How Valuable Is Your Data? Optimizing Client Recruitment in Federated Learning

Yichen Ruan
Carnegie Mellon University

Xiaoxi Zhang
Sun Yat-Sen University

Carlee Joe-Wong
Carnegie Mellon University

***Abstract*—Federated learning allows distributed clients to train a shared machine learning model while preserving user privacy. In this framework, an operator recruits user devices (i.e., clients) to occasionally perform local iterations of the learning algorithm on their data. We propose the first work to theoretically analyze the resulting performance tradeoffs in deciding which clients to recruit for federated learning, complementing other works on the selection of recruited clients in each iteration. Specifically, we define and optimize the tradeoffs between both accuracy (training and testing) and efficiency (completion time and cost) metrics. We provide efficient solutions to this NP-Hard optimization problem, and verify the value of client recruitment in experiments on synthetic and real-world data. The results of this work can serve as guidelines for the real-world deployment of federated learning and an initial investigation of the client recruitment problem.**

I. Introduction

Emerging machine learning techniques have achieved great success in creating value from big data. Much of this data is generated by increasingly pervasive mobile devices. These devices, e.g., smart phones, are generally equipped with powerful sensors and considerable storage space, making them appealing sources of training data for machine learning. Traditional learning deployments assume all data is stored in a central location (e.g., in a single datacenter) where it can be accessed as needed for training a model. User data is then fully exposed to the operator of the model training. In practice, however, the private nature and the potential volume of user data from mobile devices obstruct its centralized storage and processing, making it hard to utilize these isolated knowledge bases.

The recently proposed federated learning framework allows users to contribute the power of their data to train machine learning models without sharing any raw records. However, ensuring the good performance of federated learning requires overcoming additional challenges that do not present in centralized learning. Specifically, the challenges originate in the heterogeneous distributions of data at different users (statistical challenge), and in the complexity of the edge computing system (system challenge) [9]. Much recent work aims to address them by optimizing the learning algorithm given a set of participating clients, i.e., clients that use their data to contribute model updates to the training process. However, these works neglect a complementary question: Before running the federated learning algorithm, *how should the operator recruit participating clients so as to optimize the performance of its federated learning algorithm?* In this work, we show that a good client recruitment is essential to overcoming federated learning's statistical and system challenges, complementing algorithmic innovations like carefully selecting or scheduling model updates from a given set of clients.

Client recruitment formalizes the relationship between the two market players in typical **commercial applications** of federated learning: the operators and the users. Operators are typically companies who hope to create or improve their AI products utilizing their users' data. For example, Google has utilized data from Android users to train a query suggestion model for their keyboard application [13]. Federated learning operators are responsible for setting up a coordinator that collects iterative updates from the participating clients. However, most federated learning algorithms require upfront commitments from users to compute local updates, which may consume limited battery, and send them to the coordinator, which may reveal private information, to the training on demand. Such upfront commitments are generally required to ensure convergence of the training algorithm [5]. To compensate for these commitments, recruited users may need incentives from the operator to participate in the training, as proposed in [2] to compensate privacy losses. Such compensation, however, introduces a new challenge not commonly considered in federated learning: limiting the recruitment cost.

We define **client recruitment** as the preliminary step of federated learning, in which the coordinator determines the set of candidate clients with which it will train a model. When the recruitment is finalized, we will have determined the quality and quantity of the training data, the number and types of local devices, and the associated cost of compensating users. A good client recruitment is fundamental to the successful execution of federated learning and complements the more commonly considered *client selection* [7], in which the coordinator chooses which of the recruited clients will be asked to provide updates in each training iteration. Since federated learning requires upfront client commitments as discussed, recruiting clients is necessary to ensure the success of subsequent client selection. Indeed, careful recruitment will reduce the number of clients required to make training commitments (by almost 5x in our experiments), improving federated learning's overall efficiency and impact on user privacy. However, client recruitment raises **new challenges** compared to client selection: unlike client selection algorithms that utilize information revealed during the training, we must base recruitment decisions on information known before training begins, which requires more detailed statistical analysis of the anticipated learning accuracy. Com-

ISBN 978-3-903176-37-9 © 2021 IFIP

plicating this problem further, client recruitment additionally decides the model's *generalizability* and *representativeness*, which client selection cannot control.

The **contributions of this paper** are: 1) We construct a comprehensive system model to *quantify the quality measures of federated learning*, including not only the training loss, but also the model's generalizability, the reliability and completion time of training, and the operating expense (Section IV); 2) We formulate an *optimization framework* to capture the complex tradeoffs in client recruitment (Section V); 3) We introduce *approximation methods for our quality metrics* that can be computed in practice even when clients' data distributions are unknown (Section V-A); 4) We exploit the structure of this NP-hard optimization problem to provide a *provably optimal, tractable* solution (Section VI); and finally, 5) We demonstrate our work's *practical feasibility* by learning models with higher accuracy and fewer clients compared to heuristic recruitment methods, on synthetic and real datasets (Section VII).

II. Related Works

Federated learning was first introduced with the FedAvg algorithm [5]. Many experiments since then have shown that *FedAvg* can indeed produce accurate machine learning models (e.g. [13]). The convergence of *FedAvg* has been proved for strongly convex and Lipschitz continuous objectives [3], [11]. These works also prove that the non-IID distribution of local datasets can greatly obstruct convergence.

Client selection, which studies the scheduling of client participation in each global round of federated learning, complements client recruitment. E.g., [7] proposes an adaptive selection algorithm to maximize the number of participating clients in each round while subject to resource restrictions. Similar topics are discussed in [12], which assumes clients follow a specific scheduling policy for global aggregations. Client recruitment complements these selection policies by ensuring that a suitable group of clients is available to be selected in the first place: without a good recruitment, generic client selection algorithms cannot guarantee the convergence of federated learning to a globally optimal solution [1]. Client selection also requires all clients to stay active and ready to be summoned anytime, even if they are not always selected, which without a good recruitment process is unrealistic due to the system challenges to be discussed in Section III.

In practical settings, client recruitment can thus limit the cost of federated learning since it pre-excludes disqualified clients before any training steps are taken or incentives offered, while client selection still requires the operator to pay recruited clients, who have committed to being available for training even if they are never ultimately selected. The client recruitment method discussed in this paper is independent of the remaining training details, and can thus be coupled with any learning algorithm and selection strategies.

III. Federated Learning Background

Federated learning trains a single model by attempting to minimize the model's empirical risk, i.e., the training loss, over data from multiple clients. Let $\mathcal{U}$ denote the set of K candidate clients, each with a dataset $\mathcal{D}_k = \{(u_i, v_i)\}_i$, where u_i, v_i are a feature vector and the corresponding label. Let $l(w; u, v)$ be a loss function with weight vector w and a data record (u, v). The local empirical risk of client k is:

$$\tilde{R}_k(w; \mathcal{D}_k) = \frac{1}{|\mathcal{D}_k|} \sum\nolimits_i l(w; u_i, v_i) \tag{1}$$

The ultimate goal of training is thus to find w that minimizes the empirical risk over the global dataset $\mathcal{D}_x = \cup_k \mathcal{D}_k$:

$$\min_w \tilde{R}_x(w; \mathcal{D}_x) = \sum\nolimits_{k=1}^{K} \frac{n_k}{n_x} \tilde{R}_k(w; \mathcal{D}_k) \tag{2}$$

Here $n_k = |\mathcal{D}_k|$, and $n_x = \sum_k n_k$, representing the total samples of all recruited clients determined by the recruitment decision x, which we formally define in Section IV. To minimize $\tilde{R}_x$, the distributed stochastic gradient descent (SGD) paradigm is utilized. A central coordinator maintains a global weight w, and each client maintains a local weight w_k. The training repeats the following 3 steps for $t = 1, ..., T$.

1) *Synchronization*: The coordinator broadcasts the latest global weight $w^{\tau t}$ to the clients through the network. Clients then update their local weights $w_k^{\tau t}$ with $w^{\tau t}$.

2) *Local optimization*: Each client k runs SGD in parallel for τ steps to minimize its local risk $\tilde{R}_k$, getting $w_k^{(t+1)\tau}$.

3) *Aggregation*: The coordinator aggregates clients' local weights $\{w_k^{(t+1)\tau}\}_k$ by setting the next global weight as their weighted average: $w^{(t+1)\tau} = \sum_k \frac{n_k}{n_x} w_k^{(t+1)\tau}$.

Client selection chooses which clients perform these steps in each iteration, while client recruitment chooses the set of eligible clients $\mathcal{U}_x$. This paper assumes all algorithm parameters are given, including τ and T. Our results are thus robust to different algorithm settings and complement optimizations of these parameters. The challenges of federated learning are:

Statistical Challenge: Non-IID datasets. Unlike in the data center where data is assumed to be identically and independently distributed among workers, clients' data distributions in federated learning may well be non-IID. Thus, we suppose that the samples in every local dataset $\mathcal{D}_k$ are independently drawn from a distinct distribution $\mathcal{P}_k$. Non-IID data can greatly decelerate the training. The training loss can be bounded by a monotonically increasing function of the average difference in local and global distributions $|\tilde{P}_k - \tilde{P}_x|$ [11]:

$$\text{training loss} \propto \sum\nolimits_k \frac{n_k}{n_x} |\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}_x| \tag{3}$$

System Challenge: Stragglers and failures. User devices have relatively constrained computation resources (e.g. CPU, memory), which furthermore must be shared among many apps. These make stragglers (devices that take a long time to run local iterations) and even occasional device failures, e.g., due to lost power or network connectivity, more likely to appear in federated learning. These must be treated carefully to ensure the success of the learning algorithm.

Due to space limit, in this paper our analysis is largely based on the vanilla FedAvg algorithm. Incorporating client recruitment with other variant algorithms such as asynchronous

ISBN 978-3-903176-37-9 © 2021 IFIP

federated learning, model personalization, and dynamic client selection will be an interesting future research direction.

IV. System Modeling

Formally speaking, given a list of candidate clients $\mathcal{U} = \{u_k\}_k$, the goal of client recruitment is to choose the optimal subset of clients $\mathcal{U}_x \subseteq \mathcal{U}$ that will run the learning algorithm to optimize the overall performance of federated learning. In this section, we first model the local/global/population data distributions, which we then use to propose formal performance metrics for both the learning accuracy and the efficiency.

A. Data Distributions

Since federated learning trains one model for all clients, we assume all data is generated in an IID manner from a *population distribution* $\mathcal{P}$. On the other hand, each client's data individually forms a *local distribution* $\mathcal{P}_k$, which can differ from other local distributions and from $\mathcal{P}$. When we compare two local datasets, we assume the data is not identically distributed between them. In contrast, when we discuss the union of all local datasets, we treat each sample as IID distributed in $\mathcal{P}$. E.g., suppose we are training a model to predict temperature from features such as the amount of sunlight and rainfall. Then $\mathcal{P}$ represents the joint distribution of world-wide temperature with these features. Since a client can only collect temperature data in a small region, its local distribution $\mathcal{P}_k$ only reflects the regional climate characteristics. As a result, clients in different regions possess divergent local distributions. In the meanwhile, all these data points are essentially generated within the same Earth climate system. Thus when forged together, they do follow the world-wide distribution $\mathcal{P}$ in an IID manner.

A local dataset $\mathcal{D}_k$ may not describe its local distribution $\mathcal{P}_k$ well if insufficient data points were collected. In practice, the operator estimates $\mathcal{P}_k$ by indirectly evaluating its *empirical distribution* $\tilde{\mathcal{P}}_k$, which converges to the real $\mathcal{P}_k$ when $\mathcal{D}_k$ grows larger. Similarly, we define the global dataset $\mathcal{D}_x = \cup_k \mathcal{D}_k$ as the union of all recruited datasets. Data in $\mathcal{D}_x$ forms the *global empirical distribution* $\tilde{\mathcal{P}}_x$. Likewise, $\tilde{\mathcal{P}}_x$ is a weighted average of local empirical distributions: $\tilde{\mathcal{P}}_x = \sum_k \frac{n_k}{n_x} \tilde{\mathcal{P}}_k$. Since all data in $\mathcal{D}_x$ is independently drawn from $\mathcal{P}$, $\tilde{\mathcal{P}}_x$ can be regarded as an empirical estimation of $\mathcal{P}$ when a reasonably large number of clients are recruited. In the climate data example, if we have recruited clients from all climatic zones in the world, the union of this data $\tilde{\mathcal{P}}_x$ becomes a good representative of the whole Earth climate system $\mathcal{P}$.

We suppose the operator can estimate $\mathcal{P}$ with a small benchmark dataset (e.g., as in [10]) $\tilde{D} \sim \mathcal{P}$, and we denote its empirical distribution by $\tilde{P}$. Since $\tilde{D}$ is small in size, it cannot be directly used for training. Instead, this educated guess of the population allows the operator to gauge the clients' quality and representativeness. E.g., the operator may estimate the distribution of rainfall from historical climate data. We show how to estimate data quality with $\tilde{P}$ in Sections V and VII.

B. Performance Metrics

We will consider two categories of performance measures. The first category is the accuracy of the output model for the distribution $\mathcal{P}$, which includes not only the training loss, but also the model's generalizability and its representativeness. The second measures the training efficiency, which includes the time to complete the training, and the cost incurred.

Reduce training loss with high-quality data: A dataset $\mathcal{D}_k$ is considered of good quality if its distribution $\tilde{\mathcal{P}}_k$ resembles the population $\mathcal{P}$. From (3), if $\tilde{\mathcal{P}}_x$ resembles $\mathcal{P}$ (see "representativeness" below), the quality of the local datasets directly determines the training loss. A dataset $\mathcal{D}_k$ with a small distribution divergence $|\tilde{\mathcal{P}}_k - \mathcal{P}|$ yields small training loss.

Reduce generalization error with more data: Given a loss function l and dataset $\mathcal{D}$, the generalization error $|\tilde{R} - R|$ is the divergence between the empirical risk $\tilde{R}(w; \mathcal{D}) = (\sum l(w; u, v))/|\mathcal{D}|$ and the real risk $R(w) = \mathbb{E}_{\mathcal{P}}[l] = \int l(w; u, v) d\mathcal{P}$. While the training loss gauges the model's performance on the training data, the generalization error reflects its accuracy when applied to *new samples drawn from the recruited distributions*. If a client has insufficient data, its local empirical distribution $\tilde{\mathcal{P}}_k$ may poorly approximate $\mathcal{P}_k$, which implies a large generalization error. Existing works generally omit the generalization error as they take the training data as given. For us, however, client recruitment determines the size of the dataset, affecting the generalization error.

Choose for population representativeness: For the trained model to be applicable to *unrecruited datasets*, the recruited clients, when forged together, must be representative of the population $\mathcal{P}$. Indeed, if the clients do not cover portions of the population space, we will perform poorly in those areas. E.g., including polar region data complicates the training of models that predict worldwide temperatures, but failing to do so can degrade the model's performance in this region.

Control the completion time: Federated learning is useless if the training process does not complete in reasonable time. We define the completion time as the expected time for the coordinator to finish all T rounds of aggregations.

Control the cost: Since the size of an individual local dataset is usually small, a typical execution of federated learning may need thousands of recruited clients. The operator should thus make sure the resulting expense is affordable.

V. Problem Formulation

We formulate client recruitment as the following optimization problem: Given a set of candidate clients $\mathcal{U} = \{U_k\}_{k=1}^K$, let $x \in \{0,1\}^K$ be a binary vector denoting the recruitment decision for each client. The operator picks an optimal subset $\mathcal{U}_x = \{U_j | x_j = 1\}$ to minimize an objective function f, subject to a given maximum completion time I_t and cost I_c.

Problem 1. *Client Recruitment*

$$\begin{aligned} \min_{x \in \{0,1\}^K} \quad & f(x) = \gamma_{tl} f_{tl}(x) + \gamma_{ge} f_{ge}(x) + \gamma_{rp} f_{rp}(x) \\ \textit{s.t.} \quad & g_t(x) \leq I_t, g_c(x) \leq I_c \end{aligned}$$

Here f consists of 3 terms that determine the accuracy of the trained model: f_{tl}, f_{ge}, f_{rp}, which respectively upper bound the training loss, the average generalizability, and the representativeness. $f(x)$ determines the goodness of the trained

ISBN 978-3-903176-37-9 © 2021 IFIP

model when applied to existing or future data points generated by both recruited and unrecruited clients. The coefficients $\gamma_{tl}, \gamma_{ge}, \gamma_{rp}$ determine the relative importance of these terms, and g_t, g_c are respectively the completion time and cost.

A. Quantifying Accuracy Metrics

We first consider the training loss. From (3), the training loss is determined by the divergence between local and global empirical distributions $\sum_k \frac{n_k}{n}|\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}_x|$. However, since the global distribution can only be determined after the recruitment process, it is hard to optimize the divergence directly. We thus use the fact that $\tilde{\mathcal{P}}_x$ resembles $\mathcal{P}$ (Lemma 2) to define:

$$f_{tl}(x) = \sum\nolimits_k \frac{x_k n_k}{n_x} |\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}| \tag{4}$$

Sharing this metric preserves user privacy since it does not actually require the individual empirical distributions $\tilde{\mathcal{P}}_k$'s. Instead, the required information from the clients $|\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}|$ only encodes the distance of local distributions to the population. Below we provide tractable methods and formula to approximate f_{tl}, **without the need to know** $\tilde{\mathcal{P}}_k$ **or** $\mathcal{P}_k$. We verify the effectiveness of these methods in Section VII.

- *Counting classes:* Consider a classification problem with L classes, and suppose $\tilde{\mathcal{P}}_k$ and $\tilde{\mathcal{P}}$ have densities $\tilde{p}_k$ and $\tilde{p}$. We can write $|\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}| = \int |\tilde{p}_k(u,v) - \tilde{p}(u,v)|dudv = \sum_{i\in[L]} \int |\tilde{p}_k(u|v)\tilde{p}_k(v=i) - \tilde{p}(u|v)\tilde{p}(v=i)|du$. Assume $\tilde{p}_k(u|v) = \tilde{p}(u|v)$, i.e., local features have the same distribution as the population given the label. Thus, $\int |\tilde{p}_k - \tilde{p}| \propto \sum_{i\in[L]} |\tilde{p}_k(v=i) - \tilde{p}(v=i)|$, where $\tilde{p}$ is known a prior. Denoting by C_i^k the number of data points with label i in $\mathcal{D}_k$, $\tilde{p}_k(v=i) = C_i^k / \sum_{i=1}^L C_i^k$. Thus, the whole $\sum |\tilde{p}_k(v=i) - \tilde{p}(v=i)|$ can be easily computed by simply counting the number of labels each client sees. Estimating $\tilde{P}$ from $\tilde{D}$ entails the same simple counting process.
- *Gaussian graphic model approximation:* For general supervised learning with continuous labels, we can formulate the features and the label as a Gaussian graphic model. A local empirical distribution is then fully specified by the mean and covariance $(\tilde{\mu}_k, \tilde{\Sigma}_k)$. The quality measure then becomes the divergence between two Gaussian distributions, which can be quantified by the Kullback–Leibler divergence: $|\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}| \propto D_{KL}\left(\mathcal{N}(\tilde{\mu}_k, \tilde{\Sigma}_k), \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})\right)$. Estimating $\tilde{P}$ from $\tilde{D}$ entails computing $\tilde{\mu}, \tilde{\Sigma}$ and inferring the graph connectivity (e.g. from the covariance), which is tractable.

Next, we model the average generalizability. Since the training objective of federated learning $\tilde{R}_x$ is an average of local empirical risks as in (2), we similarly quantify the average generalization error of local datasets by $\sum_k \frac{x_k n_k}{n_x}|\tilde{R}_k - R_k|$. To formulate it, we rely on Lemma 1 as follows:

Lemma 1. *There exists a class of convex learning problems (e.g. linear regression), for which we can obtain the following generalization error bound for all clients k:*

$$|\tilde{R}_k - R_k| = O\left(n_k^{-0.5}\right) \tag{5}$$

For example, [8] proves this bound for the linear regression model. A tighter convergence bound taking the form $O(n_k^{-\beta})$ with $\beta > 0.5$ is also possible using more sophisticated statistical tools. For simplification and to accommodate non-convex models that may have looser risk generalization bounds, this paper assumes a relatively big $\beta = 0.5$. However, our analysis can be easily extended to any $\beta < 1$. We thus define:

$$f_{ge}(x) = \sum\nolimits_k \frac{x_k n_k}{n_x} n_k^{-0.5} \tag{6}$$

We then model the representativeness. To make sure the chosen distributions can represent basic characteristics of the population distribution, we seek to minimize the divergence between $\tilde{\mathcal{P}}_x$ and $\mathcal{P}$. As is discussed in Section IV-A where we assume $\tilde{\mathcal{P}}_x$ is an empirical distribution of $\mathcal{P}$, and using the central limit theorem, we have the following uniform bound:

Lemma 2. $\tilde{\mathcal{P}}_x - \mathcal{P}$ *converges in distribution to the Gaussian distribution with 0 mean at the rate of* $O(n_x^{-0.5})$.

Therefore, statistically, when n_x grows larger, $\tilde{\mathcal{P}}_x$ becomes a more accurate representative of $\mathcal{P}$. We thus define:

$$f_{rp}(x) = n_x^{-0.5} \tag{7}$$

Combining (4), (6), and (7), the objective f can be expressed as in (8). Since the coefficient γ_{rp} is a positive constant independent of x, we normalize it to 1. The s_k value here is a weighted sum of the client's quality representation $|\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}|$ and quantity representation $n_k^{-0.5}$. It thus enables us to quantify the quality-quantity tradeoff when choosing user datasets.

$$\begin{aligned} f(x) &= \gamma_{rp}\left(\frac{\sum x_k n_k s_k}{n_x} + n_x^{-0.5}\right) \\ s_k &= \frac{\gamma_{tl}}{\gamma_{rp}}|\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}| + \frac{\gamma_{ge}}{\gamma_{rp}} n_k^{-0.5} \end{aligned} \tag{8}$$

B. Quantifying System Metrics

Now we analyze the completion time. We assume for each round of the training, the coordinator will wait up to a predetermined duration E_0. The global weight will then be calculated based on the weights received before the deadline.

We model the client failure as a Markov chain. An active client crashes with probability q_f, and a failed client recovers with probability q_r. Here $q_r, q_f > 0$. Suppose there are m recruited clients. Letting A^t be the number of active clients at iteration t, which has the following properties.

Proposition 1. A^t *is an ergodic Markov chain. In the steady state, the probability that there are* i *active clients equals*

$$\pi_i \triangleq \mathbb{P}(A^\infty = i) = \frac{\binom{m}{i}(q_r/q_f)^i}{(1 + q_r/q_f)^m} \tag{9}$$

Proof. It can be easily verified that A^t is homogeneous, positive recurrent and aperiodic, thus it's ergodic. To get (9), we use the condition that $\sum_i \pi_i = 1$, and $\pi_i P_{i0} = \pi_0 P_{0i}$ for all i, where P_{ij} is the transition probability. □

Since ergodic Markov chains converge exponentially fast, we only consider the steady state. The probability that no clients fail is thus $(\frac{q_r}{q_f + q_r})^m$. To model the system heterogeneity of clients, we partition clients into N groups according to

ISBN 978-3-903176-37-9 © 2021 IFIP

their devices, network qualities, battery levels etc. Suppose each group has m_z clients for $z = 1, ..., N$, and all the clients inside a group z have the same failure rate q_f^z and recovery rate q_r^z. Since clients are running independently, the probability that all clients in all groups are active is then $\prod_{z=1}^{N}(\frac{q_r^z}{q_f^z+q_r^z})^{m_z}$.

If a client k in group z is active, we model its per-iteration runtime as a random variable $Y_k^z \sim \exp(\lambda^z)$. The expected full iteration runtime when all clients are active is: $\Gamma(m_1, ...m_N) = \mathbb{E}[\min\{\max_z \max_k Y_k^z, E_0\}]$. The completion time is as follows, where m_z depends on the recruitment.

$$\begin{aligned} &g_t(x) = g_t(m_1, ..., m_N) \\ =&T\left(\Gamma\prod_{z=1}^{N}(\frac{q_r^z}{q_f^z+q_r^z})^{m_z} + E_0(1-\prod_{z=1}^{N}(\frac{q_r^z}{q_f^z+q_r^z})^{m_z})\right) \end{aligned} \tag{10}$$

Intuitively, the completion time increases when we recruit more clients. This is summarized in Proposition 2.

Proposition 2. *The completion time $g_t(m_1, ...m_N)$ increases when any $m_z, z = 1, ..., N$ increases.*

Proof. It can be shown $\frac{\partial\Gamma}{\partial m_z} > 0$, thus $\frac{\partial g_t}{\partial m_z} > 0 \ \ \forall m_z$. □

Finally, we consider the cost. The cost depends on specific payment mechanisms (e.g. [2]) adopted. Here we assume a generic case where each client k has an exogenous price c_k:

$$g_c(x) = \sum\nolimits_k^K x_k c_k \le I_c \tag{11}$$

VI. The Optimal Client Recruitment

From Section V, each client $U_k \in \mathcal{U}, k = 1, .., K$ can be characterized by a tuple $(|\tilde{\mathcal{P}}_k - \tilde{\mathcal{P}}|, n_k, \mathcal{Z}(k), c_k)$, representing respectively the distribution divergence, the local dataset size, the group number, and the ask price. Clients in a group z have failure rate q_f^z, recovery rate q_r^z and processing rate λ^z. As discussed above, the client can readily compute this information and send it to the operator without significant privacy loss at the start of the recruitment.

Combining (8), (10), and (11), Problem 1 becomes:

Problem 2. *Client Recruitment*

$$\begin{aligned} \min_{x\in\{0,1\}^K} \quad & f(x) = \frac{1}{n_x}\sum_k x_k n_k s_k + n_x^{-0.5} \\ \textit{s.t.} \quad & g_t(x) = (\Gamma - E_0)\prod_{z=1}^{N}\left(\frac{q_r^z}{q_f^z+q_r^z}\right)^{m_z} + E_0 \le \frac{I_t}{T} \\ & g_c(x) = \sum\nolimits_k^K x_k c_k \le I_c \end{aligned}$$

Proposition 3. *Problem 2 is NP-Hard.*

Proof. Let $I_t = \infty, s_k = 0$, then $\min f \Leftrightarrow \max n_x$. We thus reduce Problem 2 to the NP-Hard Knapsack problem. □

A. Unconstrained Optimization

We first consider the unconstrained version of Problem 2, i.e., when all the limits I_t, I_c approach infinity. This is useful when the operator has gained complete right of usage of the clients (so that they can be used for free without time limit). This unconstrained optimization can be solved in polynomial time using the following proposition:

Proposition 4. *(Unconstrained Client Recruitment) Suppose clients are sorted by their s values, i.e. $s_1 \le ... \le s_K$. The solution to problem Problem 2 without constraints must be of the form: $x^* = (1, 1, ..., 1, 0, 0, ...0)$, i.e., if a client j is recruited, all the clients $k < j$ must also be recruited.*

Proposition 4 indicates that recruiting more clients does not always help improve the accuracy. Intuitively, when more client participate, the overall dataset grows larger and the representativeness should thus improve. However, a chosen dataset $\mathcal{D}_k$ itself may be small in size, making its data biased from $\mathcal{P}_k$. This will enlarge its generalization error. Worse still, if $\mathcal{P}_k$ is also biased from the population $\mathcal{P}$, the training loss will increase as well due to the increased divergences in local distributions. Based on the proposition, we can solve the unconstrained client recruitment problem by simply sorting the devices by their s values, then comparing the objective values for all the K possible choices of x^*. The **time complexity** is dominated by the sorting step, which is $O(K \log K)$.

To prove the proposition, we use the following lemma.

Lemma 3. *Consider two recruitments x^0 and x^j that contain the same set of clients, except that the latter includes client j while the former does not. If $f(x^j) \le f(x^0)$, then $f(x^j)$ decreases as we increase n_j, the number of data points in j.*

Proof. For convenience we rewrite $f(x^j) = f^j(n_j)$. Note that

$$f^j(n_j) = \frac{\sum_k x_k^0 n_k s_k + n_j s_j}{\sum_k x_k^0 n_k + n_j} + \big(\sum_k x_k^0 n_k + n_j\big)^{-0.5} \tag{12}$$

$$\frac{df^j}{dn_j} = \frac{\sum_k x_k^0 n_k (s_j - s_k)}{(\sum_k x_k^0 n_k + n_j)^2} - \frac{0.5}{(\sum_k x_k^0 n_k + n_j)^{1.5}} \tag{13}$$

As n_j increases, (13) is either i) strictly negative, or ii) first positive then negative. Thus, (12) will either i) strictly decrease, or ii) first increase then decrease. Using the condition $f^j(n_j) = f(x^j) \le f(x^0) = f^j(0)$, the value of n_j must fall into the decreasing interval. Therefore, further increasing n_j will only cause the objective f to decrease. □

We then prove Proposition 4:

Proof. (Proposition 4) We prove by contradiction. Assume the clients are already sorted by the s value. Suppose the optimal recruitment $x^* = x^j$, where client j is the last recruited client, and there exists at least one unrecruited client U_i, such that $i < j, x_i^j = 0$. Denote by $f(x|U(s, n))$ the objective value for recruiting clients in x, plus an additional client U who has parameters s and n. Let x^0 be a copy of x^j, except that client j is not recruited. We thus have $f(x^*) = f(x^j) \le f(x^0)$, and:

$$\begin{aligned} &f(x^j|U(s_i, n_i)) \le f(x^j|U(s_j, n_i)) \\ =&f(x^0|U(s_j, n_i + n_j)) < f(x^0|U(s_j, n_j)) = f(x^j) \end{aligned} \tag{14}$$

The first inequality is due to the condition $s_i \le s_j$. The second follows from Lemma 3 with the fact that $f(x^j) \le f(x^0)$.

ISBN 978-3-903176-37-9 © 2021 IFIP

Therefore, adding the unchosen client i to the recruitment x^j results in a smaller objective value $f(x^j|U(s_i,n_i))$, contradicting that $x^*=x^j$. □

B. Constrained Optimization

As we would expect from our NP-hardness result (Proposition 3), Proposition 4 does not hold when incorporating the constraints. To solve the constrained optimization Problem 2, we first relax the completion time constraint $g_t \le I_t$ by N linear constraints $\mathcal{G}_t(m_1,...,m_N)=\{m_z \le M_t^z\}_{z=1}^N$ on m_z.

$$M_t^z = \min\Big\{\sum\nolimits_{k=1}^{K} \mathbf{1}(\mathcal{Z}(k)=z), \\ \text{argmax}_{m_z}\{g_t(0,...,0,m_z,0,...,0)\le I_t\}\Big\} \tag{15}$$

According to Proposition 2, if $(m_1,...,m_N)$ satisfies the original completion time constraint $g_t(m_1,...m_N) \le I_t$, it also satisfies the relaxed constraint $\mathcal{G}_t(m_1,...,m_N)$. We then construct a new optimization Problem 3. Here we define $s'_k = n_k s_k, I_s = \sum_k^K s'_k$. Problem 3 maximizes a linear objective, subject to $N+2$ linear constraints. This is a multi-dimensional Knapsack problem, and can be solved by the dynamic programming (DP) algorithm [4].

Problem 3. *Data Quantity Maximization*

$$\begin{aligned} \max_{x\in\{0,1\}^K} \quad & n_x = \sum\nolimits_k x_k n_k \\ s.t. \quad & m_z = \sum\nolimits_k \mathbf{1}(\mathcal{Z}(k)=z)x_k \le M_t^z, z=1,...,N \\ & g_c(x) = \sum\nolimits_k x_k c_k \le I_c, \;\; g_s(x)=\sum\nolimits_k x_k s'_k \le I_s \end{aligned}$$

As in conventional DP procedures, we construct a $N+3$ dimensional table $\phi(k,m_1,...,m_N,c,s)$ to keep track of the algorithm states. $\phi(k,m_1,...,m_N,c,s)$ represents the maximum value of n_x we can get, under the conditions that: 1) we only pick from the first k clients (the order of clients does not matter); 2) we recruit at most m_z clients for each group z; 3) the cost is less than or equal to c; and 4) the sum $\sum s'_k \le s$. Conditions 2) to 4) correspond to the three constraints in Problem 3. The DP algorithm gradually increments the recruitment boundary k. For each k, the following recursive relation guarantees the consistency of ϕ:

$$\phi(k,m_1,...,m_N,c,s) = \max\{\phi(k-1,...,m_{\mathcal{Z}(k)}-1, \\ ...,c-c_k,s-s'_k)+n_k, \phi(k-1,m_1,...,m_N,c,s)\} \tag{16}$$

In practice, c and s may be float numbers, but we can easily normalize them to integers. The correctness of the algorithm is obvious by induction. The **time complexity** is bounded by the size of the DP table, which is $O(KI_cI_s\prod_{z=1}^N M_t^z)$.

Now we go back to the original Problem 2. We can observe that when the value of s is fixed, minimizing the objective f is equivalent to maximizing the number of samples n_x. Since the ϕ table records a one to one mapping of s to the maximum n_x, we can utilize ϕ to reconstruct the original objective f.

Formally speaking, given $(m_1,...,m_N,s)$, we define

$$f' = \frac{s}{\phi(K,m_1,...,m_N,I_c,s)} \\ + (\phi(K,m_1,...,m_N,I_c,s))^{-0.5} \tag{17}$$

Algorithm 1 *DP and Revisit.* Solving Problem 2.

```
procedure OPTIMIZE
    $\phi \leftarrow$ (solve Problem 3 with DP), $f^* \leftarrow \infty$
    if $\phi(K,M_1^t,...,M_N^t,I_c,I_s) \le 0$ then
        // Infeasible
        return $\infty$
    for $s=0$ to $I_s$ do
        for $m_1=0$ to $M_t^1$ do
            ......
                for $m_N=0$ to $M_t^N$ do
                    if $g_t(m_1,...,m_N) \le I_t$ then
                        $f^* \leftarrow \min(f^*,$ Equation (17))
    return $f^*$
```

Intuitively, for a solver x^* of Problem 2, if its corresponding s^* and m_z^* are recorded during the DP iteration, then f' should be "related" to the optimal objective value $f(x^*)$. We thus propose Algorithm 1 to solve the constrained client recruitment. Its correctness is shown below. The **time complexity** is dominated by the DP step as $O(KI_cI_s\prod_{z=1}^N M_t^z)$.

Proposition 5. *Algorithm 1 solves Problem 2.*

Proof. Let x^* be a solver of Problem 2, with $n_x^* = \sum x_k n_k, s^* = \sum x_k^* n_k s_k, m_z^* = \sum \mathbf{1}(\mathcal{Z}(k)=z)x_k^*$, then

$$\phi(K,m_1^*,...,m_N^*,I_c,s^*) = n_x^* \tag{18}$$

Otherwise, if the left hand side is smaller, the DP algorithm yields a smaller objective n_x^0 for some recruitment x^0. Both x^0 and x^* satisfy the four conditions in the definition of ϕ at $(K,m_1^*,...,m_N^*,I_c,s^*)$. But replacing x^0 with x^* yields a greater objective $n_x^* > n_x^0$. This contradicts the correctness of DP. In addition, if the left hand side is greater, the DP algorithm finds a recruitment x^0 that has $s^0 = \sum x_k^0 n_k s_k \le s^*$, $n_x^0 = \sum x_k^0 n_k > n^*$, and satisfies all the constraints in Problem 2. Thus, by recruiting x^0, we have $f(x^0) = \frac{s^0}{n_x^0} + (n_x^0)^{-0.5} < \frac{s^*}{n_x^*} + (n_x^*)^{-0.5} = f(x^*)$. This shows x^0 is a better recruitment than x^*, which contradicts the assumption that x^* is an optimal. Thus, since Algorithm 1 iterates through all the feasible elements, we must at some point visit $(K,m_1^*,...,m_N^*,I_c,s^*)$. □

VII. Performance Evaluation

We finally evaluate the performance of our client recruitment strategy with a classification problem and a regression problem. We set the aggregation deadline $E_0=30$. Unless otherwise noted, we assume clients have the default specification: Group I = $(q_f^1=0.001, q_r^1=0.6, \lambda^1=0.1)$. If a client fails upon the aggregation, we replace its $w_k^{t\tau}$ with the previous global weight $w^{t\tau}$. Throughout this section, we uniformly at random set the cost of each client in the range of 1 to 9. We consider three baseline recruitment strategies:

- **All participation:** recruiting all clients. Comparisons with this baseline show the value of intelligent client recruitment.
- **Greedy recruiting by quantity:** greedily choosing clients with the most data samples until any constraint is active.

ISBN 978-3-903176-37-9 © 2021 IFIP

- **Greedy recruiting by quality:** greedily choosing clients with best quality until any constraint is active.

In the case of unconstrained optimization, we force the greedy baselines to choose the same number of clients as the optimal recruitment. By comparing to the latter two baselines, we present the value of considering both quantity and quality of the data. Since we take the training parameters as fixed as discussed in Section III, we will not fine-tune them in the simulation. We pick these values such that all model training in all experiments are fully converged. We then only need to determine the relative weights of the training accuracy (γ_{tl}) and generalizability (γ_{ge}). In practice, they can be tuned by optimizing the unconstrained recruitment through grid search.

A. *Image Classification*

We first consider the MNIST digit recognition problem. We use the same 2NN model as [5]. All clients are equipped with the Adam optimizer and use the same set of training parameters. We use a batch size of 10 for local iterations. The initial learning rate is set to 3e-4, and decays by half every 200 steps. The local epochs $\tau = 30$ and global epochs $T = 50$.

Dataset and clients. To construct the non-IID distributions of local datasets, we assign each client a set of class labels (digits). Clients then randomly sample training images corresponding to the assigned labels. We limit each client to sample 10 to 40 images. For j from 1 to 10, we assign j label(s) to 30 new clients, resulting in total 300 candidate clients. The default MNIST test dataset is used.

Approximation of divergence. We use the "counting classes" method described in Section V-A to approximate the probability divergence. Here we have $L = 10$ classes, thus $\int |\tilde{p}_k - \tilde{p}| = \sum_{i=0}^{9} |\tilde{p}_k(v = i) - \tilde{p}(v = i)|$. For the population distribution, all classes appear with the same probability, so $\tilde{p}(v = i) \equiv 0.1$. If a client k has j labels, $\tilde{p}_k(v = i) = 1/j$ if label i was assigned, or $\tilde{p}_k(v = i) = 0$ otherwise. Thus, $\sum |\tilde{p}_k(v = i) - \tilde{p}(v = i)| = j(\frac{1}{j} - \frac{1}{10}) + (1 - j)\frac{1}{10}$, which clients can easily compute knowing only the number of labels that they see. By tuning the unconstrained recruitment, we choose $\gamma_{tl} = 0.015, \gamma_{ge} = 1$ for all experiments.

Unconstrained recruitment. The left plot of Figure 1 shows the convergence progress of the four recruitment strategies. 64 clients are recruited by the optimal strategy. The optimal recruitment converges the fastest and obtains the highest test accuracy on the fully trained models. Notably, the optimal strategy can increase the test accuracy by 5.0% compared to simply recruiting all clients, which is a big improvement for most classification problems. Figure 2 shows the distribution of recruited clients w.r.t. the number of classes assigned to them. Compared to the greedy-by-quantity strategy, the optimal recruitment chooses fewer low-quality datasets, but more high-quality ones. Also, most clients recruited by the optimal strategy contain more than 30 samples, but the greedy-by-quality recruitment includes lots of small-sized datasets.

Constrained recruitment. The left plot of Figure 3 shows the change of test accuracy when we increase the budget I_c from 20 to 60 and take I_t to be infinity. The optimal strategy obtains the highest accuracy for all the budgets I_c. In the right plot of Figure 3 we drop the I_c constraints and vary the completion time constraint I_t from $15T$ to $25T$. Apart from Group I, we also create a relatively lower-end specification Group II = $(q_f^2 = 0.01, q_r^2 = 0.5, \lambda^2 = 0.05)$. We randomly pick one third of the clients and assign them to Group II. When I_t/T is down to around half of $E_0 = 30$, only 1 or 2 clients are recruited, so the models do not appear to be trained at all. The optimal strategy exhibits the best performance when I_t is reasonably large, improving the accuracy by 10% to 20%.

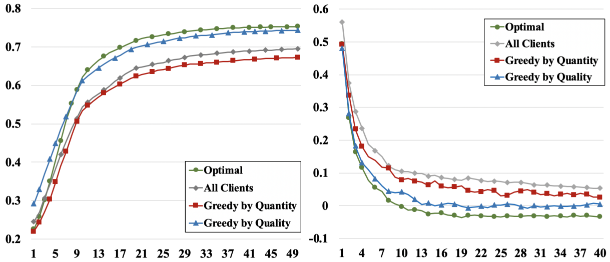

Fig. 1. Convergence curves for the unconstrained recruitments. Left: Classfication problem. The X axis is global epochs, and the Y axis is test accuracy. The optimal strategy's model yields higher test accuracy than other baselines after 10 epochs. Right: Regression problem. The X axis is global epochs, and the Y axis is the normalized MSE on the test dataset. The untrained model has MSE=1, and the closed-form solution has MSE=0. The optimal recruitment can obtain lower MSE than the closed-form solution.

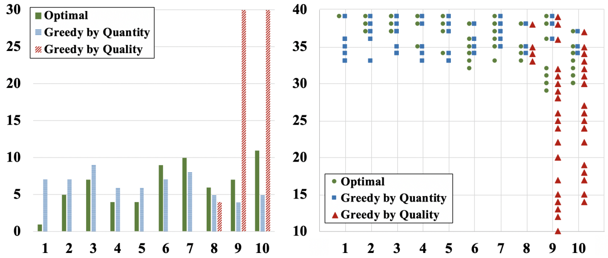

Fig. 2. The distribution of recruited clients w.r.t. the number of assigned classes (X axes). Left plot: counts of recruited clients. Right plot: sizes of local datasets for recruited clients, where each point represents a client.

B. *Climate Data Regression*

We now evaluate client recruitment with a 5-dimensional linear regression model, simulating a climate prediction task. All clients use the Adam optimizer with the initial learning rate set to 1e-3, and decay by 0.8 every 200 steps. We set the batch size as 20, $\tau = 10$, and $T = 40$.

Dataset and clients. We use the U.S. Historical Climatology Network (HCN) dataset [6], which contains climate records for climate stations in the 48 contiguous United States. The local datasets of these stations are by their nature non-IID, allowing us to evaluate how well our recruitment algorithm performs on realistic data distributions. For simplicity, we only use the data on the first day of December from 1960 to 2019, and we randomly pick 1-3 stations from each state, resulting in 117 stations. Each record contains 5 features: station latitude, station longitude, lowest temperature of the day, highest temperature of the day, and precipitation of the day. Our goal is to predict the snowfall of the day. To reflect

ISBN 978-3-903176-37-9 © 2021 IFIP

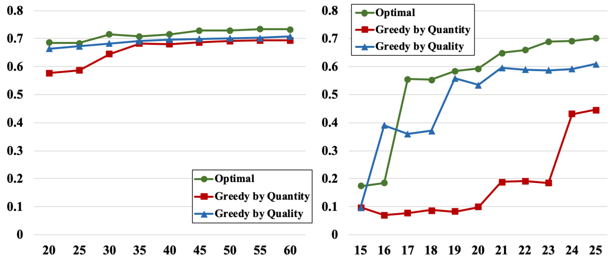

Fig. 3. Left: Test accuracy when varying the budget I_c from 20 to 60 (i.e. 4x to 12x of the expected cost per client). Right: Test accuracy when varying the per round completion time I_t/T from 15 to 25 (i.e. 1/2 to 5/6 of E_0). The optimal strategy consistently has the highest accuracy.

the uneven sizes of local datasets, we randomly drop some data so that each client has 30 to 69 samples. We test the learned models on a holdout dataset, which is generated by randomly picking 2 unused stations from each state.

Approximation of divergence. We use the second approximation method described in Section V-A by assuming the 5 features and the snowfall form a fully connected Gaussian graphic model $\mathcal{N}(\mu, \Sigma)$. Thus, each local distribution can be parameterized by the sample mean and the sample covariance $\mathcal{N}(\tilde{\mu}_k, \tilde{\Sigma}_k)$. Similarly, we approximate the population distribution $\mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$ utilizing the unused (neither training nor testing) data. Thus, we only need to compute the divergence between the local Gaussian $\mathcal{N}(\tilde{\mu}_k, \tilde{\Sigma}_k)$ and population Gaussian $\mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$. We normalize the divergences to the range of 0 and 10, and we choose the coefficients $\gamma_{tl} = 0.01$, $\gamma_{ge} = 1$.

Unconstrained recruitment. The right plot in Figure 1 shows the mean-squared error (MSE) on the holdout dataset, which includes 1-2 stations from each state, for different strategies. 37 clients are chosen by the optimal recruitment, allowing us to drop most clients as in Section 5.1. Since linear regression is a convex problem, we can easily calculate the closed-form optimal model over the full dataset. For ease of comparison, we normalize the MSE values so that the untrained model has MSE equal 1, and the closed-form solution has MSE equal 0. As in Figure 1, the optimal recruitment yields a lower MSE even than the closed-form solution, which illustrates the value of incorporating generalizability and representativeness metrics. Compared to other strategies, the optimal recruitment can decrease the MSE up to 10%.

Similar to Figure 2, Figure 4 shows the distribution of recruited clients. Here we divide the clients based on their local-population distribution divergences into 10 bins.

Constrained recruitment. Figure 5 shows the change of MSE when varying the cost and time limits, on the same setup as in Section VII-A. The optimal recruitment obtains the lowest MSE and much smaller variance in most cases.

VIII. Conclusion

This paper studies the client recruitment problem in federated learning. We first introduce and quantify five performance metrics that cover both the model's accuracy (training loss, generalization error, representativeness) and the training efficiency (completion time, cost). We then formulate the client recruitment as an NP-Hard optimization problem, and provide an optimal solution algorithm. Finally, we verify our theoretical results with experiments using both synthetic and real-world data. Our results show that recruiting more clients does not always improve the model, and intelligent client recruitment can greatly improve the accuracy of the trained model in constrained execution environments.

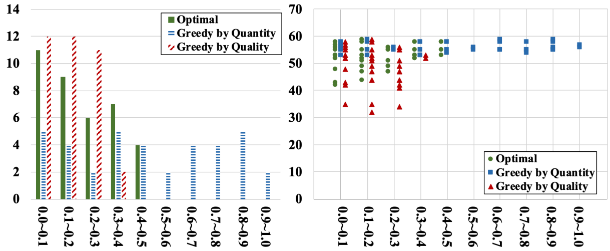

Fig. 4. The distribution (left: client count; right: dataset size per client) of recruited clients w.r.t. the distribution divergence. X axes are quantile ranges. Left bins correspond to small divergence (i.e. good quality).

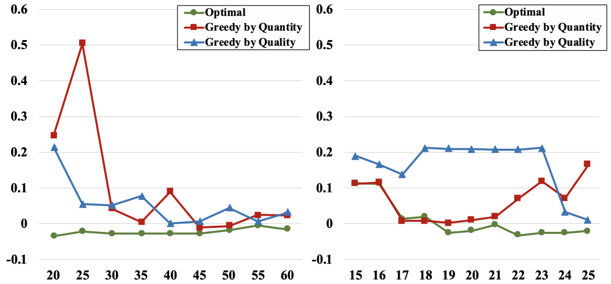

Fig. 5. Left: Test MSE when varying the budget I_c from 20 to 60. Right: Test MSE when varying I_t/T from 15 to 25. The optimal strategy consistently has the lowest error.

References

[1] Y.J. Cho, J. Wang, and G. Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020.

[2] C. Li, D. Li, et al. A theory of pricing private data. *Communications of the ACM*, 60(12):79–86, 2017.

[3] X. Li, K. Huang, et al. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[4] S. Martello. Knapsack problems: algorithms and computer implementations. *Wiley-Interscience series in discrete math and optimization*, 1990.

[5] H.B. McMahan et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[6] M.J. Menne, I. Durre, et al. An overview of the global historical climatology network-daily database. *Journal of Atmospheric and Oceanic Technology*, 29(7):897–910, 2012.

[7] T. Nishio and R. Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC*, pages 1–7. IEEE, 2019.

[8] B.L.S.P. Rao. The rate of convergence of the least squares estimator in a non-linear regression model with dependent errors. *Journal of multivariate analysis*, 14(3):315–322, 1984.

[9] V. Smith, C. Chiang, et al. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[10] T. Tuor et al. Data selection for federated learning with relevant and irrelevant data at clients. *arXiv preprint arXiv:2001.08300*, 2020.

[11] S. Wang, T. Tuor, et al. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

[12] H.H. Yang et al. Scheduling policies for federated learning in wireless networks. *IEEE Transactions on Communications*, 2019.

[13] T. Yang et al. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.