2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS) | 978-1-6654-7315-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICPADS56603.2022.00079

Heterogeneous Federated Learning for Balancing Job Completion Time and Model Accuracy

Ruiting Zhou*, Ruobei Wang*, Jieling Yu*, Bo Li† and Yuqing Li*
*School of Cyber Science and Engineering, Wuhan University, China
†Department of Computer Science and Engineering, Hong Kong University of Science and Technology, HongKong
Email: {ruitingzhou, wrb.math.cs, yjling}@whu.edu.cn, bli@cs.ust.hk, liyuqing0622@gmail.com

***Abstract*—Federated Learning (FL) is a secure distributed learning paradigm, which enables potentially a large number of devices to collaboratively train a global model based on their local dataset. FL exhibits two distinctive features in job requirement and client participation, where FL jobs may have different training criteria, and clients possess diverse device capabilities and data characteristics. In order to capture such heterogeneities, this paper proposes a new FL framework, *Hca*, which aims to strike a balance between the job completion time and model accuracy. Specifically, *Hca* builds upon a number of innovations in the following three phases: i) *pre-estimation*: we first derive the optimal set of parameters used in training in terms of the number of training rounds, the number of iterations and the number of participating clients in each round; ii) *client selection*: we design a novel device selection algorithm, which selects the most effective clients for participation based on both client historical contributions and data effectiveness; iii) *model aggregation*: we improve the classic FedAvg algorithm by integrating the model loss reduction in consecutive rounds as a weighted factor into aggregation computation. To evaluate the performance and effectiveness of *Hca*, we conduct theoretical analysis and testbed experiments over an FL platform FAVOR. Extensive results show that *Hca* can improve the job completion time by up to 34% and the model accuracy by up to 9.1%, and can reduce the number of communication rounds required in FL by up to 75% compared with two state-of-the-art FL frameworks.**

***Index Terms*—Federated Learning, Client Selection, Time and Accuracy Balancing**

I. Introduction

Federated learning (FL), a privacy-preserving distributed learning paradigm, enables potentially a large number of devices to collectively train a global machine learning (ML) model using the devices' own local dataset [1], [2]. With FL, an server in a cloud is used to coordinate participating clients synchronously in multiple communication rounds. In each round, a small subset of these clients is selected to train an ML model using their local data. After a number of training iterations are performed, clients will send their local model updates to the FL server, which aggregates these updates using an aggregation algorithm (*e.g.,* FedAvg [3]) to update the global model parameters and send them back to clients. This process iterates until a pre-specified model accuracy or the maximum training time (*i.e.,* job completion time) is reached [4].

While the training process of FL is similar to that of conventional ML, FL executions have distinctive features in job requirements and client participation. Improving FL model accuracy is generally achieved by increasing computational loads, which consume a larger amount of energy and need more time. But some FL jobs require making decisions in a short time, such as Gboard - Google keyboard, which makes typing suggestion while typing [5]. The trade-off between time and accuracy brings new challenges and opportunities in FL training [6], [7]. To start with, we need to properly select three critical control parameters including the total number of training rounds, the number of iterations and the number of participating clients in each round [8], [9]. This turns out to be challenging in that there exist intrinsic correlations among these parameters, and their collective impacts on the job completion time or model accuracy are largely unknown. On the other hand, exhaustive search for the optimal values is practically infeasible. Second, FL potentially involves a large number of clients, each with heterogeneous device capability and data characteristics. For instance, the clients with high computing capability and more relevant datasets can contribute more to the quality of the model training. Consequently, how to select the most effective set of clients for participation during each training round becomes critically important.

Existing works in FL have focused on improving model accuracy [10] or reducing energy cost [11], usually with a random client selection algorithm. Considerable efforts have been made to improve the effectiveness of client selection [12]. However, they typically only consider device performances such as local loss or bias, while ignoring a crucial factor - the quality of the data. In FL, a major challenge is that data distribution among clients is unbalanced and not independent and identically distributed (non-IID). Irrelevant or even harmful data can significantly affect the FL training performance. Proper client selection can mitigate this problem. We will further discuss this in detail in Sec. II. To comprehensively capture the heterogeneity of FL, in this paper, we propose a new FL framework, *Hca*, for optimizing the FL training performance. First, we assign weighted indices to the model loss and job completion time to capture the different job training preferences. The value of the index is defined by FL jobs. Minimizing the sum of two weighted factors can strike a

The corresponding author is Bo Li.

This work was supported in part by the NSFC Grants (62072344 and U20A20177), RGC RIF Grant R6021-20, and RGC GRF Grants (16209120 and 16200221).

978-1-6654-7315-6/23/$31.00 ©2023 IEEE
DOI 10.1109/ICPADS56603.2022.00079

balance between job completion time and model accuracy in the training process. Second, to characterize the heterogeneity in device capability and data characteristics, we define two criteria in the client selection process: *historical effectiveness* which is related to the weighted index and captures the estimated convergence rate; and *data effectiveness*, measuring the matching degree between client's dataset and the target job's dataset, as well as the similarity between two participating clients' datasets. The objective is to select the most effective clients with higher values in these two criteria.

To strike a balance between job completion time and model accuracy, we consider the FL training process in three stages: i) *pre-estimate stage*, which aims to derive the three control parameters in polynomial time; ii) *client selection stage*, which focuses on selecting the most effective clients based on historical and data effectiveness. This stage utilizes the three control parameters obtained in the first stage to enable faster and better FL training while satisfying job preference; iii) *model aggregation stage*, which improves the classic *FedAvg* aggregation algorithm to further enhance the FL model training accuracy. The combined optimizations in the three stages not only meet the basic requirements in job completion time and model accuracy, but also maximize the training performance. The problem turns out to be particularly challenging from two aspects, i) neither the job completion time nor the model loss presents an explicit expression related to the three control parameters; ii) the client section problem is a 0-1 quadratic program (QP), which is proven to be NP-hard.

In this context, we propose a new FL framework, *Hca*, to solve the three-stage problem. We first reformulate the trade-off objective in the pre-estimate stage into a function capturing the three control parameters through analysis on the upper bounds of the training time and model accuracy. The reformulated problem can be solved by the block coordinate decent (BCD) algorithm [13]. During the client section stage, we first linearize the 0-1 QP into an integer linear program (ILP). We then use a randomized pairwise rounding algorithm to obtain the required integer solution. Finally, in the model aggregation stage, we select most relevant local updates and integrate the model loss reduction between consecutive rounds as a weighted factor into the aggregation calculation, to further enhance the FL model accuracy.

We conduct extensive experiments to evaluate the performance of *Hca*. The experiments train a CNN model on three common datasets over the FL platform FAVOR [14]. The highlights of the results are: i) the weighted factor α plays a trade-off between the time and accuracy requirements; ii) *Hca* reduces the job completion time by up to 34% and improves the model accuracy by up to 9.1% at the same time when compared with FAIR [15] and FedAvg [3]; iii) *Hca* reduces the number of training rounds required by up to 67% and 75%, when compared to the above two benchmarks.

The rest of the paper is organized as follows. We present the related work in Sec. II. We introduce system model in Sec. III and describe the design of *Hca* in Sec. IV. We evaluate *Hca* in Sec. V and conclude the paper in Sec. VI.

II. Related Work

FL Framework Optimization. Since Mcmahan *et al.* [3] proposed the first FL framework and demonstrated its effectiveness, many efforts have been devoted to improving the performance of FL. Some works [16]–[18] investigated the theoretical convergence guarantees in heterogeneous settings, while others were proposed to improve the structure of FL framework. Briggs *et al.* [19] proposed a hierarchical clustering approach to categorize clients by the similarity of their local updates. Wang *et al.* [20] designed a hierarchical aggregation approach by clustering clients and explored the optimal cluster structure with resource constraints. Wang *et al.* [21] filtered out the irrelevant updates by ameliorating aggregation method. Leroy *et al.* [22] designed an Adam-based per-coordinate averaging strategy for global aggregation. Wang *et al.* [9] proposed a control algorithm to minimize the loss function under a given resource budget. Luo *et al.* [8] analyzed how to optimally select the essential control variables to minimize the total cost while ensuring convergence. None of the existing approaches jointly considered the client selection and heterogeneous requirements of different FL jobs. Our proposal, *Hca* is different in that it aims to jointly optimize the training time and model accuracy through proper selection of training parameters and model aggregation.

FL Client Selection. Takayuki *et al.* [23] presented a client selection method based on the hardware and wireless resources. Ribero *et al.* [24] proposed a client selection strategy by utilizing the progression of clients' weights from an Ornstein-Uhlenbeck process. Wang *et al.* [14] selected clients by leveraging deep reinforcement learning (DRL) technique to speed up convergence. Chen *et al.* [25] proposed a client selection scheme by minimizing the variance of the stochastic gradient. Deng *et al.* [15] constructed a quality-aware selection scheme by learning quality estimation. Such client selection strategies largely rely on the client performance such as local losses or bias, while completely ignoring the effect of clients' data properties. Another relevant work considered data property [26], and designed a method to select high-quality clients and data samples. Different from all the above works, *Hca* designs a comprehensive client selection criterion by incorporating both client's device performance and data properties with proven theoretical guarantee.

III. System Model

A. System Overview

Heterogeneous Job Preferences. We assume that an FL job comes with its maximum completion time requirement C and accuracy demand ε, where ε is the required difference between the model loss and the minimum loss (which is determined by the property of the loss function). As discussed, *Hca*, aims to balance an FL job's completion time and model accuracy and selects the most efficient participating clients for model aggregation, while satisfying two training requirements. Considering FL jobs have different preferences on the completion time and model accuracy, we define a weight index α for a

FL job, to capture the trade-off between the two values, where $\alpha \in [0,1]$. The closer α is to 1, the higher value the FL job puts on the accuracy than the completion time. α affects both the control parameter estimation process before FL training and the client selection process in each training round.

Training Process and Decision Overview. To effectively achieve the unique requirement of time and accuracy trade-off, we consider that a training process consists of three stages. *First*, upon the arrival of an FL job, *Hca* computes three control parameters based on the job preference and requirements: i) K, the total number of training rounds; ii) τ, the number of iterations in each round; and iii) M, the number of participating clients per round. *Second*, in each round, *Hca* evaluates the effectiveness of available clients based on their device capabilities and data characteristics, and selects the most effective clients from N available clients. Let a set of binary variables $\{x_i^k \in \{0,1\} | i \in \mathbb{I}^k, k \in K\}$ denote the selection, where $\mathbb{I}^k$ is the set of available clients in round k. If client i is selected to participate in the training process in round k, $x_i^k = 1$, otherwise $x_i^k = 0$. *Third*, *Hca* performs model aggregation. Stage 2 and stage 3 are carried out under the premise of the control parameters determined in stage 1. The first two stages will be described in details in the next two subsections.

B. Pre-estimation Stage

Problem Formulation. To achieve the trade-off between the two preferences, a weight index α[1] is introduced to merge the two requirements. We formulate the control parameter pre-estimation problem as an optimization problem whose objective is to minimize the weighted completion time and model loss while satisfying job training requirements. We call this problem as PEP (Pre-estimation Problem). Let C_{tot}^K be the total training time after K rounds, and $F(\boldsymbol{w}^K)$ be the model loss after K rounds. $F(\boldsymbol{w}^*)$ is the minimum loss, which is a fixed value. The PEP for each incoming job can be formulated as follows:

$$\textbf{P1}: \quad \min_{K,M,\tau} \quad (1-\alpha)\mathbb{E}[C_{tot}^K] + \alpha \mathbb{E}[F(\boldsymbol{w}^K)] \tag{1}$$

$$\text{s.t.} \quad \mathbb{E}[C_{tot}^K] \leq C, \tag{1a}$$

$$\mathbb{E}[F(\boldsymbol{w}^K)] - F(\boldsymbol{w}^*) \leq \varepsilon, \tag{1b}$$

$$K, M, \tau \in \mathbb{Z}^+, \tag{1c}$$

Constraint (1a) guarantees that the expected training time is less than the time requirement. Constraint (1b) ensures that the expected loss after training K rounds is less than the loss requirement.

C. One-round Client Selection Stage

Selection Criteria. For each training round k, given the collection of available clients $\mathbb{I}^k$, *Hca* selects at least M clients for this round to maximize the quality of the aggregated global model. *Hca* evaluates the effectiveness of client i for the target job from two aspects: i) *historical effectiveness*, which applies to clients that have participated in the target job before. It evaluates the effectiveness of each available client based on staleness mechanism [27] using training time and quality from previous training rounds. The weight parameter α is integrated into this evaluation function to capture the heterogeneity of job preference; and ii) *data effectiveness*, which measures the quality and the fitness of client i's dataset for the target job. Data effectiveness includes the data *relevance* (γ_i^k) between client i and the target job, as well as the data *similarity* ($s^k(i,j)$) between the two participating clients i and j.

Historical Effectiveness. In particular, the historical effectiveness of client i in round k (U_i^k) is defined as:

$$\widehat{U}_i^k = \begin{cases} \dfrac{\alpha \widehat{q}_i^k}{(1-\alpha)\widehat{C}_{tot}^{k,i}}, & \widehat{C}_{tot}^{k,i} \neq 0, \alpha \neq 1 \\ \widehat{q}_i^k & \widehat{C}_{tot}^{k,i} \neq 0, \alpha = 1 \\ -\widehat{C}_{tot}^{k,i} & \widehat{C}_{tot}^{k,i} \neq 0, \alpha = 0 \\ 0, & otherwise \end{cases} \tag{2}$$

where $\widehat{q}_i^k$ and $\widehat{C}_{tot}^{k,i}$ are the estimated utility and estimated completion time in round k, both of which are estimated from historical training information. $\widehat{U}_i^k$ is a concept similar to the training speed, representing the estimated model quality per unit of time. We multiply the weight factor α and $(1-\alpha)$ in front of $\widehat{q}_i^k$ and $\widehat{C}_{tot}^{k,i}$, in order to meet the time and accuracy trade-off. We use the reduction of the model loss to define the accuracy factor in round t, *i.e.,* $q_i^t = F(\boldsymbol{w}^{t-1}) - F(\boldsymbol{w}_i^t)$. So $\widehat{q}_i^k$ is defined as:

$$\widehat{q}_i^k = \begin{cases} \dfrac{\sum_{t=1}^{k-1} q_i^t \beta^{k-t}}{\sum_{t=1}^{k-1} \beta^{k-t}}, & \exists k,\ s.t.\ q_i^k > 0 \\ 0, & otherwise \end{cases} \tag{3}$$

Similarly, the estimated completion time of round k $\widehat{C}_{tot}^{k,i}$ is also obtained from the historical information:

$$\widehat{C}_{tot}^{k,i} = \begin{cases} \dfrac{\sum_{t=1}^{k-1} C_{tot}^{t,i} \beta^{k-t}}{\sum_{t=1}^{k-1} \beta^{k-t}}, & \exists C_{tot}^{k,i} \\ 0, & otherwise \end{cases} \tag{4}$$

where $C_{tot}^{t,i}$ is the completion time of round t.

Data Effectiveness – Relevance: Besides the historical information, to avoid the negative effect of unbalanced and non-IID data distribution on participating clients, we also hope to obtain information from data as a selection criterion under the premise of protecting privacy. First, we define the relevance γ_i^k to measure the matching degree between client i's dataset and the target job's dataset. To protect the privacy of clients and the job, *Hca* adopts private set intersection (PSI) protocol [28], which is a widely used lightweight security protocol. Let Y_i^k denote the set of client i 's labels in round k, and let Y denote the target labels. The dataset of client i in round k is D_i^k. $D_i^{'k}$ represents the dataset that overlaps with the target labels, *i.e.,* $D_i^{'k} = \{(x,y) | y \in Y_i^k \cap Y\}$. We define γ_i^k as follows:

$$\gamma_i^k = \frac{|D_i^{'k}|}{|D_i^k|}, \tag{5}$$

[1]Since completion time and model loss have different range of values, we normalize two values in the same range, according to their actual values in training, which will be further explained in Sec. V-A.

Obviously, the larger γ_i^k is, the more suitable client i is for the training.

Data Effectiveness – Similarity: Clients with similar data appearing in the same training round may waste training resources, as we want to get as much data information as possible from the available clients for training. Therefore, we use a privacy-preserving method to measure the data similarity between clients, and select clients whose data information is as unique as possible to participate in each training round. *Hca* measures the similarity $s^k(i,j)$ of two clients' datasets D_i and D_j by a privacy-preserving method [26][2]. In particular, client i locally generates content embedding vectors $\phi_i^k = \{\phi_{i,m}^k | m \in [S_i^k]\}$, where each embedding vector $\phi_{i,m}^k \in \mathbb{R}^{L_\phi}$. Then a projection matrix $w \in \mathbb{R}^{l_\phi \times L_\phi}$ is selected to encode the L_ϕ-dimensional vector into l_ϕ-dimensional vector $h(\phi_{i,m}^k)$, where $l_\phi < L_\phi$ [29]. Each client computes the projection vector $h(\phi_{i,m}^k) = sgn(w \cdot \phi_{i,m}^k)$, where $sgn(\cdot)$ denotes signum function. Then the sketch of dataset D_i^k is $H_i^k = \sum_{m \in [S_i^k]} h(\phi_{i,m}^k)$. To protect the privacy of each sample, a randomized response mechanism is applied to generate a noisy sketch $\widehat{H}_i^k$ to replace H_i^k. Given the noisy content sketch of each client, the similarities between two clients' datasets D_i^k and D_j^k is defined as:

$$s^k(i,j) = \frac{\widehat{H}_i^k \cdot \widehat{H}_j^k}{|\widehat{H}_i^k||\widehat{H}_j^k|}. \tag{6}$$

If $s^k(i,j)$ is too large, the datasets of clients i and j are too similar, hence the training efficiency by selecting these two clients is low.

Problem Formulation. The goal in the second stage is to maximize the quality of clients and the data diversity[3] at the beginning of round k.

$$\textbf{P2}: \quad \max_{\mathbf{x}^k} \quad \sum_{i=1}^{N} U_i^k x_i^k - \sum_{1 \le i < j \le N} s^k(i,j) x_i^k x_j^k \tag{7}$$

$$\text{s.t.} \quad M \le \sum_{i=1}^{N} x_i^k \le R, \tag{7a}$$

$$\widehat{C}_{tot}^{k,i} x_i^k \le \frac{C}{K}, \forall i \in \mathbb{I}^k \tag{7b}$$

$$\gamma_i^k x_i^k \ge \gamma_0, \forall i \in \mathbb{I}^k \tag{7c}$$

$$x_i^k \in \{0,1\}, \forall i \in \mathbb{I}^k \tag{7d}$$

Constraint (7a) indicates that in round K, the central server selects at least M and at most R participating clients, where $R = \min\{2M, \frac{M+N}{2}\}$. Since clients may drop during training, more than M clients are selected before each round, and the top M training models are selected for the current round's aggregation. Note that constraint (7b) ensures that the estimated completion time does not exceed the average one-round maximum completion time according to the estimation result of **P1**. Constraint (7c) filters out mismatched clients by an upper bound γ_0.

[2]This method sketches each client's dataset by a low-dimensional vector based on JL-transformation [29] and protects the privacy of each sample using a random response mechanism. The high efficiency and low computation cost of this method has been proven in [26].

[3]Same as **P1**, two items in the objective function are normalized.

Challenges. Notice that **P2** is a 0-1 quadratic programming (QP). The heaviest k-subgraph problem (HSP) which is NP-hard [30] can be reduced to **P2** by ignoring constraints (7b) and (7c).

IV. Design of *Hca*

A. Design Overview

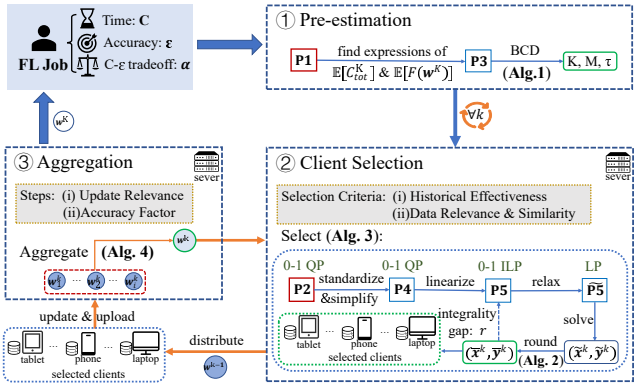

Fig. 1: Architecture of *Hca*.

As shown in Fig. 1, *Hca* consists of three stages.

i. Pre-estimate stage. To meet the target job's weighted time and accuracy demand, we first rewrite **P1** to a complex non-convex optimization program **P3** expressed by three decision variables K, M and τ. Then we solve **P3** and obtain the value of K, M and τ by Alg. 1.

ii. Client selection stage. After determining the control parameters, we design a multi-criteria client selection scheme in each training round k. To tackle **P2**, we first standardize and simplify it to **P4**. Then we perform a reasonable linearization, and convert **P4** to **P5**, in which a 0-1 variable y_{ij}^k is proposed to replace the quadratic term $x_i^k x_j^k$. For the linearized problem **P5**, we first relax its integral constraints to get the fractional solution, and then carefully design a theoretically guaranteed rounding algorithm Alg. 2 to round the fractional solution to the desired integer solution. We integrate the series of methods into Alg. 3 as the client selection scheme.

iii. Model aggregation stage. We make some improvements to the FL model aggregation. The aggregation first selects top M most relevant local updates with current global update. Then The accuracy factor q_i^k is added to the weight factor for the aggregation calculation. These two steps are included in Alg. 4 as the FL aggregation algorithm.

We next introduce three stages in Sec. IV-B, Sec. IV-C and Sec. IV-D respectively.

B. Solving Pre-estimation Problem

The process of solving **P1** has[4] two steps:

i. Explore the specific mathematical expression of $\mathbb{E}[C_{tot}^K]$ and $\mathbb{E}[F(\boldsymbol{w}^K)]$ in **P1**, and rewrite **P1** to a non-convex

[4]**Due to the space limitation, the details of Sec. IV-B, the proofs for all Lemmas and Theorems can be found in the technical report [31].**

optimization program **P3** expressed by three decision variables K, M and τ.

ii. Analyze the properties of **P3** and give the value of K, M and τ in Alg. 1.

C. Solving One-Round Client Selection Problem

Main Idea. The main idea is three-fold. The first step is the preprocess step which standardizes and simplifies **P2**. Second, we linearize the quadratic problem into a 0-1 integer linear programming (ILP) by importing an auxiliary variable and some constraints accordingly. Third, we relax the ILP into an LP, and then adopt a carefully designed rounding method to get the fractional solution. Finally, we obtain the integer solution of **P2**.

Standardize and Simplify. Since we leverage cosine similarity to calculate $s^k(i,j)$, we have $s^k(i,j) \in [-1,1]$. For the computation convenience, we define $s^{'k}(i,j) = 1 - s^k(i,j)$. We first filter out the clients that violate constraints (7b) and (7c) to get a qualified client set, denoted as $\mathbb{I}^{'k}$. We reindex the available clients in $\mathbb{I}^{'k}$ and denote the largest index as N_1. This process is simple and fast in practice. After performing the above two step, **P2** is simplified to **P4**:

$$\textbf{P4}: \quad \max_{\mathbf{x}^k} \quad \sum_{i=1}^{N_1} U_i^k x_i^k + \sum_{1\le i<j\le N_1} s^{'k}(i,j) x_i^k x_j^k \tag{8}$$

$$\text{s.t.} \quad M \le \sum_{i=1}^{N_1} x_i^k \le R, \tag{8b}$$

$$x_i^k \in \{0,1\}, \forall i \in \mathbb{I}^{'k}. \tag{8c}$$

Linearization. **P4** is still intractable because of the quadratic objective function and the integral constraints. We leverage a linearization technique which replaces the quadratic term $x_i^k x_j^k$ by a new variable y_{ij}^k and adds some constraints to express the relationship between y_{ij}^k and x_i^k. The linearization of **P4** is:

$$\textbf{P5}: \quad \max_{\mathbf{x}^k,\mathbf{y}^k} \quad \sum_{i=1}^{N_1} U_i^k x_i^k + \sum_{1\le i<j\le N_1} s^{'k}(i,j) y_{ij}^k \tag{9}$$

$$\text{s.t.} \quad M \le \sum_{i=1}^{N_1} x_i^k \le R, \tag{9a}$$

$$y_{ij}^k \le x_i^k, \forall 1 \le i < j \le N_1, \tag{9b}$$

$$y_{ij}^k \le x_j^k, \forall 1 \le i < j \le N_1, \tag{9c}$$

$$y_{ij}^k \ge 0, \forall 1 \le i < j \le N_1, \tag{9d}$$

$$x_i^k \in \{0,1\}, \forall i \in \mathbb{I}^{'k}. \tag{9e}$$

It's obvious that **P4** and **P5** have the same optimal solution.

Relaxation and Rounding. Next we relax constraints (9d) and (9e) to $y_{ij}^k \in [0,1]$ and $x_i^k \in [0,1]$, respectively. The relaxed **P5** is denoted by $\tilde{\textbf{P5}}$. $\tilde{\textbf{P5}}$ can be solved easily by an LP solver. Let $\tilde{\boldsymbol{x}}^{\boldsymbol{k}}, \tilde{\boldsymbol{y}}^{\boldsymbol{k}}$ denote the optimal solution of $\tilde{\textbf{P5}}$, where $\tilde{\boldsymbol{x}}^{\boldsymbol{k}} = \{x_i^k | i \in \mathbb{I}^{'k}\}$, and $\tilde{\boldsymbol{y}}^{\boldsymbol{k}} = \{y_{ij}^k | 1 \le i < j \le N_1\}$. Next we adopt a randomized pairwise rounding method to round $(\tilde{\boldsymbol{x}}^{\boldsymbol{k}}, \tilde{\boldsymbol{y}}^{\boldsymbol{k}})$ into an integral solution $(\overline{\boldsymbol{x}}^{\boldsymbol{k}}, \overline{\boldsymbol{y}}^{\boldsymbol{k}})$.

Rounding Algorithm. Since each $\overline{y}_{ij}^k$ depends on $\overline{\boldsymbol{x}}^{\boldsymbol{k}}$, we have to consider how to round $\tilde{\boldsymbol{x}}^{\boldsymbol{k}}$ first. A good rounding algorithm should satisfy all constraints, and make the integrality gap as small as possible. Our rounding algorithm is shown in Alg. 2. Alg. 2 chooses a pair of fractions $\tilde{x}_{i_1}^k$ and $\tilde{x}_{i_2}^k$ for rounding in each iteration in the loop of line 3-17, until a single fraction is left. This pair-by-pair manner can guarantee that: i) either $\tilde{x}_{i_1}^k$ or $\tilde{x}_{i_2}^k$ or both are converted to an integer after one round iteration; ii) in this process, $\tilde{x}_{i_1}^k$ and $\tilde{x}_{i_2}^k$ compensate each other, which ensures $\tilde{x}_{i_1}^k + \tilde{x}_{i_2}^k$ keeps unchanged, no matter line 7 or line 8 is executed, so that the solution after the whole rounding still satisfies the constraints; and iii) the expectation of each x_i^k remains the same before and after rounding.

Lemma 1. *The integral solution $(\overline{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{k}}, \overline{\boldsymbol{y}}_{\boldsymbol{ij}}^{\boldsymbol{k}})$ are feasible to **P5**.*

We analyze the integrality gap incurred by Alg. 2, and obtain the following lemma.

Lemma 2. *The integrality gap incurred by Alg. 2 is r,* i.e., *the objective value obtained by Alg. 2 is no less than the objective value of the constant r times the optimum solution of $\tilde{\textbf{P5}}$.*

$$\mathbb{E}[\tilde{P5}(\{(\overline{\boldsymbol{x}}^{\boldsymbol{k}}, \overline{\boldsymbol{y}}^{\boldsymbol{k}}), \forall k\})] \ge r\tilde{P5}(\{\tilde{\boldsymbol{x}}^{\boldsymbol{k}}, \tilde{\boldsymbol{y}}^{\boldsymbol{k}}), \forall k\}) \tag{10}$$

where

$$r = \frac{\min\{s^{'k}(i,j)\}}{\max\{s^{'k}(i,j)\}} \frac{M-1}{N_1}. \tag{11}$$

Algorithm 2 Randomized Pairwise Rounding, $\forall k$

Input: $(\tilde{\boldsymbol{x}}^{\boldsymbol{k}}, \tilde{\boldsymbol{y}}^{\boldsymbol{k}})$
Output: $(\overline{\boldsymbol{x}}^{\boldsymbol{k}}, \overline{\boldsymbol{y}}^{\boldsymbol{k}})$

1: $\theta_i \triangleq \tilde{x}_i^k, \forall i$;
2: $\mathbb{I}^{''k} \triangleq \mathbb{I}^{'k} \backslash \{i | \theta_i \in \{0,1\}\}$;
3: **while** $|\mathbb{I}^{''k}| > 1$ **do**
4: Select $i_1, i_2 \in \mathbb{I}^{''k}$, where $i_1 \ne i_2$;
5: $\omega_1 \triangleq \min\{1-\theta_{i_1}, \theta_{i_2}\}$;
6: $\omega_2 \triangleq \min\{\theta_{i_1}, 1-\theta_{i_2}\}$;
7: With the probability $\frac{\omega_2}{\omega_1+\omega_2}$, Set $\theta_{i_1}^{'} = \theta_{i_1} + \omega_1$, $\theta_{i_2}^{'} = \theta_{i_2} - \omega_1$;
8: With the probability $\frac{\omega_1}{\omega_1+\omega_2}$, Set $\theta_{i_1}^{'} = \theta_{i_1} - \omega_2$, $\theta_{i_2}^{'} = \theta_{i_2} + \omega_2$;
9: **if** $\theta_{i_1}^{'} \in \{0,1\}$ **then**
10: Set $\overline{x}_{i_1}^k = \theta_{i_1}^{'}$, $\mathbb{I}^{''k} = \mathbb{I}^{''k} \backslash \{i_1\}$;
11: **else** Set $\theta_{i_1} = \theta_{i_1}^{'}$;
12: **end if**
13: **if** $\theta_{i_2}^{'} \in \{0,1\}$ **then**
14: Set $\overline{x}_{i_2}^k = \theta_{i_2}^{'}$, $\mathbb{I}^{''k} = \mathbb{I}^{''k} \backslash \{i_2\}$;
15: **else** Set $\theta_{i_2} = \theta_{i_2}^{'}$;
16: **end if**
17: **end while**
18: **if** $|\mathbb{I}^{''k}| = 1$ **then**
19: Set $\overline{x}_i^k = 1$ for the only $i \in \mathbb{I}^{''k}$;
20: **end if**
21: **for** $1 \le i < j \le N_1$ **do**
22: $\overline{y}_{ij}^k = \overline{x}_i^k \overline{x}_j^k$;
23: **end for**
24: Return $(\overline{\boldsymbol{x}}^{\boldsymbol{k}}, \overline{\boldsymbol{y}}^{\boldsymbol{k}})$.

Selection Framework. The client selection framework is presented in Alg. 3. Lines 1-6 filter out the clients that violate the time constraint (7b) and the relevance constraint (7c). Line 7 obtain the fractional solution, and line 8 calls the subroutine Alg. 2 to get the final selection decision.

D. Model Aggregation

Based on the classic *FedAvg*, we make some improvements on model aggregation, including: i) before aggregation, selects M clients with the best local models that are the most relevant with the current global model $\boldsymbol{w}^{k-1}$; and ii) during aggregation, adds the accuracy factor q_i^t to the aggregation weight. For the measurement of relevance, we utilize the method of literature [21]. Given the client i 's local model $\boldsymbol{w}_i^k = (w_{i1}^k, w_{i2}^k, \cdots, w_{id}^k)$, where d is the dimension of the model, the estimated relevance between the local model and the global model is:

$$\boldsymbol{e}(\boldsymbol{w}_i^k, \boldsymbol{w}^{k-1}) = \frac{1}{d}\sum_{j=1}^{d} I(sgn(w_{ij}^k) = sgn(w_j^k)). \tag{12}$$

As shown in Alg. 4, after receiving all the local models $\boldsymbol{w}_i^k$, the server first filters out those irrelevant local models following Eq. (12) in lines 1-4, and then conducts the accuracy-aware weighted update in lines 5-7.

Algorithm 3 Selection Algorithm, $\forall k$

Input: $C, K, \gamma_0, \mathbb{I}^k$
Output: $\{x_i^k | i \in \mathbb{I}^k\}$

1: **for** each $i \in \mathbb{I}^k$ (in parallel) **do**
2: Compute $\widehat{C}_{tot}^{k,i}$ by Eq. (4), and γ_i^k by Eq. (5);
3: **if** $\widehat{C}_{tot}^{k,i} > \frac{C}{K}$ **or** $\gamma_i^k < \gamma_0$ **then**
4: Set $x_i^k = 0, \mathbb{I}^{'k} = \mathbb{I}^k \backslash \{i\}$;
5: **end if**
6: **end for**
7: Solve **P5** to obtain its fractional solution $(\tilde{\boldsymbol{x}}^{\boldsymbol{k}}, \tilde{\boldsymbol{y}}^{\boldsymbol{k}})$;
8: Invoke Alg. 2 to round $(\tilde{\boldsymbol{x}}^{\boldsymbol{k}}, \tilde{\boldsymbol{y}}^{\boldsymbol{k}})$ to $(\overline{\boldsymbol{x}}^{\boldsymbol{k}}, \overline{\boldsymbol{y}}^{\boldsymbol{k}})$;
9: Return $\{x_i^k | i \in \mathbb{I}^k\}$;

Algorithm 4 FL Aggregation, $\forall k$

Input: Received local models $\boldsymbol{w}_i^k, i \in \mathcal{I}^k$
Output: Global model $\boldsymbol{w}^k$

1: **for** $i \in \mathcal{I}^k$ (in parallel) **do**
2: compute the update's relevance $\boldsymbol{e}(\boldsymbol{w}_i^k, \boldsymbol{w}^{k-1})$ following Eq. (12);
3: **end for**
4: Sort clients according to $\boldsymbol{e}(\boldsymbol{w}_i^k, \boldsymbol{w}^{k-1})$ and select the top M clients, set the set of their indices as $\mathcal{I}^k$;
5: **for** $i \in \mathcal{I}^k$ (in parallel) **do**
6: $\boldsymbol{w}^k = \frac{\sum_{i \in \mathbb{I}^k} q_i^k |D_i^k| \boldsymbol{w}_i^k}{\sum_{i \in \mathbb{I}^k} q_i^k |D_i^r|}$;
7: **end for**
8: Return $\boldsymbol{w}^k$;

Theorem 1. Hca *can meet the job completion time and model accuracy requirements of the FL job.*

V. Performance Evaluation

A. Experimental Setup

FL Platform and Model. The testbed experiments are carried out on the FL platform FAVOR [14]. We create 100 clients and each client with a PyTorch model is simulated as a thread running synchronously in a global iteration. We conduct experiments to train a classic CNN model with 5×5 convolutional layers. The output channels of the first and second layers are 16 and 32 respectively, and each layer has a 2×2 max pooling. The model is trained based on three datasets: MNIST, Fashion-MNIST (FMNIST), and CIFAR-10. MNIST is a dataset of handwritten digits and FMNIST is a dataset of Zalando's fashion article images, both of which have a training set of 60,000 examples and a test set of 10,000 examples. CIFAR-10 dataset consists of 50,000 training images and 10,000 test images. The images from each dataset are divided into ten categories. The training task is multi-classification, and we use cross-entropy as the loss function. By default, we set $\alpha = 1$.

Training Data. To simulate the non-IID distribution of clients' datasets in the real scenario, we select an amount of data from one main dataset and randomly choose other data from the other two datasets. In addition, the amount of data from different categories is random. We call the three processed datasets non-IID MNIST, non-IID FMNIST, and non-IID CIFAR-10 respectively. Considering the realistic scenario that clients are continuously collecting data and updating its dataset, we randomly distribute a small part of data to each client after each round of training. Before each communication round, we calculate the relevance γ_i^k by Eq. (5), and the similarity by Eq. (6). We use cross-entropy to measure the loss. The values of job completion time and loss are mapped to the range of 0-1 for normalized processing, according to the actual observed time and loss range.

B. Performance of Pre-estimation

TABLE I: The value of three parameters for different α

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	1
K	2	4	5	7	10	21	23	27	33	60
M	12	12	12	5	12	12	12	12	12	10
τ	3	3	3	7	3	3	3	3	3	10

We first verify the efficiency of the first stage: the pre-estimation of three parameters K, M and τ. We train a CNN model by adopting *Hca* on non-IID MNIST. We vary the value of α and obtain the value of K, M and τ, and continue the training by involving the second and third stage in *Hca*. Table I shows the value of three parameters for different α.

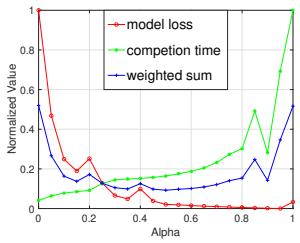

Fig. 2: Normalized completion time, model loss and their weighted sum.

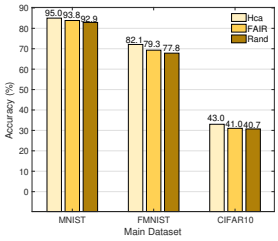

Fig. 3: Test accuracy with different selection strategies on different non-IID datasets.

Impact of α**.** We verify the effect of α on job requirements. Specifically, we compare the model loss, the completion time and their weighted sum with different α. From Fig. 2, we can observe that when the value of α increases, the completion time (green line) decreases, and the model loss (blue line)

increases, which indicates that α has the trade-off effect between the time and accuracy requirement.

C. Performance of Client Selection

Benchmarks. We compare our client selection scheme with two commonly adopted client selection methods: Random and FAIR [15]. To complete the training, we use the first and the third stage in *Hca* to obtain three parameters and perform model aggregation.

- Random: which selects R clients randomly.
- FAIR [15]: which greedily selects R clients with the best training quality (the ones with the largest global loss reduction).

Results. We fix $K = 60, M = 10, \tau = 10$, which are obtained from pre-estimation by setting $\alpha = 1$. We conduct FL by three selection methods on different datasets for 20 times, and take the average as the finial output. Fig. 3 shows the comparison of their average accuracies on non-IID MNIST, non-IID FMNIST and non-IID CIFAR-10. We can observe that for all datasets, the performance of our selection strategy is the best. The random selection has the worst performance.

D. Performance of Model Aggregation

Benchmarks. We compare our model aggregation method with FedAvg [3] and CMFL [21].

- FedAvg [3]: $\boldsymbol{w}^k$ is updated without integrating q_i^k.
- CMFL [21]: which filters out the irrelevant local models and averages local model parameters by FedAvg's method.

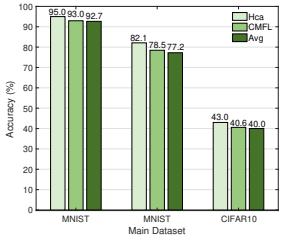

Fig. 4: Test accuracy with different aggregation methods on different non-IID datasets.

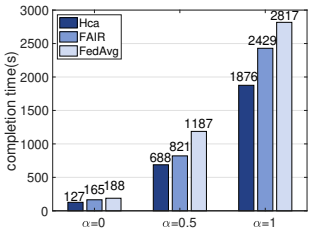

Fig. 5: Completion time of different frameworks.

Results. We compare three aggregation methods by implementing the first and the second stage in *Hca*, and conduct training on a CNN model for 20 times. Fig. 4 shows the training accuracies of different aggregation approaches. From Fig. 4 we can observe that on MNIST, FMNIST and CIFAR-10, FedAvg performs the worst, whose average accuracy is 2.5%, 6.3% and 7.5% less than that of *Hca*. We can conclude that our aggregation approach is superior to the other two approaches.

E. Performance of Hca

Benchmarks. We finally evaluate the overall performance of *Hca*. We compare *Hca* with two commonly used FL frameworks: FedAvg [3] and FAIR [15]. In addition to CNN multi-classification task, we also study the performance of logistic regression task performed on the FashionMNIST dataset.

- FedAvg [3]: which randomly selects a fraction of clients for training and aggregates models by FedAvg algorithm with random K, M and τ.
- FAIR [15]: which greedily selects R clients with the best training quality (the ones with the largest global loss reduction) and integrates the quality into model aggregation with random K, M and τ.

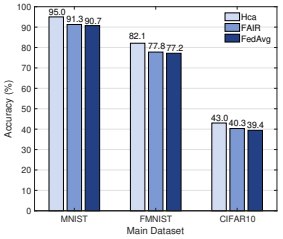

Fig. 6: Test accuracy with different FL frameworks on different non-IID datasets.

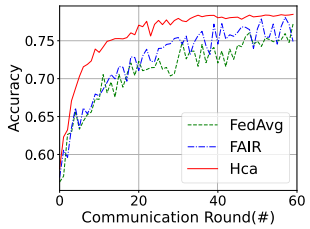

Fig. 7: Test accuracy for logistic regression task on non-IID FMNIST.

Results. To evaluate the performance of *Hca* on job completion time, we vary the value of α and plot the completion time of three frameworks in Fig. 5. We can see that *Hca* has the shortest job completion time and can almost double FedAvg's training speed. When α grows, two benchmarks need more time to achieve the desired accuracy. Fig. 6 shows the training accuracy for CNN multi-classification task. Fig. 7 shows the model accuracy in 60 rounds for logistic regression task of the three models, in which we can observe that the model obtained by our proposed training framework for logistic regression task is significantly better than the other two FL frameworks. Fig. 8-Fig. 10 further illustrate the dynamic change of accuracy in 60 rounds. We can observe that although the accuracy gap between FAIR and *Hca* shown in Fig. 6 is not large, the convergence rate of *Hca* is much faster than the other two frameworks, and the training process is more stable. For instance, from Fig 10, *Hca* converges to a stable accuracy in about 15 rounds. The model loss is demonstrated in Fig. 11-Fig. 13. We can conclude that *Hca* has the best training performance in FL for different kinds of tasks.

VI. Conclusion

In this paper, we propose *Hca*, a heterogeneous FL framework for balancing job completion time and model accuracy. Different from existing literature, our framework consists of three stages. First, we determine the number of training rounds, the number of iterations and the number of participating clients in each round, to satisfy FL job's requirement on job completion time and model accuracy. Second, a multi-criteria client selection framework is involved in selecting the most efficient clients for the FL job. At last, we tailor an improved model aggregation algorithm to further optimize the quality of the FL model. The extensive results from testbed experiments based on real-world data verify that *Hca* achieves near-optimal performance in both model accuracy and job completion time, compared with existing FL frameworks.

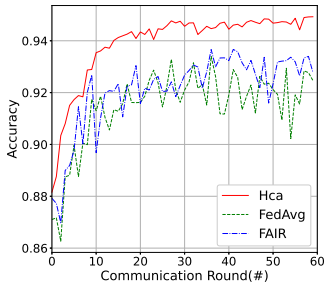

Fig. 8: Training accuracy on non-IID MNIST.

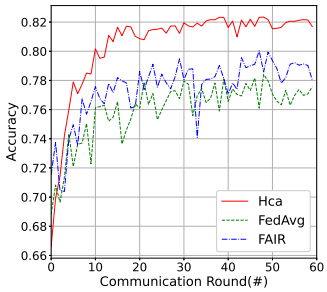

Fig. 9: Training accuracy on non-IID FMNIST.

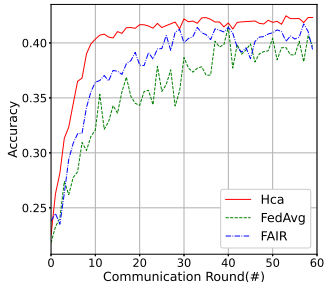

Fig. 10: Training accuracy on non-IID CIFAR-10.

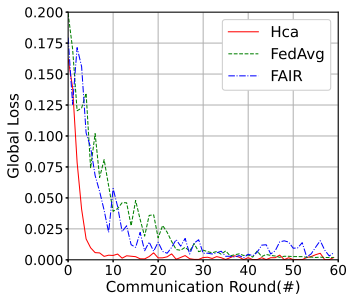

Fig. 11: Global loss on non-IID MNIST

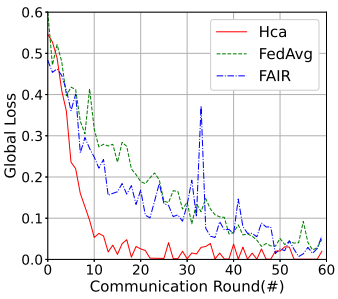

Fig. 12: Global loss on non-IID FMNIST

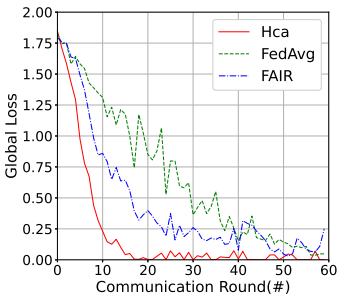

Fig. 13: Global loss on non-IID CIFAR-10

REFERENCES

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[2] 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 Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of PMLR AISTATS*, 2017.

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

[5] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[6] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2020.

[7] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," *arXiv preprint arXiv:2010.01264*, 2020.

[8] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. of IEEE INFOCOM*, 2021.

[9] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[10] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[11] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2020.

[12] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," *arXiv preprint arXiv:2101.00787*, 2021.

[13] Y. Xu, *Block coordinate descent for regularized multi-convex optimization*. Rice University, 2013.

[14] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. of IEEE INFOCOM*, 2020.

[15] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, and Y. Zhang, "Fair: Quality-aware federated learning with precise user incentive and model aggregation," in *Proc. of IEEE INFOCOM*, 2021.

[16] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. of MIT Press NIPS*, 2017.

[17] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. of ICLR*, 2019.

[18] S. U. Stich, "Local sgd converges fast and communicates little," 2019.

[19] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *Proc. of IEEE IJCNN*, 2020.

[20] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *Proc. of IEEE INFOCOM*, 2021.

[21] L. Wang, W. Wang, and B. Li, "Cmfl: Mitigating communication overhead for federated learning," in *Proc. of IEEE ICDCS*, 2019.

[22] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *Proc. of IEEE ICASSP*, 2019.

[23] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. of IEEE ICC*, 2019.

[24] M. Ribero and H. Vikalo, "Communication-efficient federated learning via optimal client sampling," 2020.

[25] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," 2020.

[26] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li, "Sample-level data selection for federated learning," in *Proc. of IEEE INFOCOM*, 2021.

[27] G. Damaskinos, R. Guerraoui, A.-M. Kermarrec, V. Nitu, R. Patra, and F. Taïani, "Fleet: Online federated learning via staleness awareness and performance prediction," in *Proc. of ACM Middleware*, 2020.

[28] B. Pinkas, T. Schneider, and M. Zohner, "Scalable private set intersection based on ot extension," Cryptology ePrint Archive, Report 2016/930, 2016, https://ia.cr/2016/930.

[29] C. Biswas, D. Ganguly, D. Roy, and U. Bhattacharya, "Privacy preserving approximate k-means clustering," in *Proc. of ACM CIKM*, 2019.

[30] D. G. Corneil and Y. Perl, "Clustering and domination in perfect graphs," *Discrete Applied Mathematics*, vol. 9, no. 1, pp. 27–39, 1984.

[31] "Hca-long-version.pdf." [Online]. Available: https://www.jianguoyun.com/p/DUhgvHUQlbPKChiz2r8EIAA