DPP-BASED CLIENT SELECTION FOR FEDERATED LEARNING WITH NON-IID DATA

Yuxuan Zhang[†], *Chao Xu*[†§], *Howard H. Yang*[‡], *Xijun Wang*[*], *and Tony Q. S. Quek*[◇]

[†]School of Information Engineering, Northwest A&F University, Yangling, Shaanxi, China
[‡]ZJU-UIUC Institute, Zhejiang University, Haining, China
[*]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
[◇]ISTD Pillar, Singapore University of Technology and Design, Singapore

ABSTRACT

This paper proposes a client selection (CS) method to tackle the communication bottleneck of federated learning (FL) while concurrently coping with FL's data heterogeneity issue. Specifically, we first analyze the effect of CS in FL and show that FL training can be accelerated by adequately choosing participants to diversify the training dataset in each round of training. Based on this, we leverage data profiling and determinantal point process (DPP) sampling techniques to develop an algorithm termed Federated Learning with DPP-based Participant Selection (FL-DP^{3}S). This algorithm effectively diversifies the participants' datasets in each round of training while preserving their data privacy. We conduct extensive experiments to examine the efficacy of our proposed method. The results show that our scheme attains a faster convergence rate, as well as a smaller communication overhead than several baselines.

***Index Terms*—** Client selection, determinantal point process, federated learning, data heterogeneity.

1. INTRODUCTION

With the rapid development of the Internet of Things (IoT) and social networking applications, there is an exponential growth of the data generated by intelligent devices, such as smartphones and laptops [1]. The sheer volume of these data and the privacy concerns prevent aggregating the raw data to a centralized data center, which further motivates an emerging distributed collaborative artificial intelligence (AI) paradigm called federated learning (FL) [2–4]. Specifically, FL enables clients to perform local model training utilizing their individual data and upload the intermediate parameters to the central server for global aggregation, after which an improved model is sent back to the clients for another round of local training [4–6]. In practice, there are usually a massive number of clients connected to the server via a resource-limited medium, e.g., the spectrum. Hence, only a limited number of clients can be selected to participate in FL during each round of training [7]. In response, the vanilla FL algorithm, called FedAvg [4], has been proposed, with which the server selects a subset of clients uniformly at random during each round of communication. While FedAvg has demonstrated its success in some applications, e.g., large-scale systems [8], recent studies [9, 10] revealed that a deterioration in the accuracy and convergence of FedAvg and its variants is almost inevitable facing the clients with non-independent and identically distributed (non-IID) data, which is a common scenario in practice. Essentially, this deterioration is mainly attributed to the weight divergence of local models trained by the clients [9, 10].

[§]Corresponding author: Chao Xu, cxu@nwafu.edu.cn.

To improve the performance of FL on non-IID data, various FL algorithms have been proposed in a line of recent work [10–17], which can be broadly divided into two categories. Particularly, the first group of work aims to reduce the weight divergence of local models by modifying the data distributions at clients via data sharing [10, 11] or data augmentation [12, 13]. However, it requires the clients to share their private datasets, thereby increasing the risk of privacy leakage and incurring extra communication costs. To this end, instead of changing individual clients' local datasets, another line of work [14–17] focuses on improving the training performance by devising efficient client selection (CS) strategies. Although the gain of CS schemes has been well demonstrated via experiments in [14–17], the role of CS on improving the performance of FL is not theoretically well-understood. Besides, to improve the effectiveness of CS in FL, the server needs to obtain a certain amount of knowledge of the clients' local data distributions. This is usually achieved by directly collecting the distributions of all clients' local datasets [14, 15], or periodically querying the gradients of all clients [16], or scratching the connection between the local data distribution and local model parameters via learning-based algorithms [17], but that increases the risk of privacy leakage or the consumption of computational and communication resources.

To fill this research gap, the present paper theoretically analyzes the role of CS in FL by resorting to the conclusion regarding the effect of mini-batch sampling in mini-batch stochastic gradient descent (SGD). Then, we propose a novel CS algorithm, Federated Learning with DPP-based Participant Selection (FL-DP^{3}S), by jointly leveraging the data profiling and k-determinantal point process (k-DPP) sampling techniques. FL-DP^{3}S adequately chooses the participants to diversify the training dataset in each training round while reducing the risk of privacy leakage and communication overhead. The effectiveness of FL-DP^{3}S is verified via extensive experiments on two public image datasets.

2. SYSTEM MODEL AND PROBLEM FORMULATION

2.1. Setting

We consider an FL system with one central server organizing C clients to collaboratively train a global model[1] parameterized by $\mathbf{w}_g$. The set of clients is denoted by $\mathcal{C} = \{1, 2, \ldots, C\}$. Each client $c \in \mathcal{C}$ possesses a local dataset $\mathbf{D}_c = \{(\mathbf{x}_c^i, y_c^i)\}_{i=1}^{n_c}$, where $(\mathbf{x}_c^i, y_c^i)$ is the i-th sample (i.e., feature-label pair) and $n_c = |\mathbf{D}_c|$ denotes the size of dataset $\mathbf{D}_c$. The goal of this FL system is to minimize the

[1]In this paper, the term model refers to the convolutional neural network (CNN), and the terms of model and its parameters are interchangeably used.

following global objective function

$$f(\mathbf{w}) = \sum_{c\in\mathcal{C}} \frac{n_c}{\sum_{c\in\mathcal{C}} n_c}\mathcal{L}_c(\mathbf{w}) = \frac{1}{\sum_{c\in\mathcal{C}} n_c}\sum_{c\in\mathcal{C}}\sum_{i=1}^{n_c}\ell((\mathbf{x}_c^i, y_c^i); \mathbf{w}) \quad (1)$$

where $\mathcal{L}_c(\mathbf{w}) = \sum_{i=1}^{n_c} \ell((\mathbf{x}_c^i, y_c^i); \mathbf{w})/n_c$ is the local empirical loss constructed from client c's dataset and $\ell((\mathbf{x}_c^i, y_c^i); \mathbf{w})$ denotes the loss function evaluated at an individual sample $(\mathbf{x}_c^i, y_c^i)$. As such, the optimal parameters of the global model $\mathbf{w}_g^*$ can be expressed as

$$\mathbf{w}_g^* = \arg\min_{\mathbf{w}} f(\mathbf{w}). \quad (2)$$

In each training round $t \in \{1, \ldots, T\}$ of FedAvg, the server randomly selects C_p clients, denoted by $\mathcal{C}_t$ (i.e., $|\mathcal{C}_t| = C_p$), and then sends them the current global model $\mathbf{w}_g^{(t-1)}$. After receiving $\mathbf{w}_g^{(t-1)}$, each client $c \in \mathcal{C}_t$ updates its own local model $\mathbf{w}_c^{(t)}$ by making E training passes over its local dataset, i.e.,

$$\mathbf{w}_c^{(t)} = \mathbf{w}_g^{(t-1)} - \sum_{e=1}^{E}\frac{\eta}{n_c}\sum_{i=1}^{n_c}\nabla_{\mathbf{w}_{L,e}^{(t)}}\ell((\mathbf{x}_c^i, y_c^i); \mathbf{w}_{L,e}^{(t)}) \quad (3)$$

with

$$\mathbf{w}_{L,e}^{(t)} = \begin{cases} \mathbf{w}_g^{(t-1)} & e = 0 \\ \mathbf{w}_{L,e-1}^{(t)} - \frac{\eta}{n_c}\sum_{i=1}^{n_c}\nabla_{\mathbf{w}_{L,e-1}^{(t)}}\ell((\mathbf{x}_c^i, y_c^i); \mathbf{w}_{L,e-1}^{(t)}) & e \neq 0 \end{cases} \quad (4)$$

where η denotes the learning rate, and $\nabla_{\mathbf{w}_{L,e}^{(t)}}\ell((\mathbf{x}_c^i, y_c^i); \mathbf{w}_{L,e}^{(t)})$ the gradient of $\ell((\mathbf{x}_c^i, y_c^i); \mathbf{w}_{L,e}^{(t)})$ on model $\mathbf{w}_{L,e}^{(t)}$. By substituting (4) into (3), we have

$$\mathbf{w}_c^{(t)} = \mathbf{w}_g^{(t-1)} - \frac{\eta}{n_c}\sum_{i=1}^{n_c} F((\mathbf{x}_c^i, y_c^i); \mathbf{w}_g^{(t-1)}; E) \quad (5)$$

in which $F((\mathbf{x}_c^i, y_c^i); \mathbf{w}_g^{(t-1)}; E)$ represents the equivalent contribution of sample $(\mathbf{x}_c^i, y_c^i)$ to the local update. After receiving all participants' uploaded local models, the server updates the global model by aggregating them as

$$\mathbf{w}_g^{(t)} = \sum_{c\in\mathcal{C}_t}\frac{n_c}{\sum_{c\in\mathcal{C}_t} n_c}\mathbf{w}_c^{(t)}. \quad (6)$$

Then, the server selects a set of clients $\mathcal{C}_{t+1}$ again and starts a new training round. This workflow repeats until the training converges.

2.2. Challenge of Data Heterogeneity

Owing to the difference in user preferences, the data samples generated by clients can be highly non-IID, deteriorating the performance of FedAvg. For instance, as demonstrated in [10], the predicting accuracy of a statistical model trained under FedAvg can reduce by 55% compared to the case with IID data. Several previous studies [14–17] have demonstrated via experiments that it is crucial to develop efficient CS strategies for improving the performance of FL under non-IID data. To further investigate the mechanism behind this improvement, as well as understanding the role of CS in each round of training, we resort to the conclusions regarding the effect of mini-batch sampling in the SGD update.

Particularly, by substituting (5) into (6), the FedAvg update in the t-th training round can be rewritten as

$$\mathbf{w}_g^{(t)} = \sum_{c\in\mathcal{C}_t}\frac{n_c}{\sum_{c\in\mathcal{C}_t} n_c}\Big(\mathbf{w}_g^{(t-1)} - \frac{\eta}{n_c}\sum_{i=1}^{n_c} F\big((\mathbf{x}_c^i, y_c^i); \mathbf{w}_g^{(t-1)}; E\big)\Big) = \mathbf{w}_g^{(t-1)} - \frac{\eta}{|\mathbf{D}_{\mathcal{C}_t}|}\sum_{(\mathbf{x}_c^i, y_c^i)\in\mathbf{D}_{\mathcal{C}_t}} F\big((\mathbf{x}_c^i, y_c^i); \mathbf{w}_g^{(t-1)}; E\big) \quad (7)$$

where $\mathbf{D}_{\mathcal{C}_t}$ denotes the union of participants' datasets. On the other hand, for a generic SGD-based training algorithm, the mini-batch update in the t-th training round can be expressed as [18, 19]

$$\mathbf{w}_s^{(t)} = \mathbf{w}_s^{(t-1)} - \frac{\eta}{|\mathcal{B}_t|}\sum_{(\mathbf{x}^i, y^i)\in\mathcal{B}_t}\nabla_{\mathbf{w}_s^{(t-1)}}\ell((\mathbf{x}^i, y^i); \mathbf{w}_s^{(t-1)}) \quad (8)$$

where $\mathcal{B}_t$ is a randomly sampled mini-batch.

By comparing (7) and (8), we note that for both FedAvg and mini-batch SGD, the model update in each round of training is determined by the involved dataset (i.e., $\mathbf{D}_{\mathcal{C}_t}$ in FL and $\mathcal{B}_t$ in SGD, respectively). This phenomenon unveils that the CS plays a role in FL similar to that of mini-batch sampling in SGD. More importantly, if the number of local iterations E is set to 1, FedAvg reduces to the Federated SGD (FedSGD) algorithm [4], where $F((\mathbf{x}_c^i, y_c^i); \mathbf{w}_g^{(t-1)}; E)$ aligns with the gradient $\nabla_{\mathbf{w}_g^{(t-1)}}\ell((\mathbf{x}_c^i, y_c^i); \mathbf{w}_g^{(t-1)})$ and (7) degenerates as follows

$$\mathbf{w}_g^{(t)} = \mathbf{w}_g^{(t-1)} - \frac{\eta}{|\mathbf{D}_{\mathcal{C}_t}|}\sum_{(\mathbf{x}_c^i, y_c^i)\in\mathbf{D}_{\mathcal{C}_t}}\nabla_{\mathbf{w}_g^{(t-1)}}\ell((\mathbf{x}_c^i, y_c^i); \mathbf{w}_g^{(t-1)}) \quad (9)$$

which essentially is the same as the mini-batch update in (8). This observation motivates us to further investigate the effect of CS in FL by resorting to that of mini-batch sampling in SGD.

Specifically, in the context of SGD, the stochastic gradient computed from $\mathcal{B}_t$ is an approximation of the true gradient calculated using the entire dataset. And, in general, the larger the variance of the gradient approximation, the slower the model training convergence [20, 21]. One approach to reduce the variance is to sample data from different regions of the feature space, termed mini-batch diversification [21, 22], since the data samples from similar regions of the feature space commonly contribute similar gradients to the SGD update. Consequently, the more diverse the data samples, the better the gradient approximation.

Following a similar vein to the above argument, we conjecture that for FL training, the convergence can be accelerated by adequately selecting the clients to diversify the training dataset in each round of training. If the data distributions of clients are accessible by the central server, such a scheduling policy can be readily devised (see [14, 15] for instance). However, such distributions are usually not available in practice due to privacy concerns. In light of this, we aim to design a novel CS algorithm for FL training with non-IID data, termed FL-DP^{3}S. With this algorithm, in each training round, participants' datasets can be diversified to accelerate the training convergence. At the same time, the risk of privacy leakage and communication overhead is effectively reduced.

3. ALGORITHM DESIGN

This section develops a novel CS algorithm by jointly leveraging the data profiling and k-DPP sampling techniques. During the initialization stage, each client profiles its local dataset utilizing the mean

Algorithm 1 Federated Learning with DPP-based Participant Selection (FL-DP^{3}S)

Initialization: Initialize the global model parameters $\mathbf{w}_g^{(0)}$.
for each client $c \in \mathcal{C}$ in parallel **do**
    Profile its local dataset with (11) and upload it to the server.
end for
The server calculates similarity matrix $\mathbf{S}$ according to (14) and constructs a k-DPP.
for $t = 1, 2, \ldots, T$ **do**
    The server selects a set $\mathcal{C}_t$ of C_p clients by resorting to the constructed k-DPP.
    for each client $c \in \mathcal{C}_t$ in parallel **do**
        Update $\mathbf{w}_c^{(t)}$ with (5) and then upload it to the server.
    end for
    The server updates global model $\mathbf{w}_g^{(t)}$ according to (6).
end for
Output: Output the well-trained global model $\mathbf{w}_g^{(T)}$.

vector of the outputs of the first fully-connected layer (FC-1) in the global model. Then, with the data profiles of clients, a DPP-based efficient CS strategy is established.

3.1. Data Profiling of Clients

Motivated by [23], we enable each client to profile its local dataset with the mean vector of the FC-1 outputs in the global model according to Theorem 1 under Assumption 1.

Assumption 1 *Let $\mathcal{W} \in \mathbb{R}^{Q\times V}$ denote the weights of the FC-1 of a CNN model consisting of Q neurons, with $\boldsymbol{\omega}_q = [\omega_{q,1}, \omega_{q,2}, \ldots, \omega_{q,V}]$ and b_q respectively representing the weights and bias regarding the q-th neuron. Besides, let $\mathbf{o} \in \mathbb{R}^V$ denote the input features of the model's FC-1 with V dimensions, o_v the v-th feature in $\mathbf{o}$, and $z_{q,v} = o_v\omega_{q,v}$ the v-th weighted input of the q-th neuron. Then, the following conditions are satisfied: (1) The feature o_v follows some distribution $\mathcal{F}_v(\mu_v, \sigma_v^2)$ with finite mean μ_v and variance σ_v^2; (2) There exists a constant $\delta > 0$ for each neuron q in FC-1 such that:*

$$\lim_{V\to\infty} \frac{1}{s_q^{2+\delta}} \sum_{v=1}^{V} \mathbb{E}\Big[\big|z_{q,v} - \omega_{q,v}\mu_v\big|^{2+\delta}\Big] = 0 \tag{10}$$

where $s_q = \sqrt{\sum_{v=1}^{V}(\omega_{q,v}\sigma_v)^2}$.

Theorem 1 *Given a model's FC-1 and a set of input features satisfying Assumption 1, the distribution of the outputs of the q-th neuron in the FC-1, during forward propagation, tends to follow a Gaussian distribution, whose mean and variance are $u_q = \sum_{v=1}^{V}\omega_{q,v}\mu_v + b_q$ and $s_q^2 = \sum_{v=1}^{V}(\omega_{q,v}\sigma_v)^2$, respectively.*

Proof. See [23] for a detailed proof.

Remark 1 *Assumption 1 can be satisfied if the model is properly initialized and the input data are normalized as discussed in [23]. In practice, these techniques are widely used in deep learning model training [24, 25].*

According to Theorem 1, for each client c with local dataset $\mathbf{D}_c$, when given a CNN model, the outputs of the q-th neuron of the model's FC-1 tend to follow a Gaussian distribution $h_q \sim \mathcal{N}(u_q^c, (s_q^c)^2)$, where the mean u_q^c and standard deviation s_q^c are determined by the input features to the model's FC-1. It is noteworthy that for a CNN model, the input features of FC-1 are extracted by the previous convolution layers, which can be seen as the latent representations of the training data samples [23, 26]. On this basis, it is reasonable to profile each client's local dataset by using the mean vector of the FC-1 outputs, i.e.,

$$\mathbf{f}_c = [u_1^c, u_2^c, \ldots, u_Q^c], \forall c \in \mathcal{C}. \tag{11}$$

For each client c, $\mathbf{f}_c$ is called her data profile, which has a size of BQ bits if a float number is B bits long. It should be noted that the data size of each client's profile is extremely small and only needs to be updated to the central server once during the initialization stage, which consumes very little communication resources. Besides, in contrast to directly collecting the distributions of all clients' datasets, this method significantly reduces the risk of privacy leakage.[2]

3.2. DPP-Based Client Selection

With the clients' data profiles, a DPP-based CS strategy can be further devised to avoid selecting similar clients in each round of training. Note that DPP is a probabilistic model of repulsion, which has been widely adopted for solving subset sampling problems with diversity constraints in machine learning [27]. And the k-DPP is a variant of DPP, with which the size of sampled subsets is fixed at k [28].

Particularly, a DPP is a probabilistic model over subsets on a finite set, which can be derived from a positive semi-definite similarity kernel matrix [27]. For a finite set $\mathcal{M}$ with M elements, the similarity kernel matrix $\mathbf{L}$ can be expressed as $\mathbf{L} = \{l_{m,n}\}_{M\times M}$, with $l_{m,n}$ representing the similarity between the m-th and n-th elements in $\mathcal{M}$. Meanwhile, the DPP assigns a probability to sub-sampling any subset $\mathcal{Y}$ of $\mathcal{M}$, which is proportional to the determinant of the sub-matrix $\mathbf{L}_\mathcal{Y}$ regarding the subset $\mathcal{Y}$, i.e.,

$$\Pr(\mathcal{Y}) = \frac{\det(\mathbf{L}_\mathcal{Y})}{\det(\mathbf{L}+\mathbf{I})} \propto \det(\mathbf{L}_\mathcal{Y}) \tag{12}$$

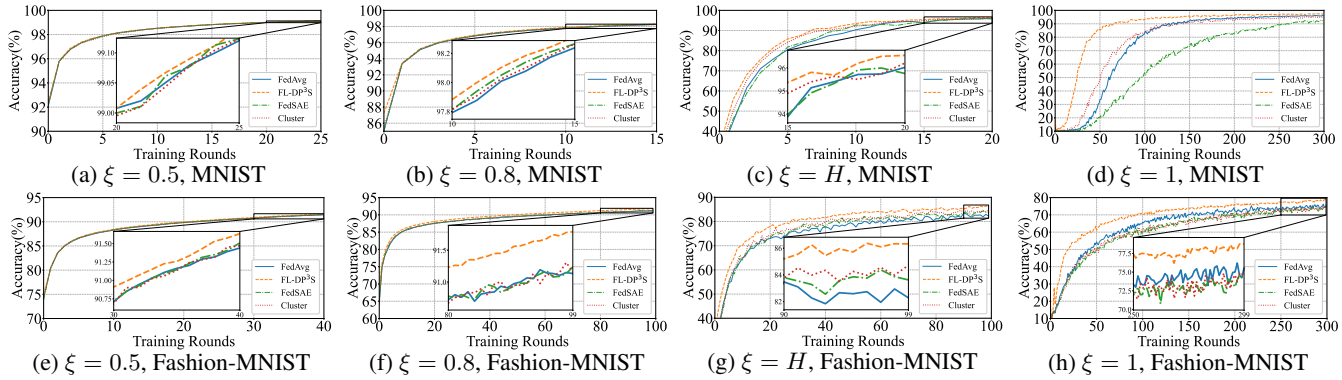
where $\mathbf{I}$ denotes the $M \times M$ identity matrix. For instances, if $\mathcal{Y} = \{m, n\} \subset \mathcal{M}$, then we have $\Pr(\mathcal{Y}) \propto l_{m,m}l_{n,n} - l_{m,n}l_{n,m}$. By nature, the value of $\Pr(\mathcal{Y})$ decreases as the similarity of elements in set $\mathcal{Y}$ increases. In other words, the more diversified the elements in $\mathcal{Y}$ are, the higher the likelihood that the set $\mathcal{Y}$ is sampled.

Furthermore, to sample sets with a fixed cardinality k, one can use k-DPP [28] which assigns probability to each subset $\mathcal{Y}$ (i.e., $\mathcal{Y} \subset \mathcal{M}$, $|\mathcal{Y}| = k$) as

$$\Pr^k(\mathcal{Y}) = \frac{\det(\mathbf{L}_\mathcal{Y})}{\sum_{|\mathcal{Y}'|=k}\det(\mathbf{L}_{\mathcal{Y}'})}. \tag{13}$$

In light of this, for the CS problem considered in this paper, the similarity kernel matrix $\mathbf{L}$ can be constructed by using the data profiles of all clients, i.e., $\mathcal{M} = \mathcal{C}$ and $\mathbf{L} = \{l_{m,n}\}_{C\times C}$, where each element $l_{m,n}$ is an appropriate measure of the similarity between the m-th and n-th clients' data profiles. Then, by setting the cardinality of sampled subsets as C_p, we can achieve the diversified CS in each round of training with the aid of the k-DPP. As an instance, we construct the similarity kernel matrix as $\mathbf{L} = \mathbf{S}^{\mathrm{T}}\mathbf{S}$ with

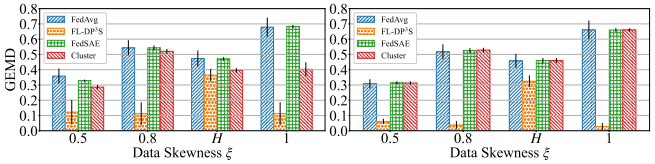
[2] We would like to note that (at least to the best of our knowledge) developing quantitative measures for privacy in FL is still an open problem [1], and it is out of the scope of this paper.



(a) $\xi = 0.5$, MNIST (b) $\xi = 0.8$, MNIST (c) $\xi = H$, MNIST (d) $\xi = 1$, MNIST

(e) $\xi = 0.5$, Fashion-MNIST (f) $\xi = 0.8$, Fashion-MNIST (g) $\xi = H$, Fashion-MNIST (h) $\xi = 1$, Fashion-MNIST

Fig. 1: Accuracy *v.s.* training rounds on MNIST and Fashion-MNIST datasets with different levels of heterogeneity.



(a) MNIST (b) Fashion-MNIST

Fig. 2: GEMD comparison on MNIST and Fashion-MNIST datasets with different levels of heterogeneity.

$\mathbf{S} = \{s_{m,n}\}_{C\times C}$ denoting the similarity matrix, where each element $s_{m,n}$ is defined as

$$s_{m,n} = 1 - \left(\frac{s^0_{m,n} - \min(\mathbf{S}^0)}{\max(\mathbf{S}^0) - \min(\mathbf{S}^0)} \right). \tag{14}$$

In (14), $s^0_{m,n} = \|\mathbf{f}_m - \mathbf{f}_n\|_2$ with $\mathbf{f}_m, \forall m \in \mathcal{C}$ representing the client's data profile, and $\mathbf{S}^0$ is defined as $\{s^0_{m,n}\}_{C\times C}$, whose maximum and minimum elements are denoted by $\max(\mathbf{S}^0)$ and $\min(\mathbf{S}^0)$, respectively.

3.3. Algorithm Workflow

We summarize the pseudocode of FL-DP^{3}S in Algorithm 1. First, the server initializes the global model. Then, the server obtains each client's data profile defined in (11), and calculates the similarity kernel matrix according to (14), with which a k-DPP can be further constructed. After the initialization, FL-DP^{3}S goes into a loop. In each round of training, the server selects the clients by resorting to the constructed k-DPP. This loop will terminate when the preset maximum iteration number T is reached.

4. EXPERIMENT

We evaluated the performance of FL-DP^{3}S by training the CNN model with two convolutional layers and two fully-connected layers on two public image datasets, i.e., MNIST [26] and Fashion-MNIST [29], each of which consists of $60,000$ data samples. Here, we set $C = 100$ and $C_p = 10$. For comparison, three stat-of-the-art FL algorithms (i.e., FedSAE [30], Cluster (i.e., Algorithm 2 in [31]) and FedAvg [4]) are used as benchmarks. Particularly, in each round of training, FedSAE prefers to select clients with a higher local loss, while Cluster tries to diversify the selected clients by considering the similarity among clients' representation gradients.

Following [17], we consider that the clients' local datasets are of a uniform size, and use data skewness ξ to represent the level of heterogeneity in the data distribution. Particularly, for the data samples possessed by one client, $\xi = 1$ indicates that they only belong to one class, $\xi = 0.8$ indicates that 80% of them belong to one class and the remaining 20% samples belong to other classes, $\xi = 0.5$ indicates that 50% of them belong to one class and the remaining 50% samples belong to other classes, and $\xi = H$ indicates that they evenly belong to two different classes. Here, we repeat each experiment 50 times (with different random seeds), and present the average accuracy of the global model on the training set in Fig. 1. As demonstrated in Fig. 1, our proposed FL-DP^{3}S algorithm outperforms the benchmarks in all cases and the superiority becomes more significant as the data heterogeneity level increases, i.e., when ξ changes from 0.5 to 0.8 to H and finally to 1. Particularly, in the extreme non-IID case with $\xi = 1$, to achieve an accuracy of 90% on MNIST, FL-DP^{3}S, Cluster, FedAvg, and FedSAE require 62, 122, 127, and 259 rounds of training, respectively.

This performance improvement is mainly attributed to the fact that, compared with the three benchmarks, FL-DP^{3}S efficiently diversifies the participants' datasets in each round of training by rationally exploiting the data profiles of clients. To verify this, we adopt the metric called group earth mover's distance (GEMD) to quantify the diversity of data samples regarding the selected clients [15], i.e.,

$$G(\mathcal{C}_t) = \sum_{j=1}^{N} \left\| \frac{\sum_{c\in\mathcal{C}_t} n_c \mathcal{P}_c(y=j)}{\sum_{c\in\mathcal{C}_t} n_c} - \mathcal{P}_g(y=j) \right\|. \tag{15}$$

In (15), N represents the number of different classes in the union of all clients' datasets $\mathbf{D}_g$, i.e., $\mathbf{D}_g = \cup_{c\in\mathcal{C}}\mathbf{D}_c$. Besides, $\mathcal{P}_c(y=j)$ and $\mathcal{P}_g(y=j)$ denote the proportion of the number of the j-th class data in the local dataset of client c and that in the union of datasets $\mathbf{D}_g$, respectively. And, a smaller $G(\mathcal{C}_t)$ means that the data samples in the union of participants' datasets are more diverse. For both the MNIST and Fashion-MNIST datasets, Fig. 2 demonstrates the GEMD achieved by FL-DP^{3}S and three baseline FL algorithms. Combining Figs. 1 and 2, it can be observed that, in terms of the training convergence rate and accuracy, the algorithm achieving a lower GEMD commonly outperforms those with the higher GEMD. This is consistent with our previous analysis and argument that diversifying the data samples in each training round potentially improves the convergence of FL on non-IID data.

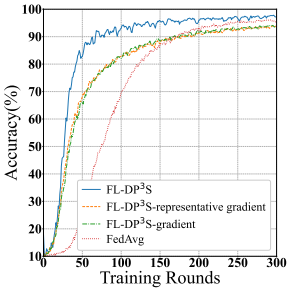


Fig. 3: Accuracy *v.s.* training rounds.

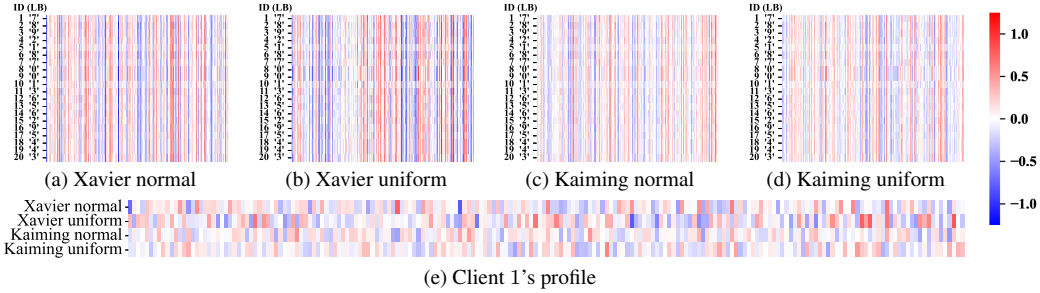


Fig. 4: Visualization of clients' profiles for the cases with different initialization schemes. In (a)-(d), ID denotes the indices of clients, and LB denotes the class label of the data samples possessed by the client. In (e), client 1 is taken as an example to clearly demonstrate how the profiles is impacted by the parameter initialization.

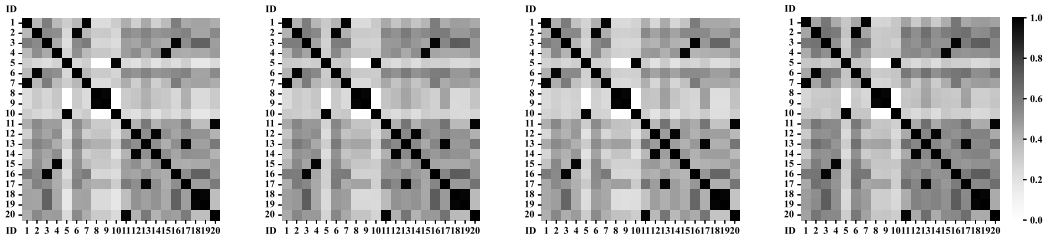


Fig. 5: Visualization of the similarity kernel matrix, where the color of the square at the intersection of row i and column j illustrates the similarity of clients i and j. The darker the color, the more similar the two clients are.

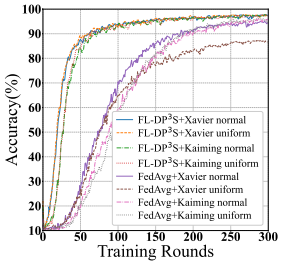


Fig. 6: Accuracy *v.s.* training rounds.

To further investigate the effects of the profiling and parameter initialization on FL-DP^{3}S, we conducted additional experiments to evaluate the performance of FL-DP^{3}S using different profiling methods and parameter initialization schemes.

Intuitively, the profile of each local dataset is determined by the initial global model parameters, so the distribution will depend on the initialization. Nevertheless, we would like to emphasize that the similarity of clients is, in fact, determined by their datasets while does not rely on parameter initialization. As such, for our proposed algorithm, the subsequent client selection and final performance would not be substantially affected by the parameter initialization. To demonstrate this, we consider the scenario with $C = 20$ clients as an example and illustrate the clients' profiles and similarities on MNIST with $\xi = 1$ when using four popular parameter initialization schemes, i.e., Kaiming uniform [32], Kaiming normal [32], Xavier uniform [25], and Xavier normal [25], in Figs. 4 and 5. By comparing Figs. 4 (a)-(e), it can be readily observed that the profile of a generic client is significantly affected by the adopted initialization scheme. However, as demonstrated in Figs. 5 (a)-(d), the difference between the similarity kernel matrices regarding the four initialization schemes is imperceptible. Furthermore, we have conducted additional experiments with $C = 100$ clients under different parameter initialization schemes and summarize the experimental results on MNIST with $\xi = 1$ in Fig. 6. This figure reveals that under different parameter initialization schemes, the performance of our proposed algorithm remains relatively consistent, while that of FedAvg is highly sensitive to parameter initialization.

Then, to further highlight the contributions of the FC-1 profiling, we conducted experiments to compare its performance with other commonly used profiling methods (e.g., profiles based on the gradients or the representative gradients [31]). The performances on MNIST with $\xi = 1$ are presented in Fig. 3. This figure shows that by implementing our proposed FC-1-based profiling (i.e., FL-DP^{3}S), the training convergence rate and accuracy can be significantly improved.

5. CONCLUSION

In this work, we have proposed a novel CS algorithm called FL-DP^{3}S to improve the performance of FL in the presence of non-IID data. Particularly, we have theoretically analyzed the effect of CS in FL by resorting to the conclusions regarding the effect of mini-batch sampling in the SGD update and proposed the FL-DP^{3}S algorithm by jointly leveraging data profiling and DPP sampling techniques. Extensive experimental results showed that compared with three baseline FL algorithms, our proposed FL-DP^{3}S algorithm could enhance the diversity of the training dataset in each training round of FL, quantified by GEMD, thereby improving the performance in terms of the convergence rate and achieved training accuracy.

6. ACKNOWLEDGMENTS

This paper was supported by the National Natural Science Foundation of China (62271413, 62271513) and Chinese Universities Scientific Fund (2452017560).

7. REFERENCES

[1] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, 2016.

[2] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, 2020.

[3] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017, pp. 1273–1282.

[5] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surv. & Tut.*, vol. 22, no. 3, pp. 2031–2063, 2020.

[6] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[7] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, 2019.

[8] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[9] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The non-IID data quagmire of decentralized machine learning," in *Proc. ICML*, 2020, pp. 4387–4398.

[10] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," *arXiv preprint arXiv:1806.00582*, 2018.

[11] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL for wireless networks: Cooperative learning mechanism using non-IID data," in *Proc. IEEE ICC*, 2020, pp. 1–7.

[12] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data," *arXiv preprint arXiv:1811.11479*, 2018.

[13] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, "FedMix: Approximation of mixup under mean augmented federated learning," in *Proc. ICLR*, 2020.

[14] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, and L. Liang, "Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications," in *Proc. ICCD*, 2019, pp. 246–254.

[15] J. Ma, X. Sun, W. Xia, X. Wang, X. Chen, and H. Zhu, "Client selection based on label quantity information for federated learning," in *Proc. IEEE PIMRC*, 2021, pp. 1–6.

[16] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning: Submodularity and convergence analysis," in *Proc. ICML'21 WKSHP on Federated Learning for User Privacy and Data Confidentiality*, 2021.

[17] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE INFOCOM*, 2020, pp. 1698–1707.

[18] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[19] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, pp. 177–186. Springer, 2010.

[20] P. Zhao and T. Zhang, "Accelerating minibatch stochastic gradient descent using stratified sampling," *arXiv preprint arXiv:1405.3080*, 2014.

[21] C. Zhang, C. Öztireli, S. Mandt, and G. Salv, "Active mini-batch sampling using repulsive point processes," in *Proc. AAAI*, 2019, pp. 5741–5748.

[22] C. Zhang, H. Kjellstrom, and S. Mandt, "Determinantal point processes for mini-batch diversification," in *In Proc. UAI*, 2017.

[23] W. Wu, L. He, W. Lin, R. Mao, C. Huang, and W. Song, "FedProf: Optimizing federated learning with dynamic data profiling," *arXiv preprint arXiv:2102.01733*, 2021.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[27] A. Kulesza and B. Taskar, "Determinantal point processes for machine learning," *Foundations and Trends® in Machine Learning*, vol. 5, no. 2–3, pp. 123–286, 2012.

[28] A. Kulesza and B. Taskar, "k-DPPs: Fixed-size determinantal point processes," in *Proc. ICML*, 2011, pp. 1193–1200.

[29] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[30] L. Li, M. Duan, D. Liu, Y. Zhang, A. Ren, X. Chen, Y. Tan, and C. Wang, "FedSAE: A novel self-adaptive federated learning framework in heterogeneous systems," in *Proc. IJCNN*, 2021, pp. 1–10.

[31] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi, "Clustered sampling: Low-variance and improved representativity for clients selection in federated learning," in *Proc. ICML*, 2021, pp. 3407–3416.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. ICCV*, 2015, pp. 1026–1034.