AALBORG UNIVERSITY

A Contribution-Based Device Selection Scheme in Federated Learning

Pandey, Shashi Raj; Nguyen, Lam D.; Popovski, Petar

A Contribution-based Device Selection Scheme in Federated Learning

Shashi Raj Pandey, *IEEE Member*, Lam D. Nguyen, *IEEE Member*, and Petar Popovski, *IEEE Fellow*

Abstract—In a Federated Learning (FL) setup, a number of devices contribute to the training of a common model. We present a method for selecting the devices that provide updates in order to achieve improved generalization, fast convergence, and better device-level performance. We formulate a min-max optimization problem and decompose it into a primal-dual setup, where the duality gap is used to quantify the device-level performance. Our strategy combines *exploration* of data freshness through a random device selection with *exploitation* through simplified estimates of device contributions. This improves the performance of the trained model both in terms of generalization and personalization. A modified Truncated Monte-Carlo (TMC) method is applied during the exploitation phase to estimate the device's contribution and lower the communication overhead. The experimental results show that the proposed approach has a competitive performance, with lower communication overhead and competitive personalization performance against the baseline schemes.

Index Terms—federated learning, device selection, exploration, exploitation, personalization, generalization

I. INTRODUCTION

Federated Learning (FL), as introduced by McMahan et. al. [1], discusses unbalanced and non-i.i.d. (independent and identical distribution) data partitioning across a massive number of unreliable devices, coordinating with a central server, to distributively train learning models without sharing the actual data. In practice, the data samples are generated through device's usage, such as interactions with applications, results to such statistical heterogeneity. Towards that, related works primarily focus on improving the model performance by tackling data properties, i.e., statistical challenges in the FL [1], [2]. Noticeably, in the initial work [1], the authors show that their proposed Federated Averaging (FedAvg) algorithm empirically works well with non-i.i.d. data. However, the accuracy of FedAvg varies differently for different datasets, as observed in the existing methods [2], [3], and how client selection is made [3]–[5]. For instance, the authors in [3], [4], [6] discussed the impact of having heterogeneous clients, given time requirements for per round training execution, during the decentralized model training over unreliable wireless networks. In doing so, more devices are packed within a training round to improve model performance; however, this lead to consumption of excessive communication resources and larger communication rounds to attain a level of global model accuracy. Also, all received local updates are directly aggregated during model aggregation [1], [3], [6], [7]; thus, fairly ignoring their individual contributions and the rationale

Shashi Raj Pandey, Lam D. Nguyen, and Petar Popovski are with Connectivity Section, Department of Electronic Systems, Aalborg University, Denmark. Email: {srp, ndl, petarp}@es.aau.dk.
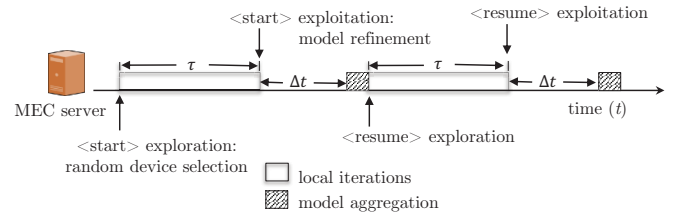

Fig. 1: An illustration of device selection strategy with a mix of exploration and exploitation.

behind selecting them. In line with that, the authors in [1]–[3] revealed that adding local computations can dramatically increase communication efficiency and improve the trained model performance. However, this additional computational load may be prohibitive for some devices.

In principle, the aforementioned issues appear primarily as a result of selecting ill-conditioned devices in the training procedure, without evaluating their marginal contribution in improving the model performance. In fact, in FL literature [2]–[4], device selection problem over wireless networks has remained an overarching challenge, particularly, due to two reasons: *first*, owing to the consequences of statistical and system-level heterogeneity, uniformly random selection of device may lead to slower convergence and poor model performance across devices [6], [8]; *second*, devices with trivial contributions may get scheduled in model training process which only adds larger communication overhead [2]. Moreover, device's contribution are unknown a priori, and its estimation is not straightforward. Some recent works [5], [7] introduced quality-aware device selection strategy with multi-arm bandits (MAB) method in the absence of estimates on available computation-communication resources with fixed dataset; however, they focus on minimization of the convergence time, leaving aside the impact of device selection on the trained model performance, particularly on unseen data, as well as the device-level performance.

In this work, we are interested in, and provide a solution to, the problem of device selection and its impact on model performance defined in terms of *generalization*, i.e., how the trained model work on unseen/new data samples, and *personalization*, i.e., the device-level performance. Different from the related literature, our approach considers the impact of device selection procedure in the decentralized training framework of FL with fairness guarantees in the model performance. An example, compared with [9], [10], where fairness is associated with device scheduling opportunities following statistical and/or system-level heterogeneity, our perspective and definition on fairness guarantees is unique. The term *fairness* captures the discrepancies in device-level model performance, i.e., personalization performance, appeared as

a direct consequence of statistical heterogeneity and device selection approach during model aggregation. Referring to Fig. 1, similar to FedAvg [1], we start by random device selection in FL during the exploration phase at each global iteration. Subsequently, in the exploitation phase, the focus is on contribution-based model aggregation for refining the global model, which, although fundamental, is not addressed by FedAvg. As the contributions of devices in enhancing the global model performance is unknown a priori, we develop a mechanism that first provides an estimation of device's contribution, which is followed by model aggregation with the objective of ensuring fairness across selected device. This way, we bring balance in improving both generalization and device-level performance of the trained model. The summary of our contributions are as follows.

- We develop a cost-efficient, simplified device selection mechanism in the FL setting that captures the notion of fairness in terms of personalized performance. It lowers the communication overhead while enhancing the generalization performance of the trained model. Therein, as an initial step, the problem is revealed as a min-max optimization problem that specifically captures the aforementioned joint objectives.

- To solve the problem of device selection, we propose a strategic mix of exploration and exploitation during each global iteration. This is in contrast to periodic exploration that particularly leads to poor model performance. In particular, we unleash a mechanism that both assess and incorporates the value of contribution of each randomly selected device during model training.

- We propose a contribution-based device selection (CDS) technique for model aggregation with the modified Truncated Monte-Carlo (TMC) [11] method during exploitation phase within random device selection rounds, i.e., the exploration phase. We show this technique eliminates the need of adding local computations to lower communication overhead and improve model convergence.

- The simulation results show that the proposed method, while offering a 4X decrease in the communication rounds, still brings a small performance improvement in terms of personalization.

Overall, the chief contribution of this work is that it incorporates a contribution-based device selection strategy for model aggregation in FL. This improves device-level performance in several aspects simultaneously: generalization, low communication overhead, and fast convergence.

II. PROBLEM SETTING

We consider a supervised learning problem in a FL setting where each available device (client) $m \in \mathcal{M}$ in the training process has a local data set $\mathcal{D}_m$ of size D_m such that the overall training data size $D = \sum_{m=1}^{M} D_m$. The devices communicate with the central aggregator (e.g., a multi-access edge computing (MEC) server) via orthogonal frequency division multiple access (OFDMA). For simplicity, downlink is treated as error-free. The data samples are defined as a collection set of input-output pairs $\{x_i, y_i\}_{i=1}^{D_m}$, where, respectively, $x_i \in \mathcal{X}$ characterizes the feature space of data sample with corresponding labels $y_i \in \mathcal{Y}$, $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}$. The goal of the

central aggregator orchestrating the distributed training process is to learn a predictor $F(w) : \mathcal{X} \to \mathcal{Y} \in \mathbb{R}$. The predictor performance is associated with learning of a model parameter $w \in \mathbb{R}^d$ that captures the empirical risk on all distributed data samples as the regularized finite-sum objective of the form

$$\min_{w \in \mathbb{R}^d} J(w) + \lambda g(w) \quad \text{where} \quad J(w) := \sum_{m=1}^{M} \frac{D_m}{D} \cdot J_m(w), \tag{1}$$

$J_m(w) := \frac{1}{D_m} \sum_{i=1}^{D_m} f_i(w)$, $f_i(w)$ is the loss function that depends upon the choice of learning algorithms [1], [6]; λ is a regularizer and $g(w)$ is a well-known regularization function that controls the complexity and captures the device-level performance. We assume f_i is $(1/\gamma)$-smooth and 1-strongly convex of g and continuous differentiable to ensure convergence of the solution; a common assumption made in several existing works [6], [12], [13].

Solving FL problem with FedAvg [1]: The learning problem in (1) is solved following a two-step approach in an iterative manner. *Step 1*: In the tth update of the global iteration, a fraction of selected devices m iterates over its on-device training data $\mathcal{D}_m$ to solve the local problem

$$w_m^{(t)} = \arg\min_{w_m \in \mathbb{R}^d} J_m(w_m | w^{(t-1)}), \tag{2}$$

and communicates the parameters $w_m^{(t)}$ to MEC. *Step 2*: At the MEC, the collected local parameters are simply aggregated as

$$w^{(t+1)} = \sum_{m=1}^{M} \frac{D_m}{D} \cdot w_m^{(t)}, \tag{3}$$

and broadcast back to the fraction of participating devices for the next global iteration. This process continues until a desired level of model accuracy is achieved. We observe, this approach of model aggregation imposes an equal value of contribution to the selected local model parameters without considering its consequences on the model performance; e.g. a deterioration due to poor local updates. This eventually exerts larger communication rounds to converge to a high-quality global model while alongside inducing irregularities in the device-level performances.

A. Device Selection in FL with Fairness Guarantees

We note that a plain FL [1] ignores the device-level performance, particularly caused due to small data, limited computational capabilities and available communication channels to share parameters with the server, or having devices with poor local updates. Hence, the random device selection procedure may incur costs in terms of overhead. While it is also significant to maintain device-level performance, the cost overhead due to uncertainty in random device selection should be minimized. Therein, we first develop a natural extension to plain FL which captures the risk associated with the local accuracy loss using model w in each device m as

$$R_m^{(t)}(F) = \mathop{\mathbb{E}}_{\{X^m, Y^m\} \sim \mathcal{P}_m} \left[J(F(X^m; w), Y^m) \right], \tag{4}$$

where $\mathcal{P}_m$ is the underlying distribution of local data samples. Here, we look forward to characterize a predictor F that works well with dataset in all associated devices. Basically,

in a risk-averse FL setting, we aim to minimize the maximum risk across all participating devices, ensuring improved personalization and generalization performance of the trained model. Subsequently, we formulate a min-max optimization problem: it jointly selects the devices contributing the most in improving the model performance and minimize the maximal risk to ensure fairness amongst these associated devices:

$$\textbf{P:} \min_{w} \left[\max_{\tilde{\mathcal{M}}:=\{m\}} \sum_{m \in \tilde{\mathcal{M}}} R_m^{(t)}(F) \right], \quad (5)$$

$$\text{s.t.} \quad w \in \arg \min_{w \in \mathbb{R}^d} R_m^{(t)}(F_m), \quad \forall m \in \tilde{\mathcal{M}}, \quad (6)$$

$$R_m^{(t)}(F_m) \le \theta_m^{\text{th}}, \forall m \in \tilde{\mathcal{M}}, \quad (7)$$

$$\tilde{\mathcal{M}} \subseteq \mathcal{M}. \quad (8)$$

where $\tilde{\mathcal{M}}$ is a subset of devices selected based on their local contribution to the global model and satisfying fairness constraint (7) imposed as θ_m^{th}, defined with details in Section III-A. To solve **P**, we need to efficiently derive $\tilde{\mathcal{M}}$ in prior, which is not straightforward. For that purpose, we use the primal-dual decomposition method and capture the contribution of devices in terms of the quality of obtained dual solution, that is defined in details in the following Section III-A.

Next, we define and discuss the incurred time costs from the perspective of random device selection procedure.

B. Time cost

In each global iteration, the time cost for device m has two components: (i) local computation time: $t_m^{\text{comp}} = c_m D_m / f_m$, where c_m is CPU-cycles requirement to compute per bit data; f_m is CPU-cycle frequency; and (ii) communication time: $t_m^{\text{comm}} = s_m/(r_m B_m)$, where s_m is the size of transmitted data in bits; $r_m = \log_2(1 + \frac{p_m h_m}{N_0 B_m})$ is the rate; p_m is the transmission power; h_m is the channel gain; N_0 is the noise power spectral density and B_m is the corresponding bandwidth. Then, the corresponding execution time cost t^{exec} per global iteration due to random device scheduling is proportional to $t^{\text{cost}} := \max_m \{t_m^{\text{comp}} + t_m^{\text{comm}}\}, \forall m \in \mathcal{M}$. This captures the fact that minimization of t_τ^{cost} while ensuring fairness of the collaborative training in solving **P**, there is a need for careful device selection $\tilde{\mathcal{M}}$ and subsequent model aggregation.

In the following, we first reflect the risk minimization problem **P** for a known set of selected devices in a primal-dual setting. Then, we propose a cost-efficient contribution-based device selection approach with fairness guarantees by exploiting the modified TMC [11], [14] method to solve the **P** in subsequent global iterations.

III. CONTRIBUTION-BASED DEVICE SCHEDULING ALGORITHM

A. Device's local contribution in a distributed setting

We revisit the global problem (1) in its dual optimization form [13] with M devices to evaluate their local contribution in solving the learning problem. The corresponding dual optimization problem of (1) for a convex loss function f is

$$\max_{\alpha \in \mathbb{R}^D} \mathcal{R}(\alpha) := \frac{1}{D} \sum_{i=1}^{D} -f_i^*(-\alpha_i) - \lambda g^*(\phi(\alpha)), \quad (9)$$

where f_i^* and g^* are the convex conjugates of f_i and g, respectively [13]; $\alpha \in \mathbb{R}^D$ is the dual variable mapping to

---

Algorithm 1 CDS: Contribution-based Device Selection

1: `Input`: M is the number of devices in set $\mathcal{M}$; $C \in (0,1]$ is fraction of random devices selected; B is the local minibatch size; E is the number of epochs, and η is the learning rate.
2: `Initialize`: $\phi^1, C, B, E, \Delta t, \epsilon$ and η.
   `//Exploration://`
3: **for all** global iteration $t \in \{1, 2, \dots, \tau\}$ **do**
4:    Set $\mathcal{S}_t \subseteq \mathcal{M}$ with $\max(1, C \times M)$ devices;
5:    **for all** $m \in \mathcal{S}_t$ **do**
6:       Execute `DeviceUpdate`(m, ϕ^t);
7:    **end for**
   `//Exploitation://`
8:    Start exploitation timer $t' = 0$;
9:    **while** $t' < \Delta t$ **do**
10:      Define $\pi^{t'}$ as random permutation of the selected devices;
11:      Set $v_0^{t'} = V(\emptyset)$;
12:      **for all** $m \in \mathcal{S}_{t'}$ **do**
13:         **if** $|V(\phi^{t'}) - v_{m-1}^{t'}| < \epsilon$ **then**
14:            $v_m^{t'} = v_{m-1}^{t'}$;
15:         **else**
16:            Set $v_m^{t'} \leftarrow V(\{\pi^{t'}[1], \dots, \pi^{t'}[m]\})$;
17:         **end if**
18:         $\beta_{\pi^{t'}[m]} \leftarrow \frac{t'-1}{t'} \beta_{\pi^{t'-1}[m]} + \frac{1}{t'}(v_m^{t'} - v_{m-1}^{t'})$;
19:      **end for**
20:      Update $t' = t' + 1$;
21:   **end while**
22:   Sort $\beta_{\pi^{t'}[m]}$ in descending order;
23:   Obtain $\tilde{\mathcal{M}}$ as per contributions $\beta_{\pi^{t'}[m]}$ and update global variable ϕ^t (13);
24: **end for**
   `DeviceUpdate`(m, ϕ^t):
25: Solve the local sub-problem (10);
26: Update dual variables using (12);
27: **Return** $\Delta \phi_{[m]}^t, \forall m \in \mathcal{M}$;

---

the primal candidate vector; and $\phi(\alpha) = \frac{1}{\lambda D} X\alpha$. Here, we define $X \in \mathbb{R}^{d \times D_m}$ as a matrix with columns having data points for $i \in \mathcal{D}_m, \forall m$. Then, having the optimal value of dual variable α^* in (9), we obtain the optimal solution[1] of (1) as $w(\alpha^*) = \nabla g^*(\phi(\alpha^*))$ [4]. Hereafter, we use $\phi \in \mathbb{R}^d$ for $\phi(\alpha)$ for simplicity, and define a weight vector $\varrho_{[m]} \in \mathbb{R}^D$ at the local sub-problem m with its elements zero for the unavailable data points. As the consequence of the properties of f_i and g, we obtain the approximate solution to the local sub-problem:

$$\max_{\varrho_{[m]} \in \mathbb{R}^D} \mathcal{R}_m(\varrho_{[m]}; \phi, \alpha_{[m]}), \quad (10)$$

defined by the dual variables $\alpha_{[m]}, \varrho_{[m]}$. Here, we have $\mathcal{R}_m(\varrho_{[m]}; \phi, \alpha_{[m]}) = -\frac{1}{M} - \langle \nabla(\lambda g^*(\phi(\alpha))), \varrho_{[m]} \rangle - \frac{\lambda}{2} \|\frac{1}{\lambda D} X_{[m]} \varrho_{[m]}\|^2$ with $X_{[m]}$ as a matrix with columns having data points for $i \in \mathcal{D}_m$, and zero padded otherwise. Each selected device $m \in \mathcal{M}$ iterates over its computational

---

[1]Finding the optimal solution follows an iterative process to attain a global accuracy $0 \le \epsilon \le 1$ (i.e., $\mathbb{E}[\mathcal{R}(\alpha) - \mathcal{R}(\alpha^*)] < \epsilon$).

resources using any arbitrary solver to solve its local problem (10) for a defined local relative θ_m^{th} accuracy, i.e., the maximal risk in **P** in terms of fairness, that characterizes the quality of the local solution and produces a random output $\varrho_{[m]}$ satisfying

$$\mathbb{E}\left[\boldsymbol{\mathcal{R}}_m(\varrho_{[m]}^*) - \boldsymbol{\mathcal{R}}_m(\varrho_{[m]})\right] \leq \theta_m^{\text{th}}\left[\boldsymbol{\mathcal{R}}_m(\varrho_{[m]}^*) - \boldsymbol{\mathcal{R}}_m(0)\right]. \quad (11)$$

Then, the local dual variable is updated as follows:

$$\alpha_{[m]}^{t+1} := \alpha_{[m]}^t + \varrho_{[m]}^t, \forall m \in \mathcal{M}. \quad (12)$$

The selected devices then broadcast the local parameter defined as $\Delta\phi_{[m]}^t := \frac{1}{\lambda D}X_{[m]}\varrho_{[m]}^t$ to the MEC server along with its local relative accuracy, which completes the exploration phase. Next, considering all selected devices contribute equally, which is the usual case of FedAvg, the MEC server can simply aggregate the local parameters as

$$\phi^{t+1} := \phi^t + \frac{1}{M}\sum_{m=1}^{M}\Delta\phi_{[m]}^t. \quad (13)$$

and share ϕ^{t+1} back to the randomly selected devices to again solve (10). However, instantiating random device selection in each exploration phase with the global model using FedAvg demerits local model performance, particularly poor personalization and large variance in model accuracy across devices. This is because the technique does not specifically consider the actual contribution of each device in the model training, and tune parameters accordingly to minimize the empirical risk at worse performing devices; this leads to the increase of the number of communication rounds for convergence. Therefore, we depart from the naive approach of model aggregation and introduce an additional exploitation phase, where the contributions made by selected devices are taken into consideration.

In the following, we present our proposed approach that utilizes the contribution of each device within the exploration phase in improving the model performance and lowering the long-term communication overhead.

B. Contribution-based device selection with modified TMC

Within each round of global iteration that executes random device scheduling, we first eliminate the frequency of transmitting the local dual variables through an estimation of TMC at the MEC, leading to an efficient parameter aggregation scheme. Specifically, following the value of contribution of each selected device, we operate Δt rounds of random permutations on received dual variables to perform the model aggregation before proceeding to the next global iteration with random device selection strategy. We use the estimation of the contribution of each selected device $m \in \mathcal{S}_t$ in round t as

$$\beta_m = \frac{1}{|\mathcal{S}_t|!}\sum_{\pi \in \Pi(\Delta\phi_{[m]}^t)}[V(P_m^\pi \cup \{m\}) - V(P_m^\pi)], \quad (14)$$

where $V(\cdot)$ is the standard valuation of local parameter defined as its performance score while contributing to improve the model accuracy, similar to [11]. $\pi \in \Pi(\Delta\phi_{[m]}^t)$ characterizes the permutation of devices and P_m^π is the preceding devices selected for model aggregation. The details are described in Algorithm 1. Once the exploration phase is executed, using

(14), the MEC proceeds to the exploitation phase (lines 8–21) to estimate the contributions of each selected device parameters in improving the model performance (lines 12–20). Then, the devices are sorted based on their contributions, after which the final model is obtained (line 23) to execute the next round of exploration.

IV. PERFORMANCE EVALUATION

We have simulated the MEC environment with network parameters defined in [3] and conducted extensive experiments with well-known MNIST datasets [15]. We have taken into account the statistical heterogeneity involving the non-i.i.d. characteristics and unbalanced dataset, in order to appropriately setup data across the devices [6]. We shuffle and divide dataset at each device, where data samples are drawn particularly from two labels out of ten for MNIST. We have considered $M = 100$ devices on a single server with 26 core Intel Xeon 2.6 GHz, 256 GB RAM, 4 TB, Nvidia V100 GPU, Ubuntu OS. Accordingly, we have normalized the time cost per global iteration and also assumed that the devices and server are synchronized. We have then evaluated the model performance by considering the impact of device selection strategy during the model aggregation. For training, each device uses $E = 10$ epochs, the batch size $B = 10$, and the learning rate $\eta = 0.01$ as a default. Further, in all of our evaluations, unless specified, we set the value of $C = 0.1$, which exhibits a good balance between computational efficiency and convergence rate, as demonstrated experimentally in [1]. The default value of Δt is set to 1. We compare CDS against (a) two intuitive baselines: 1) Random Sampling method which choose devices randomly, 2) Greedy Algorithm [16], in which the server collects the local gradients of all devices and chooses a subset of devices providing the greatest marginal gain, and (b) a more competitive approach, namely FedFomo [17], customized for personalized federated updates of each device.

Fig. 2a shows that that our proposed contribution-based device selection approach achieves a high accuracy, approximately 97%, while using substantially fewer communication rounds compared to the Greedy and Random Sampling algorithms. This is due to the fact that we are strategically selecting device updates based on their contribution in improving the model performance during model aggregation. This is further supported by Table I, where we present a comparative analysis on the execution time cost t^{exec} in achieving around 80% of the target device-level model accuracy and the required number of communication rounds. We note a relaxed threshold accuracy allows a larger set of participating devices, and thus, leads better generalization during model training. In all cases, we see CDS demonstrates better and stable performance than these baselines, and a competitive model accuracy as compared with FedFomo. Similar observations pertain to the training loss of different sampling strategies, as shown in Fig. 2b. In Fig. 2c, we evaluate and compare the average device-level performance as the performance of the trained model on their local test dataset. Interestingly, we see that CDS provides improvement in the device-level performance: 6.54% and 4.78% higher than Greedy method and Random sampling, respectively, as well as 2.54% compared to FedFomo. This is due to the appropriateness of the local updates that are aggregate in the
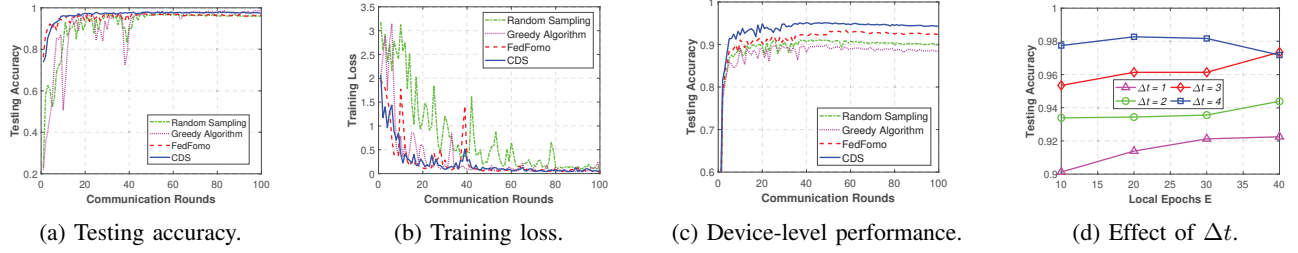
Fig. 2: Comparative performance analysis with different sampling strategies on: (a) testing accuracy, (b) training loss, (c) personalization, i.e., the device-level performance, and (d) the impact of Δt against adding more local computations during local training.

TABLE I: Comparison of execution time cost t^{exec} (in sec).

Algorithms	Average time to achieve accuracy of 80%, and number rounds to achieve	
	Average t^{exec}	Number of rounds
Random Sampling [1]	0.521664	5±1
Greedy-based solution [16]	1.768358	9±1
FedFomo [17]	0.338333	2±1
CDS - Our proposed	0.335492	2±1

model following the contributions of the selected devices in every global round. Furthermore, Random Sampling shows better performance in terms of personalization compared to the Greedy. This is reasonable as it explores the available devices to improve personalization, while the Greedy approach focuses only on improving the performance of the global model.

Finally, in Fig. 2d, we analyze the significance of Δt on eliminating the time cost required for the model convergence to a target accuracy. The insight is that, instead of adding more computation per device to lower the communication overhead [1], we perform contribution-based model aggregation that compensates the increase in local computation costs and the resulting higher t^{cost} for achieving better convergence. Specifically, relaxing Δt allows more rounds of random permutation to explore the contribution of local model updates. In return, having broader exposure of the contributing devices results in better device selection strategy for model aggregation and a better model performance as shown in Fig. 2d, while saving local computations (epoch) at the devices. We also observe the impact of relaxing Δt as a reference to evaluate and compare model performance against adding more local computations. For example, when $E = 40$, we see no significant improvement in the model performance even though we increase Δt by more than 3 rounds, which is due to the optimized device selection strategy.

V. CONCLUSION

We have presented a simplified solution of the device selection problem in Federated Learning (FL), aiming to jointly improve model performance and lower the communication costs. In doing so, we have first formulated a min-max optimization problem. to solve it, we have developed a strategy that constitutes a mixture of exploration phase, where random selection of devices is made, similar to the plain FL approach but under a primal-dual setting of the learning problem, and an additional exploitation scheme that quantifies the contribution

of selected devices in improving the model performance via efficient model aggregation. Extensive simulations on real-world dataset have demonstrated the efficacy of the proposed approach against the baselines in improving model performance, i.e., better generalization and personalization, lowering communication costs, and achieving fast convergence rate.

REFERENCES

[1] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, April 2017.

[2] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE International Conference on Communications (ICC)*, May 2019.

[4] S. R. Pandey *et al.*, "A crowdsourcing framework for on-device federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3241–3256, Feb. 2020.

[5] N. Yoshida *et al.*, "Mab-based client selection for federated learning with uncertain resources in mobile networks," in *IEEE Globecom Workshops (GC Wkshps)*, Dec. 2020.

[6] S. R. Pandey *et al.*, "Edge-assisted democratized learning toward federated analytics," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 572–588, June 2021.

[7] W. Xia *et al.*, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7108–7123, July 2020.

[8] M. N. Nguyen *et al.*, "Self-organizing democratized learning: Towards large-scale distributed learning systems," *arXiv preprint arXiv:2007.03278*, 2020.

[9] H. H. Yang *et al.*, "Age-based scheduling policy for federated learning in mobile edge networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[10] K. Guo *et al.*, "Dynamic scheduling for heterogeneous federated learning in private 5g edge networks," *IEEE Journal of Selected Topics in Signal Processing*, 2021.

[11] R. Jia *et al.*, "Towards efficient data valuation based on the shapley value," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, April 2019.

[12] F. Hanzely *et al.*, "Lower bounds and optimal algorithms for personalized federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2304–2315, 2020.

[13] S. Boyd *et al.*, *Convex optimization*. Cambridge university press, March 2004.

[14] L. D. Nguyen *et al.*, "A marketplace for trading ai models based on blockchain and incentives for iot data," *arXiv preprint arXiv:2112.02870*, 2021.

[15] "Mnist handwritten digit database, yann lecun, corinna cortes and chris burges." http://yann.lecun.com/exdb/mnist/. (Accessed on 03/02/2022).

[16] R. Balakrishnan *et al.*, "Diverse client selection for federated learning: Submodularity and convergence analysis," in *International Conference on Machine Learning Society (ICML)*, July 2021.

[17] M. Zhang *et al.*, "Personalized federated learning with first order model optimization," in *International Conference on Learning Representations (ICLR)*, May 2021.