Table 4
The contribution of each cluster in FL-Cohort approach on Synthetic, FEMNIST, and CelebA datasets.

Cluster No.	size	Synthetic	FEMNIST	CelebA
1	6	0.3504	0.380	0.3677
2	7	0.2861	0.3432	0.344
3	5	0.2694	0.1873	0.2096
4	2	0.0941	0.0895	0.0787

In the case of IID, since each client has the same number of identically distributed data, the local models are expected to get equivalently updated. The global model and all the local models are expected to converge to a maximum accuracy value simultaneously. In Non-IID data, the training performance may vary significantly due to the non-balanced local data. Thus, the local models are expected to significantly vary from each other. In our experiments, for each client, each dataset is partitioned into a training set and a test set. The test set is used for the local evaluation of each client's performance. These evaluation results are then aggregated to give the local performance evaluation of each client. Table 3 shows the calculated contributions of the clients by the Deletion method on the three different datasets. The scores of the five top contributing clients in each of the three datasets are marked red, while the lowest scores are printed blue.

We compare those with the results produced by the FL-Cohort approach and presented in Tables 4 and 5, respectively. Table 4 shows the contribution of each cluster in FL-Cohort on different datasets.

It can be seen that cluster 1 has achieved the highest scores for all three datasets Synthetic, FEMNIST, and CelebA, 0.3504, 0.380, and 0.3677, respectively. The worst-performing cluster producing the lowest scores for all datasets is cluster 4. Table 5 lists the members of each cluster produced by the FL-Cohort with their contribution values.

Each client in a cluster makes the same contributions as the other clients that belong to the same cluster. It is interesting to compare the overlap between cluster 1, i.e., the top-contributing clients according to FL-Cohort, and the top-performing clients identified by the Deletion approach in the three datasets. One can notice that all the top five clients identified by the Deletion approach in Synthetic dataset belong to cluster 1 while for the other two datasets some of the clients are

Table 5
The contribution of each client in FL-Cohort approach on the datasets of Synthetic, FEMNIST, and CelebA in Non-IID setting.

Cluster	clients	Synthetic	FEMNIST	CelebA
1	3, 7, 8, 10, 13, 20	0.0584	0.0633	0.0612
2	2, 5, 6, 9, 11, 14, 15	0.0408	0.0490	0.0491
3	12, 16, 17, 18, 19	0.0538	0.0374	0.0419
4	1, 4	0.0470	0.0448	0.0393

Table 6
The clients' evaluations produced by the proposed CA-FL approach on Synthetic, FEMNIST and CelebA datasets in IID setting. For each dataset the scores of the five top contributing clients are highlighted in red, while the lowest scores are printed in blue.

Client	Synthetic	FEMNIST	CelebA
1	1.4	1.2	**1.5**
2	1.2	1.3	**1.6**
3	1.2	**1.5**	**1.6**
4	1.2	1.1	**1.5**
5	**1.6**	1.3	1.4
6	1.3	1	1.2
7	**1.6**	**1.6**	1.3
8	1.5	1.4	**1.6**
9	1.2	1.1	1.4
10	1.4	1.1	**1.5**
11	1.3	**1.4**	1.3
12	1.3	1.2	**1.5**
13	**1.6**	**1.6**	1.4
14	1.3	**1.4**	1.4
15	1	**2.2**	1.4
16	1.4	1.2	1.2
17	**1.5**	1.1	1.4
18	1.2	1.3	**1.6**
19	1.3	**1.4**	1.2
20	**1.5**	**1.4**	1

Table 7
The clients' evaluations produced by the proposed CA-FL approach on Synthetic, FEMNIST and CelebA datasets in Non-IID setting. For each dataset the scores of the five top contributing clients are highlighted in red, while the lowest scores are printed in blue..

Client	Synthetic	FEMNIST	CelebA
1	1.6	1	1.4
2	2.2	1.8	**4.2**
3	**5.2**	**3.4**	2.6
4	1.6	1.4	1.4
5	**5.6**	2.2	1.6
6	2.6	2	1.8
7	3.6	**4.8**	**4.8**
8	4.2	2.4	2.4
9	2.2	2.2	1.8
10	**5.2**	2.6	**4.2**
11	2.6	**4.4**	2.4
12	1.6	2.2	1.6
13	2.6	**4.8**	**4.4**
14	1.6	1	**4.4**
15	2.2	2.2	1.4
16	1.6	1.6	2.2
17	1.6	2	1.6
18	**5.4**	1.6	1.8
19	1.2	**3.8**	1
20	**5.6**	2.6	3

Table 8
The clients' evaluations produced by the proposed GP-FL approach on Synthetic, FEMNIST and CelebA datasets in IID setting. For each dataset the scores of the five top contributing clients (the lowest scores) are highlighted in red, while the highest scores are printed in blue. Note that in the context of the GP-FL the low values present better (more stable) clients' behavior while the higher values stand behind unstable client behavior.

Client	Synthetic	FEMNIST	CelebA
1	1.4	1.6	1.4
2	1.2	1.7	1.3
3	1.3	1.3	**1.1**
4	1.4	1.3	**1**
5	1.5	1.6	1.4
6	1.3	**1**	1.7
7	**1.1**	1.3	**1.2**
8	1.8	**1.2**	1.7
9	1.3	**1.1**	1.5
10	**1.1**	1.3	1.3
11	1.5	1.6	1.3
12	1.5	1.5	1.7
13	**1**	**1.1**	1.3
14	1.6	1.4	1.4
15	1.2	1.6	**1.2**
16	**1**	1.8	1.4
17	**1**	1.3	1.5
18	1.6	1.3	1.3
19	1.3	1.5	1.5
20	1.4	**1**	**1**

distributed in cluster 2 (6 and 14), and even two clients (12 and 17) are assigned to cluster 3. In addition, it is interesting to mention that clients 1 and 4, which have the lowest contribution according to FL-Cohort in the three datasets, also received the lowest evaluations from the Deletion approach. We further analyze and compare the performance of our two approaches, CA-FL and GP-FL, to that of the two discussed above baseline approaches. The CA-FL is first evaluated under IID scenario by conducting 5-fold cross-validation for ten global rounds by applying it to the three datasets. The calculated evaluations are presented in Table 6. The highest scores for the three datasets are marked red, while the lowest scores are blue. On the synthetic data set, three clients (5, 7, and 13) represent the top scored ones identified by our CA-FL, and three clients (7, 13, and 15) for FEMNIST data, while four clients (2, 3, 8, 18) for CelebA data set. Since data is equally distributed, the clients have similar model performance and similar evaluation scores as a result, in all datasets used. It is worth mentioning that since here the data is not heterogeneous, the clients' performance is not drastically changing during the training, i.e., due to this their evaluation scores are close. Thus, all clients can be considered reliable under IID scenarios. The latter are rarely the case in real world data scenarios.

In Table 7, the clients' evaluations in Non-IID are calculated by running 5-fold cross-validation on each experimental dataset for ten communication rounds for Non-IID label skew data (30%).

The clients' scores (frequency of being selected as representatives) produced by the CA-FL on the three datasets are given in three different columns of Table 7. The top achieved scores for the three datasets are again marked red, while the lowest scores are blue. We benchmark first the top evaluated clients to the top contributing clients identified by the Deletion approach and FL-Cohort, respectively. In the case of the former approach, three clients (3, 10, and 20) are common with the top-scored clients identified by our CA-FL on the Synthetic dataset, and two for each of the two other datasets, clients 7 and 15 for FEMNIST and clients 10 and 14 for CelebA, respectively. In the case of FL-Cohort, the overlapping between the two lists of the top evaluated clients is even higher, namely three joint clients for the Synthetic and CelebA datasets and two for FEMNIST. Moreover, our CA-FL approach has assigned the lowest scores to the same clients that have received the lowest evaluations from the Deletion approach in the three datasets. Client 2 is only slightly higher evaluated by CA-FL in the Synthetic dataset.

We study the performance of GP-FL by comparing its evaluation results first to that produced by CA-FL and then to the two baseline approaches. Notice that, opposite to the other three approaches, the GP-FL evaluates the clients with negative contributions. Namely, it evaluates each client's behavior by considering how many times the client has changed its cluster during the training process. The calculated

Table 9
The clients' evaluations produced by the proposed GP-FL approach on Synthetic, FEMNIST and CelebA datasets in Non-IID setting. For each dataset the scores of the five top contributing clients (the lowest scores) are highlighted in red, while the highest scores are printed in blue. Note that in the context of the GP-FL the low values present better (more stable) clients' behavior while the higher values stand behind unstable client behavior.

Client	Synthetic	FEMNIST	CelebA
1	4.4	4	2.8
2	3.2	3.5	3.2
3	1.4	1.6	1.8
4	3.6	4	2
5	2.8	2.4	2.4
6	2.4	2.2	4.2
7	1	1	2
8	1.8	1.6	2.2
9	2.4	2	2.4
10	1.2	1.4	1.4
11	2.4	3.8	3
12	3.6	2.8	3.6
13	1.2	1.2	1.8
14	2.4	3.6	3
15	2.2	2	2.2
16	2	2.2	3.8
17	2	2.4	2.4
18	2.6	2.2	2.2
19	4	3.8	4.2
20	1	1.4	2

score is interpreted as a measure of client's instability with respect to its data.

In Table 8 are reported the clients' scores generated by GP-FL on the experimental datasets used and evaluated by running 5-fold cross-validation for ten global rounds under IID data. The highest scores (worst performance) for the three datasets are marked blue, while the lowest scores (most stable behavior) are red. The results are very similar to those listed in Table 6. Namely they have showed that the behavior patterns of the clients are very close, because they are not changing often their clusters. Thus, logically we have more stable behavior of clients in the case of IID data in comparison with that in Non-IID scenarios.

Table 9 shows the clients' scores generated by GP-FL on the three experimental datasets evaluated by running 5-fold cross-validation for 10 communication rounds for Non-IID label skew data (30%). In this table again the highest scores (worst performance) for the three datasets are marked blue, while the lowest scores (most stable behavior) are red. The lists of clients with red-marked scores by our two approaches, CA-FL and GP-FL, are compared. In each dataset, the two approaches have identified three joint clients. The number of overlapping clients between the blue lists of our two approaches is four for the Synthetic dataset and 3 and 2 for FEMNIST and CelebA, respectively. Very similar relations, as discussed above, are observed between the GP-FL and the Deletion approach.

6.3. Analysis of ranking correlations among the approaches

We can apply each of the four studied approaches to each of the three used experimental datasets to generate clients' rankings. These rankings can be used to analyze the correlations between any pair of approaches by applying Kendall's tau method introduced in Section 3.5. Fig. 2 visualizes the heatmaps of Kendall's tau correlation ranking scores for the rankings produced by the four approaches, i.e., CA-FL, GP-FL, Deletion approach, and FL-Cohort, on the three different datasets. The darker is the color, the stronger the correlation of the two methods, and vice versa. These correlations allow us to identify the similarity between the rankings generated by our two FL algorithms and the two baselines.

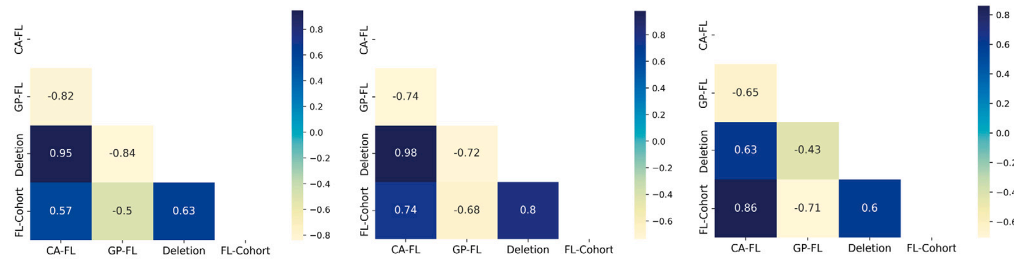
In case of the Synthetic dataset, one can notice that the correlation between CA-FL and the Deletion approach is very strong and positive, namely 0.95. This means that our approach (CA-FL), which requires much less computational resources, can evaluate the clients' contribution similarly to the Deletion approach. The GP-FL produces scores that are interpreted as ones presenting the degree to which clients' behavior is unreliable. This is also the reason for lower correlation scores between the GP-FL and the two baseline approaches (-0.84 and -0.5, respectively), since the latter approaches evaluate how the global model performance will be affected if data of a client or a group of clients is not used in the training. We can generally observe that the FL-Cohort has lower rank correlations with our two approaches than the Deletion approach. Moreover, the correlation between our two approaches is demonstrated to be strong, namely -0.82.

In FEMNIST dataset, as one can see in Fig. 2(b), similar to the Synthetic dataset, CA-FL has a stronger correlation to the Deletion approach than to the FL-Cohort. In contrast, in CelebA dataset, CA-FL has a higher correlation score with the FL-Cohort than the Deletion approach, see Fig. 2(c). In general, in Synthetic and FEMNIST datasets, CA-FL correlations with the Deletion approach are above 0.9, while with FL-Cohort are between 0.5 and 0.7.

To understand the correlations among CA-FL, GP-FL, Deletion approach, and FL-Cohort, we plot together and compare their ranking signatures produced on the three used datasets. These are presented in Fig. 3. We have initially converted the ranking scores of our two approaches into the interval $[0, 1]$ using min–max normalization. In addition, the normalized scores of GP-FL algorithm are further transformed by finding their complement to 1. Analyzing the plots in Fig. 3, one can notice that the lower correlation scores between GP-FL and CA-FL and Deletion approach, respectively, are mostly due to the disagreement on the evaluation of a few clients, namely 15, 16, and 17 for all the three datasets. However, a bigger group of clients (2, 7, 13, 14, 16, 17, and 18) is responsible for the lower correlation score between CA-FL and the Deletion approach in the case of CelebA dataset. In addition, FL-Cohort has demonstrated a different evaluation pattern in comparison to the other three approaches on workers 16, 17, 18, and 19 in all three datasets. Interestingly, clients 16, 17, and 18 are repeated in the commented cases. The first two got low evaluations by our two approaches in all three datasets, while client 18 got a high score by CA-FL and a low one by GP-FL in Synthetic dataset. The same is noticed, e.g., for clients 11 and 19 in FEMNIST dataset. Evidently, the stable performance of some clients recognized by CA-FL is not supported by GP-FL for those clients, i.e., a client can be reliable w.r.t. its local model performance but also demonstrates unstable behavior w.r.t. the class distribution of its data.

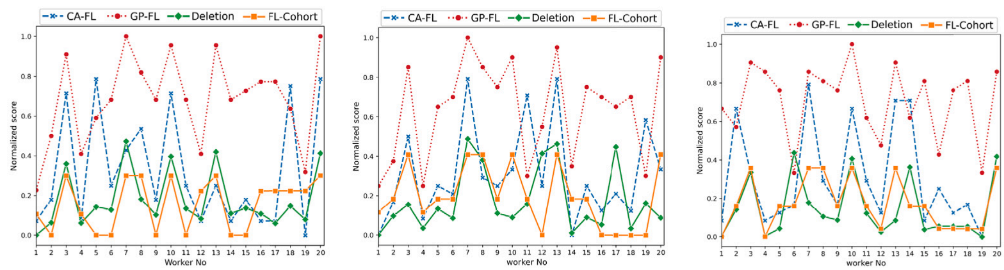
6.4. CA-FL

In this section, we report the results of the additional experiments done to study further how the evaluations of the clients based on the CA-FL algorithm can be used to improve the global model performance. We first analyze how the performance of CA-FL will be affected when different percentages of the clients are selected to participate in federated learning. The clients are initially ranked in descending order based on their scores produced by applying the CA-FL algorithm. Then, we can select different percentages of the clients and train a global model only on those clients. Fig. 4 depicts the performance comparison for CA-FL under different percentages of selected top-scored clients on Synthetic, FEMNIST and CelebA datasets. The performance of CA-FL algorithm has been evaluated by running 3-fold cross-validation on each experimental dataset for 50 communication rounds for Non-IID label skew data (30%). In general, for all three datasets we notice that the algorithm performance is declining proportionally to the increase of the percentage of the selected clients. It can be observed that CA-FL has performed better when the clients with the most reliable behavior (25%) are participating since it is faster to converge. In other words,



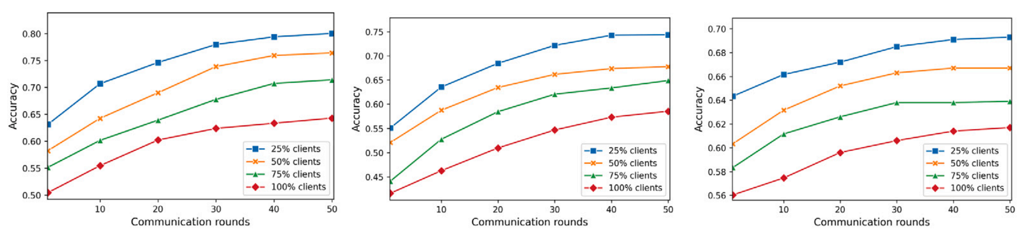
(a) Synthetic (b) FEMNIST (c) CelebA

Fig. 2. Heatmaps of Kendall's tau ranking scores of CA-FL, GP-FL, Deletion approach, and FL-Cohort on Synthetic, FEMNIST and CelebA datasets.



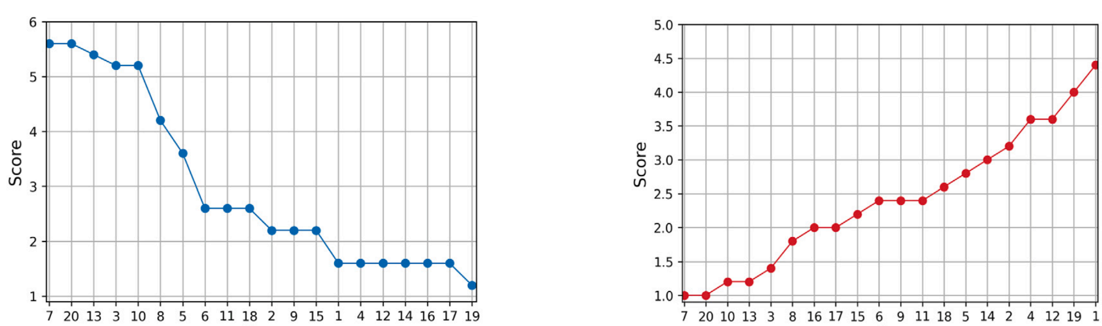
(a) Synthetic (b) FEMNIST (c) CelebA

Fig. 3. Comparison of the ranking signatures of the four approaches produced on Synthetic, FEMNIST and CelebA datasets.



(a) Synthetic (b) FEMNIST (c) CelebA

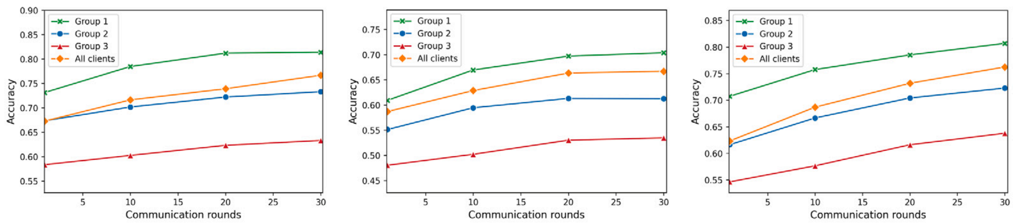
Fig. 4. Performance comparison for CA-FL when different percentages of clients are selected to participate in the training of the shared global model on Synthetic, FEMNIST and CelebA datasets.



(a) Three groups of clients can easily be recognized in CA-FL in descent order: high evaluated (most reliable behavior) clients (scores above 5), middle valued clients (scores above 2 and below 5) and finally, low evaluated (least reliable behavior) clients (scores below 2).

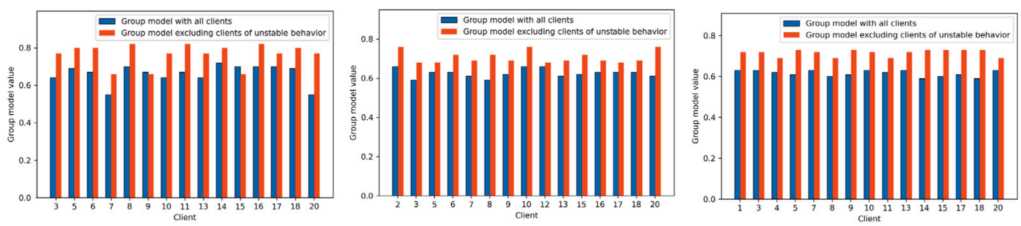
(b) Three groups of clients can be identified in GP-FL in ascent order: low scored (most reliable behavior) clients (scores below 2), middle valued clients (scores above 2 and below 3.5) and finally, high scored (least reliable behavior) clients (scores above 3.5).

Fig. 5. A plot of the clients' scores calculated by CA-FL and GP-FL on the Synthetic dataset.



(a) Synthetic   **(b)** FEMNIST   **(c)** CelebA

Fig. 6. Performance comparison of the three CA-FL global models on Synthetic, FEMNIST and CelebA datasets. The global models are built on three groups of clients split w.r.t. their behavior evaluated during the training process. **Group 1** contains the clients with most reliable behavior, **Group 2** has the clients demonstrating modest behavior w.r.t. the reliability, and **Group 3** are the clients with least reliable behavior.



(a) Synthetic   **(b)** FEMNIST   **(c)** CelebA

Fig. 7. Comparison of the performance of GP-FL models built respectively on all the clients data and when the five clients with most unreliable behavior are excluded on the Synthetic, FEMNIST and CelebA datasets.

as the number of clients with unreliable behavior increases, CA-FL performance degrades.

In addition, we conduct an experiment in which the clients are split in three groups applying binning on their CA-FL evaluations. Fig. 5(a) visualizes the results of the clients' ranking on the Synthetic dataset. As one can notice the clients can be split in three groups as follows: the clients with most reliable behavior (scores above 5), denoted as **Group 1**, the clients demonstrating more modest behavior w.r.t. the reliability (scores above 2 and below 5), **Group 2**, and finally, the clients with least reliable behavior (scores below 2), **Group 3**. For each group of clients a separate federated learning model is trained. The performance of the three built models is benchmarked to that of the global model trained on the all clients' data. This experiment is conducted on each dataset by running a 3-fold cross-validation for 30 global rounds under Non-IID label skew data (30%). Note that the three groups have different size and as well as contain different clients in the three datasets.

Fig. 6 depicts the global model accuracy in different training rounds. The results demonstrate the robustness of CA-FL, since, in the three dataset scenarios, the global model built on clients' data with the most reliable behavior has achieved higher accuracy than the models built on the other two groups of clients. In addition, only this model outperforms the overall global model trained on all clients' data. Evidently, such splitting of clients in groups w.r.t. their behavior and building group global models will allow to ensure fairness in FL scenarios by providing each client a model properly reflecting its respective contribution to the federation.

6.5. GP-FL

In this section, we first analyze how the performance of GP-FL will be affected when the five clients with the most unreliable behavior are excluded from participating in federated learning. This scenario is studied on all three used datasets. The performance of the built GP-FL models is benchmarked to the models built on all the clients' data. See Fig. 7. One can notice that the model trained on the set of clients that do not contain ones with non-stable behavior outperforms the overall global model using the data from all the clients in the three dataset scenarios.

Similarly to the experiment conducted for CA-FL, the clients are split in three groups using binning on their GP-FL evaluations. Fig. 5(b) plots the produced ranking signature on the Synthetic dataset. We have obtained a small group of clients with high scores respectively representing clients with unstable behavior (scores above 3.5), denoted as **Group 3**, and two almost equal size groups of clients demonstrating not so unstable behavior (scores above 2 and below 3.5), **Group 2**, and ones with stable behavior (scores below or equal to 2), **Group 1**, respectively. In addition, in Fig. 8, we compare the performance of the three GP-FL global models in terms of convergence speed on Synthetic, FEMNIST, and CelebA datasets. It is worth noting that with different datasets, again Group 1 always achieves higher accuracy than other groups' models and also outperforms the overall global model trained on the data of all the clients. This may be because the clients in Group 1 have stable behavior, thereby, the results have a prominent gradation. Therefore, only the clients with stable behavior can be selected for aggregating the global model.

7. Conclusion

In this paper, we have proposed a new approach for quantifying the contribution of each client to the overall global model according to its behavior. The proposed approach uses our FL algorithms, CA-FL and GP-FL, respectively. We have studied the robustness of our approach to evaluate the client's behavior during the training process on three LEAF datasets. The obtained experimental results have demonstrated the capability of our approach in realistically assessing the client's contribution to the overall FL model. In summary, without incurring any considerable communication and computation costs, our method has shown robustness in evaluating each client's behavior respectively contribution to the federated model and has the potential to be used for this purpose.

The two FL models can also be extended to more complex cases with a robust unlearning procedure in future work. In addition, we will

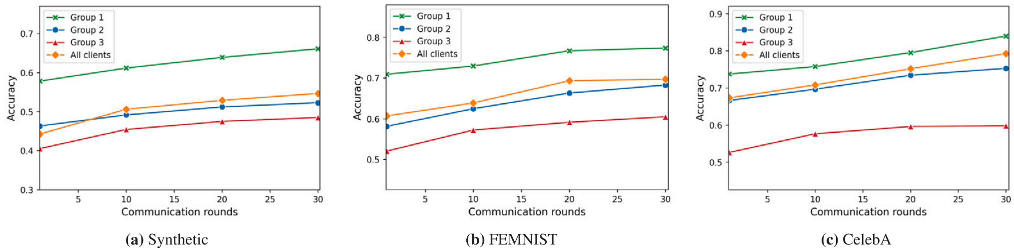


Fig. 8. Performance comparison of the three GP-FL global models on Synthetic, FEMNIST and CelebA datasets. The global models are built on three groups of clients split w.r.t. their behavior evaluated during the training process. **Group 1** contains the clients with stable behavior, **Group 2** has the clients demonstrating not so stable behavior, and **Group 3** are the clients with unstable behavior.

consider developing a hybrid approach that combines CA-FL and GP-FL, which could lead to an advanced FL model that selects the most reliable clients while ignoring the clients with unstable behavior to train a global model.

CRediT authorship contribution statement

Ahmed A. Al-Saedi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Veselka Boeva:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Formal analysis, Data curation, Conceptualization. **Emiliano Casalicchio:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Veselka Boeva's and Emiliano Casalicchio's research is funded partly by the Knowledge Foundation (KKS), Sweden, through the Human-Centered Intelligent Realities (HINTS) Profile Project (contract 20220 068).

Veselka Boeva's research is also partly supported by KKS Foundation through the S.E.R.T. Research Profile Project at Blekinge Institute of Technology. All authors have read and agreed to the published version of the manuscript.

Data availability

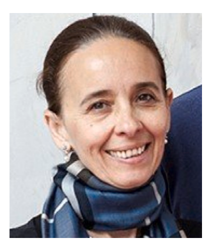
No data was used for the research described in the article.

References

[1] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: International Conference on Artificial Intelligence and Statistics, 2016.
[2] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (2019) 50–60.
[3] Y. Liu, L. Xu, X. Yuan, C. Wang, B. Li, The right to be forgotten in federated learning: An efficient realization with rapid retraining, in: IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, 2022, pp. 1749–1758.
[4] Z. You, X. zhou Wu, K. Chen, X. Liu, C. Wu, Evaluate the contribution of multiple participants in federated learning, in: International Conference on Database and Expert Systems Applications, 2021.
[5] V. Siomos, Contribution evaluation in federated learning: Examining current approaches, 2021.
[6] J. Zhang, C. Li, A. Robles-Kelly, M. Kankanhalli, Hierarchically fair federated learning, 2020, arXiv arXiv:2004.10386.
[7] G. Wang, C.X. Dang, Z. Zhou, Measure contribution of participants in federated learning, in: 2019 IEEE International Conference on Big Data, Big Data, 2019, pp. 2597–2604.
[8] C. Düsing, P. Cimiano, Towards predicting client benefit and contribution in federated learning from data imbalance, in: Proceedings of the 3rd International Workshop on Distributed Machine Learning, 2022.
[9] A.A. Al-Saedi, V. Boeva, E. Casalicchio, Reducing communication overhead of federated learning through clustering analysis, in: 2021 IEEE Symposium on Computers and Communications, ISCC, 2021, pp. 1–7.
[10] A.A. Al-Saedi, V. Boeva, Group-personalized federated learning for human activity recognition through cluster eccentricity analysis, in: International Conference on Engineering Applications of Neural Networks, 2023.
[11] T. Li, M. Sanjabi, V. Smith, Fair resource allocation in federated learning, 2019, arXiv arXiv:1905.10497.
[12] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.
[13] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 3730–3738, http://dx.doi.org/10.1109/ICCV.2015.425.
[14] F. Sattler, S. Wiedemann, K.-R. Müller, W. Samek, Robust and communication-efficient federated learning from non-i.i.d. data, IEEE Trans. Neural Netw. Learn. Syst. 31 (2019) 3400–3413.
[15] C. Briggs, Z. Fan, P. András, Federated learning with hierarchical clustering of local updates to improve training on non-IID data, in: 2020 International Joint Conference on Neural Networks, IJCNN, 2020, pp. 1–9.
[16] Z. Li, Y. He, H. Yu, J. Kang, X. Li, Z. Xu, D.T. Niyato, Data heterogeneity-robust federated learning via group client selection in industrial IoT, IEEE Internet Things J. 9 (2022) 17844–17857.
[17] Y. Xie, Z. Wang, D. Gao, D. Chen, L. Yao, W. Kuang, Y. Li, B. Ding, J. Zhou, FederatedScope: A flexible federated learning platform for heterogeneity, Proc. VLDB Endow. 16 (2022) 1059–1072.
[18] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: User-level privacy leakage from federated learning, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2018, pp. 2512–2520.
[19] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D.S. Doermann, A. Innanje, Ensemble attention distillation for privacy-preserving federated learning, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 15056–15066.
[20] J. Huang, C. Xu, Z. Ji, S. Xiao, T. Liu, N. Ma, Q. Zhou, AFLPC: An asynchronous federated learning privacy-preserving computing model applied to 5G-V2X, Secur. Commun. Netw. (2022).
[21] L. Wang, W. Wang, B. Li, CMFL: Mitigating communication overhead for federated learning, in: 2019 IEEE 39th International Conference on Distributed Computing Systems, ICDCS, 2019, pp. 954–964.
[22] M. Asad, A. Moustafa, T. Ito, M. Aslam, Evaluating the communication efficiency in federated learning algorithms, in: 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design, CSCWD, 2020, pp. 552–557.
[23] N. Bouacida, J. Hou, H. Zang, X. Liu, Adaptive federated dropout: Improving communication efficiency and generalization for federated learning, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, 2020, pp. 1–6.
[24] B. Huang, X. Li, Z. Song, X. Yang, FL-NTK: A neural tangent kernel-based framework for federated learning analysis, in: International Conference on Machine Learning, 2021.
[25] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, X. Wang, Resource-efficient and convergence-preserving online participant selection in federated learning, in: 2020 IEEE 40th International Conference on Distributed Computing Systems, ICDCS, 2020, pp. 606–616.
[26] Y.J. Cho, J. Wang, G. Joshi, Client selection in federated learning: Convergence analysis and power-of-choice selection strategies, 2020, arXiv arXiv:2010.01243.

[27] L. Liu, J. Zhang, S. Song, K.B. Letaief, Hierarchical quantized federated learning: Convergence analysis and system design, 2021, arXiv arXiv:2103.14272.
[28] S. Caldas, P. Wu, T. Li, J. Konecný, H.B. McMahan, V. Smith, A. Talwalkar, LEAF: A benchmark for federated settings, 2018, arXiv arXiv:1812.01097.
[29] F. Sattler, K.-R. Müller, W. Samek, Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints, IEEE Trans. Neural Netw. Learn. Syst. 32 (2019) 3710–3722.
[30] M. Duan, D. Liu, X. Ji, R. Liu, L. Liang, X. Chen, Y. Tan, FedGroup: Ternary cosine similarity-based clustered federated learning framework toward high accuracy in heterogeneous data, 2020, arXiv arXiv:2010.06870.
[31] A. Ghosh, J. Chung, D. Yin, K. Ramchandran, An efficient framework for clustered federated learning, IEEE Trans. Inform. Theory 68 (2020) 8076–8091.
[32] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, J. Jiang, Multi-center federated learning, 2020, arXiv arXiv:2108.08647.
[33] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, R. Vidal, Federated multi-task learning under a mixture of distributions, in: Neural Information Processing Systems, 2021.
[34] Y. Ruan, C. Joe-Wong, FedSoft: Soft clustered federated learning with proximal local updating, in: AAAI Conference on Artificial Intelligence, 2021.
[35] Y. Mansour, M. Mohri, J. Ro, A.T. Suresh, Three approaches for personalization with applications to federated learning, 2020, arXiv arXiv:2002.10619.
[36] Fairness and privacy preserving in federated learning: A survey, Inf. Fusion 105 (2024) 102198, http://dx.doi.org/10.1016/j.inffus.2023.102198.
[37] S.K. Shyn, D. Kim, K. Kim, Fedccea : A practical approach of client contribution evaluation for federated learning, 2021, arXiv arXiv:2106.02310.
[38] X. Deng, J. Li, C. Ma, K. Wei, L. Shi, M. Ding, W. Chen, Low-latency federated learning with DNN partition in distributed industrial IoT networks, IEEE J. Sel. Areas Commun. 41 (2022) 755–775.
[39] T. Wang, J. Rausch, C. Zhang, R. Jia, D.X. Song, A principled approach to data valuation for federated learning, in: Federated Learning, 2020.
[40] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D.T. Niyato, Q. Yang, A sustainable incentive scheme for federated learning, IEEE Intell. Syst. 35 (2020) 58–69.
[41] B. Zhao, X. Liu, W. neng Chen, When crowdsensing meets federated learning: Privacy-preserving mobile crowdsensing system, 2021, arXiv arXiv:2102.10109.
[42] R. Zeng, S. Zhang, J. Wang, X. Chu, FMore: An incentive scheme of multi-dimensional auction for federated learning in MEC, in: 2020 IEEE 40th International Conference on Distributed Computing Systems, ICDCS, 2020, pp. 278–288.
[43] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, K.S. Ng, Towards fair and privacy-preserving federated deep models, IEEE Trans. Parallel Distrib. Syst. 31 (2019) 2524–2541.
[44] T. Song, Y. Tong, S. Wei, Profit allocation for federated learning, in: 2019 IEEE International Conference on Big Data, Big Data, 2019, pp. 2577–2586.
[45] R. Jia, D. Dao, B. Wang, F.A. Hubis, N. Hynes, N.M. Gürel, B. Li, C. Zhang, D.X. Song, C.J. Spanos, Towards efficient data valuation based on the Shapley value, in: International Conference on Artificial Intelligence and Statistics, 2019.
[46] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: L.M.L. Cam, J. Neyman (Eds.), Proceedings of the Berkley symposium on mathematical statistics and probability, Vol. 1, 1967, pp. 281–297.
[47] X. Jin, J. Han, K-medoids clustering, in: Encyclopedia of Machine Learning, 2010.
[48] M.J. van der Laan, K.S. Pollard, J. Bryan, A new partitioning around medoids algorithm, J. Stat. Comput. Simul. 73 (2003) 575–584.
[49] S. van Dongen, Graph Clustering by Flow Simulation (Ph.D. thesis), 2000.
[50] J. Vlasblom, S.J. Wodak, Markov clustering versus affinity propagation for the partitioning of protein interaction graphs, BMC Bioinformatics 10 (2009) 99.
[51] P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[52] C.G. Bezerra, et al., An evolving approach to data streams clustering based on typicality and eccentricity data analytics, Inform. Sci. 518 (2020) 13–28, http://dx.doi.org/10.1016/j.ins.2019.12.022.
[53] P. Angelov, Anomaly detection based on eccentricity analysis, in: 2014 IEEE Symposium on Evolving and Autonomous Learning Systems, EALS, 2014, pp. 1–8, http://dx.doi.org/10.1109/EALS.2014.7009497.
[54] J.G. Saw, et al., Chebyshev inequality with estimated mean and variance, Amer. Statist. 38 (1984) 130–132.
[55] I. Škrjanc, et al., Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey, Inform. Sci. 490 (2019) 344–368.
[56] M. Kendall, Rank correlation methods, 1949.
[57] R.N. Forthofer, R.G. Lehnen, Rank correlation methods, in: Public Program Analysis: A New Categorical Data Approach, Springer US, Boston, MA, 1981, pp. 146–163, http://dx.doi.org/10.1007/978-1-4684-6683-6_9.
[58] Y. Liu, S. Qin, Y. Sun, G. Feng, Resource consumption for supporting federated learning in wireless networks, IEEE Trans. Wireless Commun. 21 (2022) 9974–9989.
[59] G. Drainakis, P. Pantazopoulos, K.V. Katsaros, V. Sourlas, A.J. Amditis, D.I. Kaklamani, From centralized to federated learning: Exploring performance and end-to-end resource consumption, Comput. Netw. 225 (2023) 109657.
[60] J. Huang, R. Talbi, Z. Zhao, S. Bouchenak, L.Y. Chen, S. Roos, An exploratory analysis on users' contributions in federated learning, in: 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA, 2020, pp. 20–29.
[61] G. Navas-Palencia, Optimal binning: mathematical programming formulation, 2020, arXiv arXiv:2001.08025.
[62] P. Flach, Machine learning - The art and science of algorithms that make sense of data, 2012.
[63] S. Kolouri, et al., Optimal mass transport: Signal processing and machine-learning applications, IEEE Signal Process. Mag. 34 (2017) 43–59.
[64] V.N.L. Duy, I. Takeuchi, Exact statistical inference for the Wasserstein distance by selective inference, Ann. Inst. Stat. Math. 75 (2021) 127–157.
[65] T. Nishio, R. Shinkuma, N.B. Mandayam, Estimation of individual device contributions for incentivizing federated learning, in: 2020 IEEE Globecom Workshops, GC Wkshps, 2020, pp. 1–6.

Ahmed A. Al-Saedi holds a bachelor's degree in computer science from Al-Mustansiriyah University, Iraq, and a master's degree in computer science in information and communication systems from TU Kaiserslautern (TUK), Germany. His current research interests include federated learning, swarm learning, and Edge AI.



Veselka Boeva is Professor of Computer Science at Blekinge Institute of Technology, Sweden. She received an M.Sc. degree in Mathematics with specialization in Computer Science at the Plovdiv University, Bulgaria. She was awarded a Ph.D. degree in Computer Science from the Technical University of Sofia branch Plovdiv, Bulgaria in 2000.

Emiliano Casalicchio research interests are in large-scale distributed systems, focusing on performance, energy consumption, and security.