The Analysis and Optimization of Volatile Clients in Over-the-Air Federated Learning

Fang Shi, Weiwei Lin, *Senior Member, IEEE*, Xiumin Wang, *Member, IEEE*, Keqin Li, *Fellow, IEEE*, and Albert Y. Zomaya, *Fellow, IEEE*

***Abstract*—This paper investigates the implementation of Federated Learning (FL) in an over-the-air computation system with volatile clients, where each client operates under a limited energy budget and may unexpectedly drop out during local training sessions. The dropout of clients not only wastes energy but also diminishes their participation frequency, necessitating careful client selection by the server in each communication round. However, the diversity of training tasks and the random nature of client dropout present challenges such as the absence of an explicit objective function and the unavailability of client performance metrics. To address these challenges, we first analyze the convergence of the over-the-air federated learning system with volatile clients to identify the key factor influencing the model's convergence speed. Building upon this analysis, we propose an approximation of the objective function as the optimization goal for client selection. To mitigate energy waste, we introduce a dynamic client selection strategy termed DCSE, based on Exp3 with multiple plays and energy constraints, aiming to reconcile the dilemma of unknown local training states and limited resource constraints. Theoretical analysis demonstrates that our proposed solution maintains a constant bound on the difference from the optimal solution, affirming its theoretical feasibility. Furthermore, experimental results validate the effectiveness of the proposed strategy in enhancing FL by accelerating convergence speed, improving test accuracy, and reducing wasted energy.**

***Index Terms*—Client selection, energy constraint, federated learning, over-the-air computation, volatile client.**

Manuscript received 26 August 2023; revised 7 June 2024; accepted 7 July 2024. Date of publication 16 July 2024; date of current version 5 November 2024. This work was supported in part by Guangdong Major Project of Basic and Applied Basic Research under Grant 2019B030302002, in part by the National Natural Science Foundation of China under Grant 62072187, in part by Guangzhou Development Zone Science and Technology Project under Grant 2021GH10, and in part by the Major Key Project of PCL, China under Grant PCL2023A09. Recommended for acceptance by Y. Liu. *(Corresponding author: Weiwei Lin.)*

Fang Shi is with the College of Mathematics and Informatics (Software Engineering), South China Agricultural University, Guangzhou 510642, China (e-mail: csshifang@scau.edu.cn).

Weiwei Lin is with the School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: linww@scut.edu.cn).

Xiumin Wang is with the School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: xmwang@scut.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Albert Y. Zomaya is with the School of Computer Science, The University of Sydney, Sydney, NSW 2050, Australia (e-mail: albert.zomaya@sydney.edu.au).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2024.3427709, provided by the authors.

Digital Object Identifier 10.1109/TMC.2024.3427709

I. Introduction

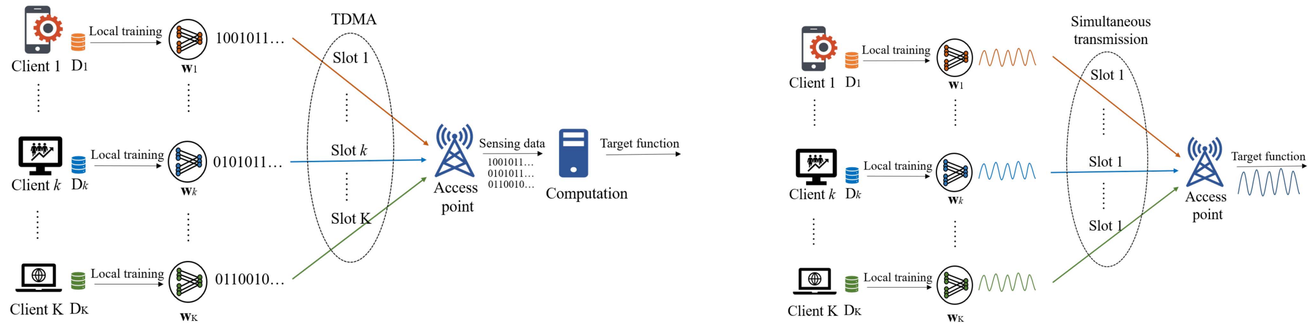
A. Background

As THE number of intelligent devices equipped with advanced sensors, computing, and communication capabilities continues to rise, the data they collect (e.g., photos, videos, and locations) offers endless possibilities for intelligent applications powered by Machine Learning (ML) [1]. However, traditional centralized ML is inadequate as these data-driven technologies require massive amounts of data to function effectively. To address this challenge, emerging edge learning technologies within B5G Networks leverage distributed signal processing [2], encompassing advancements like edge computing [3], semantic communication [4], and wireless sensing network [5]. In response to the imperative need for heightened privacy and empowered by these technologies, Federated Learning (FL) [6] has emerged as a timely solution. As a privacy-preserving solution, it allows training to be performed locally. Its main idea is that multiple clients/devices train a shared model collaboratively by conducting model training locally and retaining their data without transmitting it. The clients only need to upload their model updates to the central server for collaboration.

Although FL reduces the transmission load by replacing transmission data with transmission model updates, the network congestion still exists if too many clients participate in the collaborative training as the limited bandwidth. To solve this bottleneck and improve the communication efficiency, some works propose model compression [7], [8], multiple local iterations [9], [10], and client selection [11], [12], [13], [14]. While previous studies have focused on implementing FL based on a separated communication-and-computation principle encounter difficulty in accommodating the massive access under the limited radio resource and stringent latency constraints imposed by emergence applications, such as auto-driving and instant messaging [15]. To tackle this challenge, some studies propose a low-latency multi-access scheme known as over-the-air computation (AirComp) [16], [17]. AirComp leverages the characteristic of the wireless medium where multiple signals can be superimposed simultaneously in an analog manner to enable concurrent transmission. In other words, it integrates model transmission and aggregation into a single step by using the signal superposition characteristics of multiple access channels. The illustration "transmit-then-compute" versus Over-the-air computation FL is shown in Fig. 1.

1536-1233 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.



(a) Traditional "transmit-then-compute". (b) Over-the-air computation.

Fig. 1. Traditional "transmit-then-compute" versus Over-the-air computation.

B. Motivation

Undoubtedly, some scaling approaches can allow more clients in model training and make AirComp FL perform well in the ideal learning environment, such as power scaling [18]. Moreover, Yang et al. [19] used experiments to prove that selecting more clients participating in the model training can make a better training performance. And combined with beamforming design, they designed a client selection method to allow more clients to participate in model training. In [20], Guo et al. proposed a joint scheme that combines client selection and power control strategy to reduce the performance gap with the ideal FedAvg scheme. These works share a similar observation that covering more clients indeed benefits the overall training performance.

However, since most of the clients participating in model training are terminal devices connected by wireless networks, they exhibit significant dynamics. This is particularly pronounced in wireless sensing networks [21], where sensors and users may be mobile, leading to environmental changes. Furthermore, sensor failure or malfunction can occur due to environmental factors or depletion of electrical energy [22], [23]. Thus, some limited and unstable factors in the real learning environment need to be included in consideration. The first and important one is the *limited energy* on the client. In the training process, energy is a key metric for judging whether a client can join FL training, and it limits the times of client participating. Especially in the energy-limited learning scenario, it is impossible that the client can participate in model training for an unlimited times [24], [25], [26]. Thus, how to use the limited energy to carry out model training is a practical problem. The second and more practical is the *client dropout*, which mSay occur during the local training process due to various unexpected and unstable reasons (e.g., poor connection, limited resource, and user abort (need to make phone)). For these clients that may dropout training unexpectedly, existing works in traditional FL framworks refer them to volatile clients [27], [28], [29].

Although existing works in the traditional FL framework have made some contributions to this concern, there is still a lack of consideration in AirComp FL. It is well known that clients need to perform amplitude alignment before transmitting information. In cases where all clients can successfully complete local training, adjusting the transmit power based on initial information provided by clients and selecting clients that satisfy constraints would be a good strategy. However, if some clients experience interruptions during local training due to unexpected factors, the feasibility of adjusting transmit power before clients complete local training needs to be further analyzed, as it may result in aggregation errors. Additionally, if the server consistently selects clients with a high dropout probability, the energy on the client side will be wasted, reducing the client's remaining energy and participation times, thereby further slowing down model convergence. Considering these concerns, we believe it is important to discuss the analysis and optimization problem of volatile clients with limited energy in over-the-air federated learning.

Based on these analyze, this paper will focus on investigating the selection problem of volatile clients with limited energy in over-the-air federated learning. Specifically, we tackle the aforementioned concerns by analyzing the impact of the volatile clients on workflow, energy consumption, and convergence performance. Based on these results, we propose a dynamic client selection scheme for optimizing model training. Specifically, this paper's contributions can be listed as follows.

- We investigate an over-the-air federated learning system with volatile clients, where each client possesses a limited energy budget and may unexpectedly drop out during local training. The system confronts challenges stemming from the lack of an explicit objective function, the unavailability of local training state information, and energy limitations due to the diverse nature of training tasks and the stochastic dropout of clients.
- We investigate the influence of the selected client set and channel noise on FL training performance, establishing a stringent upper bound on convergence speed. Our analysis reveals that convergence speed is primarily dictated by the total aggregation number and the power scaling factor.
- We propose DCSE, a dynamic client selection method based on Exp3 with multiple plays and energy constraints. We incorporate a penalty term to address the impact of energy consumption and provide theoretical validation, demonstrating that the upper bound of regret remains below a finite constant, thereby establishing the theoretical viability of our solution.

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

- We assess the efficacy of our approach through comprehensive experiments conducted on two machine learning tasks employing various client selection schemes. Our experimental findings demonstrate that our strategy significantly enhances FL performance in terms of convergence speed, test accuracy, and energy consumption.

II. Related Works

Diverging from conventional distributed learning and eadge computing, the distinctive features of FL lie in constrained transmission resources and an inherently fluctuating learning process. Addressing the transmission bandwidth bottleneck, certain initiatives have integrated over-the-air computation with FL, termed AirComp FL, aiming to enhance transmission efficiency. Additionally, in response to the challenge of sluggish convergence arising from an unstable learning process, other endeavors have introduced client selection mechanisms to mitigate the influence of volatile contributors. Consequently, the subsequent sections will delve into relevant research endeavors focusing on these dual aspects.

A. AirComp FL

As shown in Fig. 1(a), AirComp leverages the characteristic of wireless medium where multiple signals can be superimposed simultaneously in an analog manner to enable concurrent transmission and solve the bottleneck of bandwidth. In related works on AirComp FL, Yang et al. [19] earlier proposed combining AirComp and FL to improve the transmission effectiveness of FL. In follow-up works, some researchers proposed the joint optimization strategy of transmit power and model training [30], [31]. To further improve the training performance, some works disscussed the optimization problem with IRS-assisted AirComp FL [32], [33]. In addition, some works took resource consumption into consideration and analyzed the impact of resource consumption on model training. For example, Yang et al. [34] disscussed an optimization problem of minimizing the total consumption under a latency constraint, and proposed an iterative algorithm with low complexity. In [35], Hu et al. discussed the energy minimization problem in IRS and AirComp assisted network and proposed a novel FL model. To minimize energy consumption, Zeng et al. proposed energy-efficient strategies for bandwidth allocation and scheduling. These strategy adapt to clients' channel states and computation capacities, aiming to reduce their total energy consumption while ensuring learning performance [36].

The aforementioned works share the same assumption: they assume that all participating clients are situated in a stable environment, where each client can successfully complete the local training. However, many participating clients are edge devices characterized by greater instability, there is a risk of unexpected termination. Especially in the process of local training, prior studies have made notable contributions to this concern in traditional FL framework [27], [28], [29]. Clearly, such unexpected terminations also impact the training performance of AirComp FL, and perhaps even more so. This is because AirComp FL involves an additional step in amplitude alignment compared to traditional FL, requiring the adjustment of the power transmission factor based on all transmitted clients' information. Obviously, if all clients can successfully complete local training, adjusting the transmit power based on initial information provided by clients and selecting clients that satisfy constraints would be a good strategy. However, if some clients experience interruptions during local training due to unexpected factors, the feasibility of adjusting transmit power before clients complete local training needs to be further analyzed, as it may result in aggregation errors. Moreover, such unexpected terminations will inevitably impact the training performance of the model and result in wasted energy consumption on the client side. Especially in energy-limited networks, the unknown training state and energy consumption of clients significantly increase the challenge of client selection. Accordingly, for the energy-limited network, it is imperative to examine the impact of volatile clients on AirComp FL.

B. Client Selection

Since the bandwidth limits the number of participants, client selection as a common method has been used in traditional FL framwork for improving training performance. The earliest documented instance of client selection dates back to [11], Nishio et al. proposed FedCS that prioritizes faster clients for participation in training, effectively reducing the overall training time for the global model. In the following works, some researcher proposed different selection strategies tailored to different application scenarios. For example, in [27], Wang et al. casted FL with client dropout as a special case of a larger class of FL problems. Meanwhile, they selected local updates from clients with similar data distributions as substitutes for dropout clients, aiming to minimize substitution errors and enhance convergence performance. In [37], a latency-based client scheduling scheme is proposed to shorten the time interval for each round. In [29], Huang et al. addressed the stochastic client selection problem in the presence of volatile clients, and proposed a client selection method based on Exp3 without resource constraint. Notably, in our previous works [38], [39], we conducted an analysis on how different factors of volatile clients affecting model training. based on this analysis, we then proposed diverse client selection strategies tailored to various scenarios.

In AirComp FL, although not constrained by bandwidth, existing research has demonstrated that both the quality and quantity of participating clients significantly influence model training outcomes [18], [19], [20]. For instance, Yang et al. in [19] conducted experiments highlighting that involving more clients in model training improves overall performance. Similarly, Guo et al. proposed a joint scheme in [20], combining client selection and power control strategies to narrow the performance gap. In [40], An et al. proposed a joint power control and data size selection approach for AirComp FL to enhance training performance. In [41], Sun et al. introduced an online energy-aware dynamic client scheduling policy to maximize the average aggregation numbers. Meanwhile, in their subsequent work [24], they further analyzed the client's queuing theory problem within the energy constraint and designed an energy-aware dynamic

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

TABLE I
LIST OF SYMBOLS

Symbol	Description	Symbol	Description
$\mathbf{w}_{k,t}$	local model at client	$\mathbf{w}_t$	global model at server
A_t	client selection set	$x_{k,t}$	completion state of local training
A_t'	client aggregation set	$E_{k,t}^{cmp}$	energy consumption of local training
$\boldsymbol{n}_t$	noise vector	$E_{k,t}^{com}$	energy consumption of communicatin
σ_t	power scaling factor	$b_{k,t}$	progress bar of local training
$h_{k,t}$	channel gain	$E_{k,t}^{suc}$	total energy consumotion
$P_{k,t}$	transmit power	$E_{k,t}^{wat}$	wasted energy consumption

client selection strategy to boost FL performance. However, these studies did not account for the impact of client dropout. As discussed in the introduction, client dropout affects not only transmit power adjustments but also the quality of aggregation. Hence, we believe that analyzing the impact of client dropout on client selection and aggregation in AirComp FL is crucial.

III. System Model

A. System Overview

In this paper, we investigate an over-the-air federated learning system with volatile clients. The FL system comprises a set of clients $\mathcal{K} = \{1, 2, \ldots, K\}$, each with a local dataset $\mathcal{D}_k$, and connected to the server via a wireless network. The primary aim of federated learning is to train a global model $\mathbf{w}$ that minimizes the global loss function, which can be expressed as a specific mathematical formula

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) = \frac{1}{K}\sum_{k=1}^{K} F_k(\mathbf{w}), \tag{1}$$

where the dimension of $\mathbf{w}$ is d, $F(\cdot)$ and $F_k(\cdot)$ represent the global and local loss functions, respectively. The local loss of client k at round t is typically expressed as the sum of empirical risks over all training data, i.e.,

$$F_k(\mathbf{w}) = \frac{1}{|\mathcal{D}_k|}\sum_{i=1}^{|\mathcal{D}_k|} l_i(\mathbf{w}), \tag{2}$$

where $|\mathcal{D}_k|$ is the cardinality of $\mathcal{D}_k$, the function $l_i(\cdot)$ is used to denote the training loss of the i-th data sample, which depends on the specific training task. For instance, in the case of linear regression, $l_i(\mathbf{w})$ for an input-output pair (x_i, y_i) can be expressed as $\frac{1}{2}(x_i^\top \mathbf{w} - y_i)^2$. Similarly, in logistic regression, $l_i(\mathbf{w})$ can be expressed as $-\log(1+\exp(-y_i x_i^\top \mathbf{w}))$, and in support vector machines, $l_i(\mathbf{w})$ can be expressed as $\max\{0, 1 - x_i^\top \mathbf{w}\}$ [42]. In more complex non-convex problems, such as neural networks, it can be expressed as mean square and cross-entropy errors.

AirComp is a novel communication scheme that differs from traditional orthogonal multiple access methods in that it allows for simultaneous transmission and computation over the air, and enables multiple transmissions via the same channel to improve communication efficiency. Fig. 1(b) provides a detailed illustration of this principle. In a scenario where all clients successfully complete their local training, the server only needs to consider the initial information provided by the clients to adjust the transmit power and select clients that meet the energy constraint to participate in the training. However, in case of unexpected interruptions during local training of some clients, the server can only adjust the transmit power when the clients return to the trained model and trained state. Consequently, the workflow with dropout clients differs from that of a normal learning network. In each round, the workflow involves several sequential steps:

- *Client selection and model distribution:* The server selects a subset of clients A_t that meet the energy constraint, and then disseminates the latest global model to the chosen clients.
- *Local training:* Each selected client $k \in A_t$ trains their local models on their respective datasets $\mathcal{D}_k$. However, due to unforeseeable circumstances, some clients may experience local training failures and drop out. Only the clients completing the local training will update their local models using stochastic gradient descent (SGD), denoted by $\mathbf{w}_{k,t+1} = \mathbf{w}_{k,t} - \eta_t \mathbf{g}(\mathbf{w}_{k,t}, \xi_{k,t})$. In this equation, η_t represents the learning rate at round t, and $\mathbf{g}(\mathbf{w}_{k,t}, \xi_{k,t}) = \frac{1}{b}\sum_{i\in\xi_{k,t}} \nabla l_i(\mathbf{w}_{k,t})$ represents the stochastic gradient computed on a randomly sampled mini-batch $\xi_{k,t}$ of size b from $\mathcal{D}_k$.
- *Model transmission and model aggregation:* Once the selected clients finish their local training, the server immediately adjusts the transmit power according to the returned training state and transmission information. The simultaneous time-synchronized signals are transmitted by the clients successfully finishing the local training. The weighted sum received by the server is represented by $\mathbf{w}_t = \frac{1}{\sum_{k\in A_t} x_{k,t}} \sum_{k\in A_t} x_{k,t}\mathbf{w}_{k,t}$. Here, $x_{k,t} \in \{0, 1\}$ represents the completion state of local training on client k at communication round t. If $x_{k,t} = 1$, it indicates that the local training at client k has been successfully completed, otherwise, $x_{k,t} = 0$.

B. Over-the-Air Computation for Aggregation

We make the assumption that all clients communicate their updates over a wireless fading multiple access channel in an analog manner for the purpose of model aggregation. To allow for the aggregation of local updates over-the-air, the server synchronizes the transmissions of all clients completing the local training, and the transmit power of each client is adjusted to match that of others. In each round t, the target function at the server side can be formulated as:

$$\mathbf{w}_t = \frac{1}{\sum_{k\in A_t} x_{k,t}} \sum_{k\in A_t} x_{k,t}\mathbf{w}_{k,t}. \tag{3}$$

To better analyze, we define A_t' to represent the set of clients completing the local training, i.e., $A_t' = \{k \in A_t : x_{k,t} = 1\}$. Thus, the target function is equivalent to

$$\mathbf{w}_t = \frac{1}{|A_t'|}\sum_{k\in A_t'} \mathbf{w}_{k,t}. \tag{4}$$

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

During the transmission process, since the existence of channel noise, the server-side signal $\boldsymbol{y}_t$ is [24]

$$\boldsymbol{y}_t = \sum_{k\in A_t} P_{k,t} h_{k,t} \mathbf{w}_{k,t} + \boldsymbol{n}_t, \tag{5}$$

where $h_{k,t}$ represents the wireless channel gain between client k and the server at round t, $\boldsymbol{n}_t \sim CN(0, \sigma_0^2 \boldsymbol{I})$ represents the additive white Gaussian noise vector, $P_{k,t}$ represents the transmit power of client k, and it is assumed to be aligned with the transmit power of other selected clients. Specifically,

$$P_{k,t} = \frac{\sqrt{\sigma_t}}{h_{k,t}}, \tag{6}$$

where σ_t is the power scaling factor. Upon receiving $\boldsymbol{y}_t$, the model received by the server side can be represented as

$$\bar{\mathbf{w}}_t = \frac{1}{\sqrt{\sigma_t}} \boldsymbol{y}_t = \frac{1}{\sqrt{\sigma_t}} \sum_{k\in A'_t} h_{k,t} P_{k,t} \mathbf{w}_{k,t} + \frac{\boldsymbol{n}_t}{\sqrt{\sigma_t}}. \tag{7}$$

According to (4) and (6), the aggregated model can be obtained as

$$\bar{\mathbf{w}}_t = \frac{1}{|A'_t|} \sum_{k\in A'_t} \mathbf{w}_{k,t} + \frac{\boldsymbol{n}_t}{|A'_t|\sqrt{\sigma_t}}. \tag{8}$$

From the above equation, it is evident that the quality of the aggregation model is primarily influenced by the aggregation number, power scaling factor, and noise. Clearly, if we set the controllable parameter σ_t based solely on the clients' initial information, it is likely to be greater than the actual required value, leading to an increase in the aggregation error. Therefore, setting the value of σ_t appropriately is crucial.

IV. Performance Analysis and Problem Formulation

A. Energy Consumption

We consider an energy-limited wireless scenario where all clients have limited energy and need it to carry out model training. Since the energy is mainly consumed during the local training and model transmission processes, we focus on analyzing the energy consumed in computation and communication phases.

Correspondingly, in each communication round, the energy consumption of client k in computation phase is determined by several factors, such as the training power $P_{k,t}^{cmp}$, the CPU frequency $f_{k,t}$, and the time of local training $\tau_{k,t}^{cmp}$. If we use $E_{k,t}^{cmp}$ to represent the energy consumption of local training, its mathematical relationship can be expressed as

$$E_{k,t}^{cmp} = P_{k,t}^{cmp} f_{k,t}^3 \tau_{k,t}^{cmp}, \tag{9}$$

where $\tau_{k,t}^{cmp}$ represents the time of local training, and its expression can be formulated as

$$\tau_{k,t}^{cmp} = \frac{|\mathcal{D}_k| \cdot M}{f_{k,t}} b_{k,t}, \tag{10}$$

where $M = NLE * BPS * CPB$ represents the size of training model, $b_{k,t} \in [0, 1]$ represents the progress bar of local training completion on client k at round t. Factors such as low battery and instability can cause the progress bar to be less than one.

Once the local training is complete ($b_{k,t} = 1$), the client transmits the updated model to the server, and the local updates from multiple clients are aggregated over the air. Inspired by [24], we define $E_{k,t}^{com}$ to represent the energy consumed by client k during the communication phase at round t, and its specific expression is

$$E_{k,t}^{com} = ||P_{k,t}\mathbf{w}_{k,t}||^2 \, \tau_t^{com} = \frac{d\sigma_t}{Bh_{k,t}^2} ||\mathbf{w}_{k,t}||^2 . \tag{11}$$

Here, we express the transmission time as the duration required from sending the first data to sending the last data, i.e., $\tau_t^{com} = \frac{d}{B}$, where d and B respectively represent the data size and the bandwidth. Similarly, we define $E_{k,t}^{suc}$ to represent the total energy consumed by client k during the local training phase and communication phase. For the case $b_{k,t} = 1$, its formulation can be expressed as

$$\begin{aligned} E_{k,t}^{suc} &= E_{k,t}^{cmp} + E_{k,t}^{com} \\ &= P_{k,t}^{cmp} f_{k,t}^2 |\mathcal{D}_k| \cdot M + \frac{d\sigma_t}{Bh_{k,t}^2} ||\mathbf{w}_{k,t}||^2 . \end{aligned} \tag{12}$$

For the case $b_{k,t} < 1$, since it fails to provide updates to the server, wasting the client's energy, we consider this is an ineffective participation. Further, we represent the wasted energy on client as

$$E_{k,t}^{wat} = b_{k,t} P_{k,t}^{cmp} f_k^2 |\mathcal{D}_k| \cdot M. \tag{13}$$

It is worth noting that conventional over-the-air federated learning lacks a parameter reflecting the progress of local training, denoted as $b_{k,t}$. This omission arises because it does not account for the phenomenon of client dropout. Therefore, in this work, we introduce $b_{k,t}$ to illustrate the effect of various training cases on energy consumption. For convenience, the commonly used symbols in this paper have been summarized in Table 1.

B. Problem Formulation

Given the initial global model vector $\mathbf{w}_0$ and the communication round set $\mathcal{T} = \{0, \ldots, T-1\}$, our task is to minimize the expected global loss after T communication rounds, denoted as $\mathbb{E}[F(\mathbf{w}_T)]$, subject to an energy constraint by optimizing the client selection set A_t and power scaling factor σ_t. The expected global loss is calculated by taking into account the randomness of data sampling for local stochastic gradient descent and channel noise. In summary, we seek to solve the following optimization problem:

$$\text{P1:} \min_{\{A_t, \sigma_t\}_{t\in\mathcal{T}}} \mathbb{E}\left[F(\mathbf{w}_T)\right] \tag{14}$$

$$\text{s.t.} \sum_{t=0}^{T-1} A_{k,t} \left(x_{k,t} E_{k,t}^{suc} + (1 - x_{k,t}) E_{k,t}^{wat} \right) \le E_k, \quad \forall k, t \tag{14a}$$

$$A_{k,t} \in \{0, 1\}, \quad \forall k, t \tag{14b}$$

where the first constraint (14a) sets an energy limit $E_k > 0$ on each client, which indicates that the total energy consumed by

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

each client during the local training phase and communication phase at any round t should be less than the remaining energy. The second constraint (14b) restricts the range of optimization variables.

Observing the optimization problem P1, we can find that solving it has the following dilemmas:

- *Inexplicit form of the objective function:* Based on (1) and (2), it is evident that the global loss function is explicitly defined based on the training task, which makes it challenging to derive a closed-form expression for $\mathbb{E}[F(\mathbf{w}_T)]$. A feasible solution is to find an approximation of the objective function and transform it into a problem that can be solved.
- *Unavailability of local training state:* If the server always selects all clients satisfying the energy constraint, it will lead to a waste of energy for the clients that failed in the local training, and further reduce their opportunities to participate in future rounds. On the contrary, if the server just selects the clients that can successfully finish the local training, the energy utilization rate of the client will be maximized. However, the main challenge is that the local training state $x_{k,t}$ can only be determined once the local training is completed, and the server cannot know this value beforehand. Therefore, a client selection strategy needs to be designed to minimize the wasted energy under the case that the local training state is unknown.
- *Limitation and unavailability of energy:* Referring to (11), we understand that the communication energy is primarily influenced by the computation result to be transmitted, i.e., the l_2-norm of the local model. Nevertheless, as this value can only be obtained after completing the local training, we cannot know the communication energy in advance. Recall the AirComp workflow that the client selection must be determined before local training due to the energy constraint, it makes the client selection a challenging problem to tackle.

V. Convergence Analysis and Problem Transformation

A. Convergence Analysis

To facilitate convergence analysis, we introduce a virtual sequence denoted as $\bar{\mathbf{w}}$, which is updated at the server after each communication round. The detailed update is carried out using the following rule:

$$\bar{\mathbf{w}}_{t+1} = \bar{\mathbf{w}}_t - \eta_t\left(\bar{\mathbf{g}}_t - \bar{\mathbf{n}}_t\right), \tag{15}$$

where $\bar{\mathbf{g}}_t \triangleq \frac{1}{|A'_t|}\sum_{k\in A'_t}\mathbf{g}(\mathbf{w}_{k,t},\xi_{k,t})$ is the average gradient of clients that successfully finished local training, and $\bar{\mathbf{n}}_t \triangleq \frac{\mathbf{n}_t}{\eta_t\sqrt{\sigma_t}|A'_t|}$ is the noise received at the server. To facilitate a more thorough analysis, we define $\mathbf{w}^*$ to represent the vector that minimizes $F(\mathbf{w})$, with $F(\mathbf{w}^*)$ denoting the minimum value of $F(\mathbf{w})$. In addition, we make the following assumptions for the model's convergence analysis:

Assumption 1: The local loss function for any client k is L-smooth, i.e., for all $\mathbf{v}$ and $\mathbf{v}_0$, $F_k(\mathbf{v}) \le F_k(\mathbf{v}_0) + \langle\nabla F_k(\mathbf{v}_0), \mathbf{v}-\mathbf{v}_0\rangle + \frac{L}{2}||\mathbf{v}-\mathbf{v}_0||^2$ and $\frac{1}{L}||\nabla F_k(\mathbf{v}) - \nabla F_k(\mathbf{v}_0)||^2 \le \langle\nabla F_k(\mathbf{v}) - \nabla F_k(\mathbf{v}_0), \mathbf{v}-\mathbf{v}_0\rangle$.

Assumption 2: The local loss function for any client k is μ-strongly convex, i.e., for all $\mathbf{v}$ and $\mathbf{v}_0$, $F_k(\mathbf{v}) \ge F_k(\mathbf{v}_0) + \langle\nabla F_k(\mathbf{v}_0), \mathbf{v}-\mathbf{v}_0\rangle + \frac{\mu}{2}||\mathbf{v}-\mathbf{v}_0||^2$.

Assumption 3: The stochastic gradient for the mini-batch $\xi_{k,t}$ randomly sampled from $\mathcal{D}_k$, is assumed to be unbiased, i.e., $\mathbb{E}[\mathbf{g}(\mathbf{w}_{k,t},\xi_{k,t})] = \nabla F_k(\mathbf{w}_{k,t})$.

Assumption 4: The expected norm of the stochastic gradient is bounded uniformly, i.e., $\mathbb{E}[||\mathbf{g}(\mathbf{w}_{k,t},\xi_{k,t})||^2] \le G^2$.

Building on the aforementioned assumptions, we proceed to investigate the impact of local training at the clients on the convergence performance. The related results are summarized in the following theorem, which provides the convergence result.

Theorem 1: Under Assumption 1 to 4, given the set of selected clients $A_t, t\in\mathcal{T}$, we can bound the error incurred after T rounds of over-the-air federated learning with volatile clients as follows:

$$\mathbb{E}\left[F(\bar{\mathbf{w}}_T) - F(\mathbf{w}^*)\right] \le \frac{4\eta_0^3 G^2 L^2}{\sum_{t\in\mathcal{T}}|A'_t|} + \frac{\hat{a}}{\sum_{t\in\mathcal{T}}|A'_t|^2} + C \tag{16}$$

where the learning rate satisfies $\eta_t = \eta_0(1-\frac{t^2}{tT+1})$, η_0 is the initial learning rate at $t=0$, $\hat{a} = \max\{a_t\}_{t\in\mathcal{T}}$, $a_t = (\sigma_0^2 d\eta_t L^2 + \frac{\sigma_0^2 d}{2\eta_t} + \frac{\sigma_0^2 dL}{2})/\sigma_t$, and $C = F(\bar{\mathbf{w}}_0) - F(\mathbf{w}^*) + \frac{1}{2}L\eta_0^2 G^2 T$.

Proof: Complete proof is presented in the supplement file. ■

B. Problem Transformation

It's evident that P1 falls under the category of combined nonlinear programming problems. Consequently, directly solving P1 presents challenges. Fortunately, prior research indicates that the convergence speed of federated learning via over-the-air computing remains relatively insensitive to the power scaling factor σ_t as long as the signal-to-noise ratio at the receiving end exceeds a certain threshold [24], [43]. Based on this, we set σ_t as a hyperparameter and express it as:

$$\sigma_t = \frac{\gamma_0\sigma_0^2 d}{\min_{k\in A'_t}||\mathbf{w}_{k,t}||^2}, \tag{17}$$

where $||\cdot||$ represents the l_2-norm of vector, and γ_0 represents the received SNR threshold.

However, as previously discussed, solving problem P1 is also challenging due to the implicit nature of the objective function $\mathbb{E}[F(\mathbf{w}_T)]$. To make the problem tractable, a feasible method is finding an approximation of the objective function. And fortunately, Theorem 1 presents a strict upper bound of the convergence speed and shows the convergence speed is mainly determined by the total aggregation number $\sum_{k\in\mathcal{T}}|A'_t|$. Consequently, by leveraging the convergence analysis, we can replace the objective function with its convergence bound, and by eliminating the fixed terms, P1 can be reformulated as follows:

$$\text{P2:} \max_{\{A_t\}_{t\in\mathcal{T}}} \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} A_{k,t}x_{k,t} \tag{18}$$

$$\text{s.t. constraints (14a) and (14b).} \tag{19}$$

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

In the classical over-the-air FL framework, at each communication round, all clients satisfying constraints are selected into A_t to maximize the total aggregation number. However, in the energy-limited network, selecting all clients satisfying constraints into A_t may not be the best choice, especially for the case that A_t includes many dropout clients. The reason is that the dropout clients will waste the energy, reduce the number of participation, and slow down the convergence speed. Intuitively, a better choice is to select the clients that can successfully finish the local training to increase the total aggregation times. However, it is a thorny problem due to the unavailability of local training $x_{k,t}$. Moreover, since $x_{k,t}$ has a lot of uncertainty, we can not make any statistical assumption on it.

At first glance, it may seem impossible to solve the problem under this informal setting. However, there exists an adaptive learning solution called adversarial bandit that can provide a well-defined theoretical performance bound. In adversarial bandit problems, the clients can be considered as base arms, while the selected set A_t is commonly referred to as a super arm, and the aggregation state $x_{k,t}$ corresponds to the reward of each arm. The reward in this context can be considered as a value that is predetermined before selecting, but it remains unknown to the server until the corresponding client is selected. The goal of the adversarial bandit problem is to maximize the cumulative reward, which can be formulated as an optimization problem as follows,

$$\text{P3:} \max_{\{A_t\}_{t\in\mathcal{T}}} \mathbb{E}\left[\sum_{t=0}^{T-1}\sum_{k\in\mathcal{K}} A_{k,t}x_{k,t}\right] \tag{20}$$

$$\text{s.t. constraints (14a) and (14b).} \tag{21}$$

To better analyze, we further let $\mathbb{E}[A_{k,t}] = p_{k,t}$ represent the expected probability of selecting client k at round t. Accordingly, the client selection problem can be further transformed into the probability assignment problem. Namely, P3 can be further formulated as

$$\begin{aligned}
\text{P4:} & \max_{\{\mathbf{p}_t\}_{t\in\mathcal{T}}} \sum_{t=0}^{T-1}\sum_{k\in\mathcal{K}} p_{k,t}x_{k,t} \\
\text{s.t.} & \sum_{t=0}^{T-1} A_{k,t}\left(x_{k,t}E_{k,t}^{suc} + (1-x_{k,t})E_{k,t}^{wat}\right) \le E_k, \quad \forall k,t \\
& A_{k,t} \in \{0,1\}, \quad \forall k,t \\
& p_{k,t} \in [0,1], \quad \forall k,t
\end{aligned} \tag{22}$$

where $\mathbf{p}_t = \{p_{k,t}\}_{k\in\mathcal{K}}$. Observing P4, it is clear that the communication energy is still unknown to the server. Thus, it is necessary to evaluate the client's energy required for completing local training and model transmission. In this paper, we simplify the problem by using the average energy consumption to evaluate the energy required by clients for local training and model transmission. Hence, in the following sections, our focus will be on assigning probabilities to clients.

VI. Solution and Algorithm

A. Exp3 With Multiple Plays and Energy Constraint

The Exp3 algorithm is a well-known method for solving the adversarial bandit problem, and it follows a three-step process in each round. First, it selects an arm k by sampling from the previously calculated selection weight. Then, it estimates the reward based on the observed outcome after selection. Lastly, the estimated reward is utilized to update the selection weight. The most crucial step in this process is the accurate estimation of the reward from the observed data. To ensure that each historical data point accurately reflects the true state of the arm and is not influenced by stochastic selection, Exp3 uses an unbiased estimate of the real reward, denoted by $\hat{x}_{k,t}$, which is expressed as follows:

$$\hat{x}_{k,t} = \frac{A_{k,t}}{p_{k,t}}x_{k,t}. \tag{23}$$

From the above equality, it is easy to see that $\mathbb{E}[\hat{x}_{k,t}] = x_{k,t}$, which effectively illustrates that $\hat{x}_{k,t}$ is an unbiased estimate of $x_{k,t}$. Exp3 utilizes an unbiased estimator to estimate the potential reward of each arm, and then applies exponential weight to reflect this estimation. The exponential weight is expressed as:

$$w_{k,t+1} = w_{k,t}\exp(\lambda_0\hat{x}_{k,t}), \tag{24}$$

where the selection weight $w_{k,t}$ in the Exp3 algorithm represents the importance of selecting arm k at round t, and it is computed based on the previous reward estimates. The parameter $\lambda_0 \in (0,1]$ is a tuning parameter balancing exploration and exploitation. When λ_0 is small, the algorithm tends to explore more, while a larger value of λ_0 leads to a more exploitation-oriented strategy. According to the selection weight, the selection probability can be updated as

$$p_{k,t} = \frac{w_{k,t}}{\sum_{k\in\mathcal{K}} w_{k,t}}. \tag{25}$$

Note that in the canonical Exp3 algorithm, only one arm is selected in each round, which makes it important to ensure that the probability distribution of arm selection sums up to 1.

In the canonical Exp3, since it just select an arm in each round, we need to make some modifications to the canonical Exp3 algorithm to make it suitable for our context. Thus, for the weight update, we modify it as,

$$w_{k,t+1} = \begin{cases} w_{k,t}\exp\left(\frac{\lambda_0 m_t}{|A_{ctr,t}|}\left(\hat{x}_{k,t} - e_{k,t}\right)\right), & k \notin S_t \cup A_{nctr,t} \\ w_{k,t}, \quad k \in S_t \text{ and } k \notin A_{nctr,t} \\ 0, \quad k \in A_{nctr,t} \end{cases} \tag{26}$$

where m_t represents the selection number at communication round t; $e_{k,t} = E_{k,t} - \frac{E_k}{T}$ servers as a penalty term to represent the energy consumption, with $E_{k,t} = x_{k,i}E_{k,i}^{suc} + (1 - x_{k,i})E_{k,i}^{wat}$; A_{ctr} and A_{nctr} respectively represent the set of clients satisfying and not satisfying the energy constraint; and S_t represents the set of clients whose selection probability exceeds the pre-defined upper bound during the probability allocation stage in communication round t.

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

Given the selection weight, we modify the probability allocation into a form that can be applied to multiple plays. Formally, we denote it as

$$p_{k,t} = \frac{m_t w_{k,t}}{\sum_{k \in \mathcal{K}} w_{k,t}}, \tag{27}$$

where m_t represents the number of selected clients. From the equations above, it is evident how the client's training state and energy exert influence. Specifically, when the local training state $x_{k,t}$ remains consistently equal to 1, it amplifies the selection weight, consequently raising the probability of being chosen. Similarly, a relatively high remaining energy ratio enhances both the selection weight and probability.

B. Algorithm

To provide a clear presentation for our proposed solution for client selection in problem P4, we present a detailed solution in Algorithm 1. For the probability allocation, the detailed procedures have been presented in Algorithm 2. In Algorithm 2, it should be noted that the allocated probability $p_{k,t}$ may exceed 1 due to an excessively large exponential weight, resulting in probability overflow. To address this issue, we use a capping-based technique, whereby $p_{k,t}$ is capped to $\min\{p_{k,t}, 1\}$ to limit the selection probability of "overflowed" clients to 1 (line 7-8 in Algorithm 2). Moreover, for the number of client selection, we define it as a dynamic value and set it as $m_t = \lceil |A_{ctr,t}| * r_s \rceil$, where r_s represents the selection ratio. To ensure the quality of model aggregation, we further set $m_t = \text{len}(A_{ctr,t})$ when m_t is less than the selection threshold m_0.

For the client selection, we apply a stochastic selection method similar to [44] to select the clients based on the allocated probability. The detailed procedure has been presented in Algorithm 3. Specifically, we continuously update the clients' probability based on (28) and select the clients based on this probability. It is worth noting that although we use (28) to update the client's selection probability, its expected probability remains unchanged. Its proof is as follows,

$$(p_i + \alpha) \times \frac{\beta}{\alpha + \beta} + (p_i - \beta) \times \frac{\alpha}{\alpha + \beta} = p_i, \tag{29}$$

$$(p_j - \alpha) \times \frac{\beta}{\alpha + \beta} + (p_j + \beta) \times \frac{\alpha}{\alpha + \beta} = p_j. \tag{30}$$

Overall, these algorithms and procedures provide a comprehensive and effective solution for selecting clients and allocating probabilities, while also accounting for potential probability overflow.

C. Theoretical Analysis

To prove the effectiveness of our proposed solution, we present a theoretical regret guarantee in this subsection. Before the formal proof, we first introduce the following definition.

Definition 1: Let $p^*_{k,t}$ represent the optimal allocation probability for client k at communication round t, we define the approximate expected cumulative reward of the optimal strategy in T communication rounds as

$$\sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \frac{p^*_{k,t}}{|A_{ctr,t}|} x_{k,t}. \tag{31}$$

Definition 2: Given the optimal expected cumulative reward, the regret of DCSE can be represented as

$$R_T = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \frac{p^*_{k,t}}{|A_{ctr,t}|} x_{k,t} - \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \frac{p_{k,t}}{|A_{ctr,t}|} x_{k,t}. \tag{32}$$

Algorithm 1: Dynamic Client Selection Based on Exp3 (DCSE).

Input: initialize the global model $\mathbf{w}_0$ and the selection weight $w_{k,0} = 1$ for $k \in \mathcal{K}$.
Output: the final global model $\mathbf{w}_T$.

for $t = 0, \ldots, T-1$ **do**
  the server acquires channel gains $\{h_{k,t}\}_{k \in \mathcal{K}}$ and other information.
  the server filters clients satisfying energy constraints into the set $A_{ctr,t}$ based on the remained energy of each client.
  the server calculates selection probability vector, i.e., $\mathbf{p}_t$, S_t =ProbAlloc($\{w_{k,t}\}_{k \in \mathcal{T}}$, $A_{ctr,t}$, m_t).
  the server selects clients based on the selection probability vector, i.e., A_t =CleSec($\mathbf{p}_t$).
  the selected clients perform local training and return the training states and transmission information.
  the server adjusts the transmit power according to returned information.
  the clients completing the local training transmit the updated model, and the server receives and post-processes the aggregated signal.
  the server updates the unbiased estimate of the training state based on (23) and the selection weight based on (26).
end for
Return $\mathbf{w}_T$.

Algorithm 2: Probability Allocation (ProbAlloc).

Input: the selection weight $\{w_{k,t}\}$, the set of clients satisfying constraints $A_{ctr,t}$ and the selection number m_t.
Output: the selection probability vector at round $\mathbf{p}_t$ and the overflowed set at round S_t.

1: **if** $m_t < m_0$ **then**
2:   $m_t = \text{len}(A_{ctr,t})$
3: **end if**
4: $w_t = \sum_{k \in \mathcal{K}} w_{k,t}$
5: **for** $k \in A_{ctr,t}$ **do**
6:   $p_{k,t} = \frac{m_t w_{k,t}}{w_t}$
7:   **if** $p_{k,t} > 1$ **then**
8:     $p_{k,t} = 1$ and $S_t = S_t \cup k$
9:   **end if**
10: **end for**
11: Return $\mathbf{p}_t$, S_t.

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3: Client Selection (CleSec).

Input: the probability vector at round $\mathbf{p}_t$.
Output: the select set A_t.
1: **while** len($\{0 < p_i < 1\}_{i \in A_{ctr,t}}) > 0$ **do**
2: **if** len($\{0 < p_i < 1\}_{i \in A_{ctr,t}}) = 1$ **then**
3: round $p_{i,t}$ to the nearest integer, i.e., if $p_{i,t} \geq 0.5$, we let $p_i = 1$; otherwise, $p_i = 0$.
4: **else**
5: randomly select two clients i and j, and let $\alpha = \min\{1 - p_i, p_j\}$ and $\beta = \min\{p_i, 1 - p_j\}$.
6: update p_i and p_j.

$$(p_i, p_j) = \begin{cases} (p_i + \alpha, p_j - \alpha) & \text{with probability } \frac{\beta}{\alpha+\beta}, \\ (p_i - \beta, p_j + \beta) & \text{with probability } \frac{\alpha}{\alpha+\beta} \end{cases} \tag{28}$$

7: **end if**
8: **end while**
9: $A_t = \{i : p_i = 1\}_{i \in A_{ctr,t}}$
10: Return A_t.

TABLE II
EXPERIMENTAL PARAMETERS SETTINGS

Setting	Symbols	Task1	Task2
Dataset	-	EMNIST-Letter	CIFAR-10
Number of local epoches	NLE	4	5
Batch size	b	30	64
Bits per sample	BPS	$28*28*1*8$	$32*32*3*8$
Cycles per bit	CPB	400	400
Initial learning rate	η_0	0.05	0.02
-	E_{low}	5	50
-	E_{upp}	80	300
Selection threshold	m_0	10	10
Total rounds	T	100	300

Intuitively, the regret measures the deviation of the proposed algorithm's performance from the optimal algorithm's performance. Therefore, it provides a useful metric for evaluating the effectiveness of the algorithm. Based on Definition 1 and Definition 2, we can present the following theorem that establishes an upper bound on the regret.

Theorem 2: The regret of DCSE can be bounded as

$$R_T \leq K \lambda_0 r_s \sum_{t \in \mathcal{T}} \frac{1}{|A_{ctr,t}|} + \frac{\ln K}{\lambda_0}, \tag{33}$$

and if the learning rate $\lambda_0 = \sqrt{\frac{\ln K \sum_{t=0}^{T-1} |A_{ctr,t}|}{K r_s}}$, we have

$$R_T \leq \sqrt{\frac{K \ln K r_s}{\sum_{t=0}^{T-1} |A_{ctr,t}|}}. \tag{34}$$

Proof: The complete proof has been presented in the supplement file. ∎

Theorem 2 demonstrates that the proposed solution maintains a constant bound on performance compared to the optimal solution. It indicates that our algorithm can converge to the optimal point within T rounds. Particularly, in favorable scenario where clients consistently meet energy constraints, the regret can be reduced to $R_t \leq \sqrt{\frac{\ln K r_s}{T}}$.

Further, upon closer analysis of the time complexity of proposed algorithm, it is evident that the time complexity of the proposed algorithms is less than $\mathcal{O}(KT)$. Specifically, in Algorithm 2, since it requires allocating probabilities for each client satisfying the energy consumption constraints, it needs to perform calculations for up to $|A_{ctr}| \leq K$ times. Similarly, in Algorithm 3, as it performs a maximum of $|A_{ctr}|$ updates for selection, it also requires computations for up to $|A_{ctr}| \leq K$ times. Therefore, the overall time complexity of the algorithm does not exceed $\mathcal{O}(KT)$.

VII. EXPERIMENTS

In this section, we verify the performance with two non-convex models to show that the analytical results and the proposed algorithm can perform well even when some assumptions (like strong convexity) are violated.

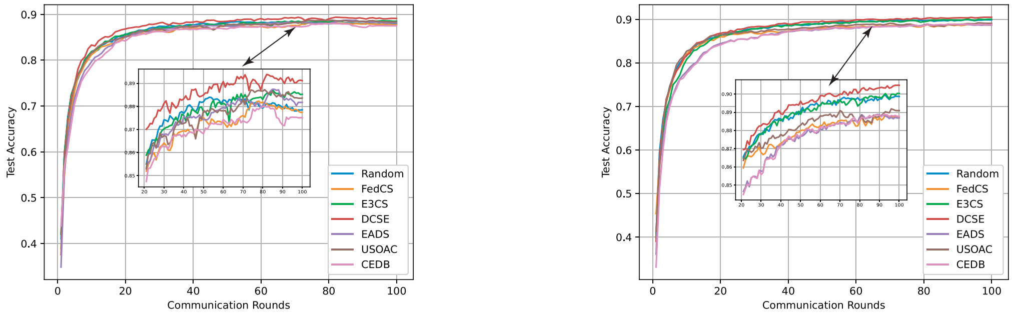
A. Basic Setup

1) Environment and Tasks: We perform experiments in a typical IoT network setup with $K = 100$ clients for federated learning. To evaluate the proposed algorithm's performance, we conduct two image classification tasks on two datasets: EMNIST-Letter [45] and CIFAR-10 [46]. For both training tasks, their models consist of convolution layers, max-pooling layers, fully connected layers, and a softmax output layer. Specifically, for the EMNIST-Letter task, the sizes of different layers are $2*5\times5$, 2×2, 2 (respectively 1280 and 256 units), 1 in sequence, resulting in a total of 339898 parameters. For the CIFAR-10 task, the sizes of different layers are $2*5\times5$, 3×3, 2 (respectively 384 and 192 units), 1 in sequence, resulting in a total of 576778 parameters. For the learning rate at different communication rounds, we set it as $\eta_t = \eta_0 * (1 - t/(T + 1/t))$.

2) Simulation of Volatile Client:

- *Client set*: We consider there are 100 clients participating in the training throughout the entire training process, and the maximum energy of each client is randomly set within the range of E_{low} and E_{upp}.
- *Client data*: To generate non-independent and identically distributed data, we adopt the same approach as in [39], where the data distribution on each client follows a Dirichlet distribution with an extent parameter γ to control the clients' data distribution, where a smaller γ means a more uneven distribution of data on the client side. In this paper, we prove the performance achieved by the proposed algorithm when $\gamma = 0.5$, $\gamma = 1$, and $\gamma = 5$.
- *Local training*: To model the clients' resource differences, we randomly set the computation power $P_{k,t}^{cmp}$ of each client k between 0.3 to 1 Watt, and the CPU frequency $f_{k,t}$ between 0.5 to 1.5 GHz. To simulate the completion status of local training, we assume that the success rate of each client k follows a Bernoulli distribution Bern(ρ_k). Here, ρ_k denotes the probability of client completing the local training. To simulate the system difference, we set ρ_k to be a random value from set [0.3,0.5,0.7]. In the case of local training failure, we randomly generate a value between

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.



(a) $\beta_t = 1$ (b) $\beta_t = 5$

Fig. 2. Impact of channel gain on training performance.

0 and 1 to simulate the training progress $b_{k,t}$ before the dropout.

- *Model transmission and aggregation*: To simulate the time-varying nature of client's transmission environment, we assume that the wireless channel follows Rayleigh fading with scale parameter β_t. For the bandwidth and the variance of channel noise, we set it to $B = 20$ MHz and $\sigma_0^2 = 10^{-6}$, respectively. Other training settings for different tasks have been presented in Table II.

3) Baseline: In this paper, the following baselines for client selection are prepared to compare performance.

- *Random*: Random is the most commonly employed scheme in FL client selection. Its primary advantage lies in its robustness, making it applicable to a wide range of scenarios [6].
- *FedCS*: FedCS, proposed by Nishio et al. [11], focuses on selecting faster clients for participation in FL training. To tailor it to our context, we made minor revisions to the algorithm. Specifically, in our scenario, FedCS consistently opts for clients with the highest average success ratio in aggregation.
- *E3CS*: E3CS is a client selection method based on Exp3 proposed by Huang et al. [29], but it does not consider energy constraints. In this paper, we compare it to underscore the significance of integrating energy constraints into algorithm design.
- *CEDB*: CEDB, proposed by Chen et al. [47], is a recently proposed client selection scheme applied in AirComp FL. Specifically, they designed a CED coefficient $q_{k,t} = (A_{k,t}|\mathcal{D}_k|^2)/(E_{k,t}^{suc}h_{k,t}^2)$ to evaluate and select clients. To tailor it to our context, we also make some minor revisions to algorithm. Specifically, as $E_{k,t}^{suc}$ cannot be known in advance, we utilize historical data to calculate the average of $E_{k,t}^{suc}$ as a substitute for $E_{k,t}^{suc}$, subsequently selecting clients with the smallest $q_{k,t}$.
- *USOAC*: USOAC is a user scheduling selection method based on channel gain, as introduced by Ma et al. [48]. Given that the original scheme did not consider the dynamics of training volatility, we made some minor adjustments to align it with our context. More precisely, we enhanced the original indicator by multiplying it with the average probability of successful integration to evaluate clients.

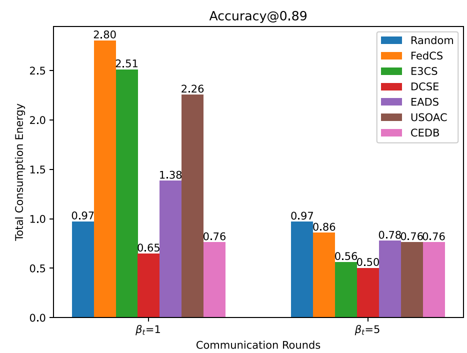


Fig. 3. Impact of channel gain on energy consumption.

The selection process involves choosing clients with the highest resulting value.

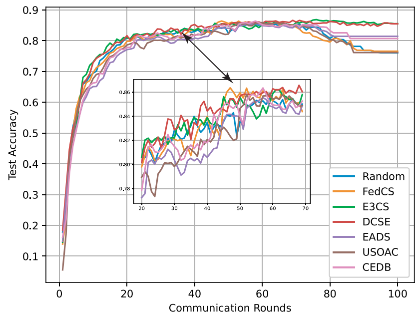
- *EADS*: EADS, proposed by Sun et al. [24], is a dynamic scheduling for over-the-air federated learning with energy constraints. Specifically, employing Lyapunov optimization, they construct a virtual energy sequence to indicate the current energy deficit, facilitating real-time decision-making. Similar methodologies have been adopted in prior works [25], [26].

B. Experimental Results

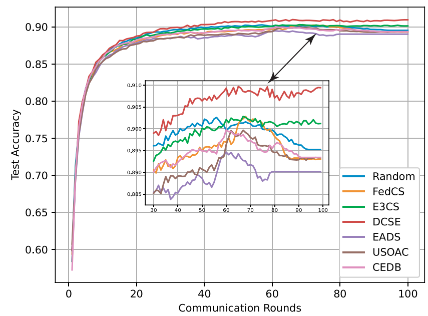
1) Impact of Channel Gain: By conducting 100 communication rounds and setting $m_t = 0.4$ in the simulation, we assess the impact of channel gain on test accuracy and total energy consumption. As depicted in Figs. 2 and 3, the proposed algorithm DCSE outperforms other algorithms, both when $\beta_t = 1$ and $\beta_t = 5$. Although the performance of other algorithms gradually approaches that of DCSE as the channel gain increases, it is evident that DCSE exhibits the lowest energy consumption. Furthermore, we observe a common trend wherein test accuracy increases as channel gain rises, while energy consumption decreases. This can be attributed to the improvement in channel gain, signifying better transmission conditions, thereby reducing the required transmit power and energy consumption, leading to an increase in the aggregation number.

2) Extreme Scenarios: To demonstrate the robustness of the proposed algorithm, we evaluate its performance along with

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.



(a) Bad scenario



(b) Good scenario

Fig. 4. Performance of different algorithms in two extreme scenarios.

TABLE III
PERFORMANCE EVALUATION FOR EMNIST-LETTER AND CIFAR-10

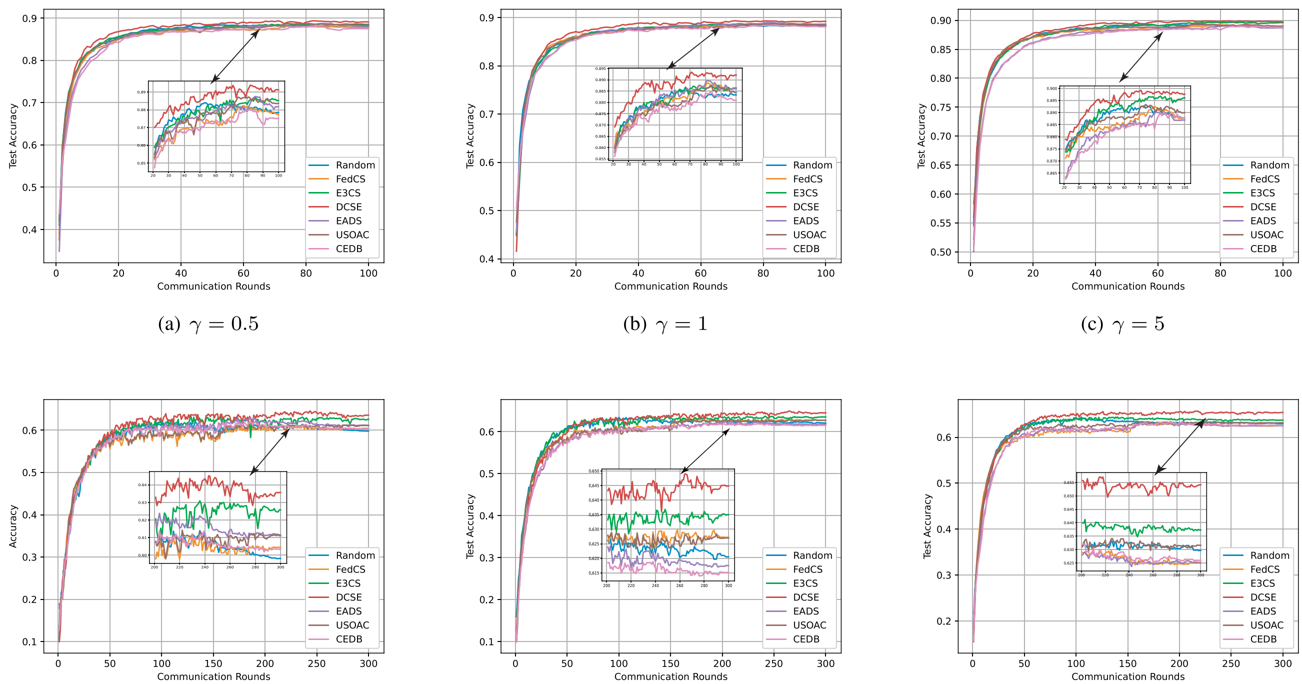
Task 1	Selection	Accuracy @**82**			Accuracy @**85**			Accuracy @**89**			Best Accuracy (%)		
	Schemes	$\gamma=0.5$	$\gamma=1$	$\gamma=5$	$\gamma=0.5$	$\gamma=1$	$\gamma=5$	$\gamma=0.5$	$\gamma=1$	$\gamma=5$	$\gamma=0.5$	$\gamma=1$	$\gamma=5$
EMNIST-Letter	Random	11	9	7	17	15	11	NaN	NaN	45	88.40	88.58	89.28
	FedCS	11	**8**	**6**	19	14	11	NaN	NaN	75	88.23	88.87	89.25
	E3CS	10	10	7	16	16	12	NaN	NaN	42	88.72	88.75	89.68
	DCSE	**8**	**8**	**6**	**13**	**12**	11	**59**	**60**	**34**	**89.38**	**89.34**	**89.92**
	EADS	13	11	9	20	16	16	NaN	NaN	80	88.75	88.99	89.02
	USOAC	10	9	7	19	15	**10**	NaN	NaN	69	88.72	88.74	89.34
	CEDB	14	11	10	21	17	16	NaN	NaN	NaN	88.01	88.46	88.99
Task 2	Selection	Accuracy @**58**			Accuracy @**62**			Accuracy @**64**			Best Accuracy (%)		
	Schemes	$\gamma=0.5$	$\gamma=1$	$\gamma=5$	$\gamma=0.5$	$\gamma=1$	$\gamma=5$	$\gamma=0.5$	$\gamma=1$	$\gamma=5$	$\gamma=0.5$	$\gamma=1$	$\gamma=5$
CIFAR-10	Random	47	33	**25**	NaN	66	47	NaN	NaN	97	61.93	63.40	64.43
	FedCS	54	42	28	NaN	155	153	NaN	NaN	NaN	61.21	63.14	63.46
	E3CS	44	**32**	27	82	**56**	**45**	NaN	NaN	98	63.65	63.68	64.43
	DCSE	**43**	41	26	**69**	62	**45**	**209**	**152**	**62**	**64.52**	**64.89**	**65.70**
	EADS	49	51	31	158	181	92	NaN	NaN	NaN	62.97	62.42	63.41
	USOAC	53	42	26	NaN	150	54	NaN	NaN	NaN	61.84	62.97	63.41
	CEDB	52	52	31	NaN	NaN	66	NaN	NaN	NaN	61.95	61.89	63.27

other algorithms in two extreme scenarios. In the first scenario, termed the "very bad scenario," we assume each client has 10 J of energy, a success rate of 0.5, a data distribution parameter $\gamma = 0.1$, and $m_t = 0.3$. Conversely, in the "very good scenario," we assume each client has 10 J of energy, a success rate of 0.8, a data distribution parameter $\gamma = 10$, and $m_t = 0.4$. As depicted in Fig. 4(a), we observe a decrease in performance for some algorithms as energy consumption decreases. Moreover, while the proposed algorithm demonstrates certain advantages in early and mid-term training, its final performance is comparable to that of E3CS. We attribute this to the uneven distribution of client data and a lower probability of successful training, resulting in increased energy consumption. Additionally, if clients have insufficient energy, for any selection method, their energy will be depleted, leading to a smaller performance gap.

3) Real Training on Public Datasets: In order to facilitate a more comprehensive comparison of the training performance, we conducted 100 and 300 communication rounds, respectively, for Task 1 and Task 2, to investigate the effects of different selection schemes on key performance indicators, including model accuracy, convergence speed, and energy consumption. The corresponding results are presented in Table III and Figs. 5–6. Based on these results, we made some interesting observations, which are summarized as follows:

- *Impact of data distribution:* As depicted in Fig. 5, all algorithms exhibit better performance when the data distribution is relatively uniform. Additionally, it's evident that the performance gap between different algorithms widens as γ decreases, whereas it narrows as γ increases. This indicates that even with a fixed selection scheme, training performance is influenced by the data distribution among clients. Uneven data distribution, such as $\gamma = 0.5$, results in poorer training performance.
- *Effectiveness of DCSE:* After examining the results presented in Fig. 5, we can observe that the proposed algorithm DCSE achieves better performance than other algorithms. During the initial training process, there is no significant performance difference among the different selection schemes since the remained energy is sufficient. However, as the wasted energy varies over the course of the training process, the performance gap between the different solutions gradually widens. Especially in the middle and late stages of training, the more complex the model, the more obvious the gap.. In addition, from the results presented in Fig. 6, we can also observe that our scheme consumes the least amount of energy. Thus, our scheme not only achieves improved training performance but also saves energy in a simulated environment.

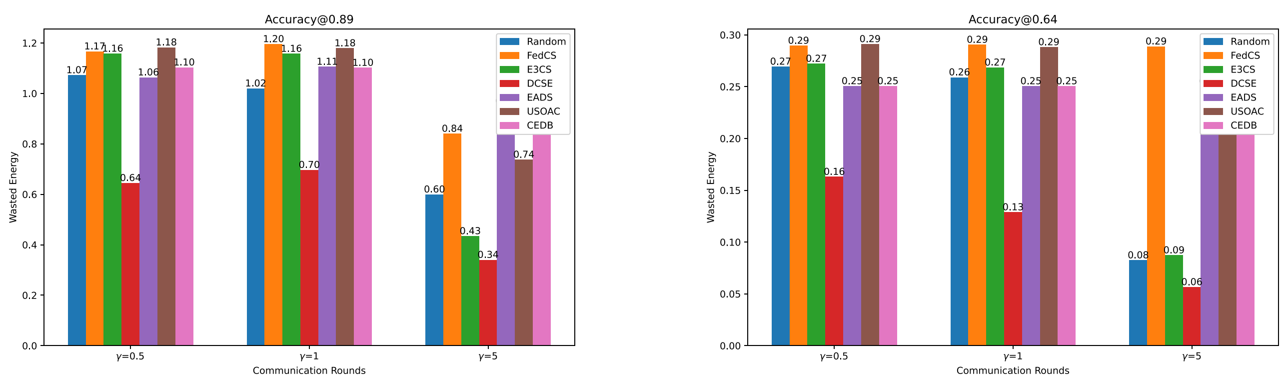
Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.



(a) $\gamma = 0.5$ (b) $\gamma = 1$ (c) $\gamma = 5$

(d) $\gamma = 0.5$ (e) $\gamma = 1$ (f) $\gamma = 5$

Fig. 5. Test accuracy versus communication rounds for EMNIST-Letter ((a),(b),(c)) and CIFAR-10 ((d),(e),(f)).



(a) EMNIST-Letter (b) CIFAR-10

Fig. 6. The wasted energy (w/h) consumed in Task 1 and Task 2 when using the different schemes.

- *Comparison of model accuracy:* In Fig. 5, we track the model accuracy of different selection strategies over communication rounds. As illustrated, our algorithm consistently outperforms other algorithms in terms of achieved accuracy. For instance, in task 1, the highest accuracy attained by DCSE increases by up to 1.6% compared to CEDB when $\gamma = 0.5$. In task 2, the best accuracy achieved by DCSE increases by 5.41% and 1.4% respectively compared to FedCS and E3CS when $\gamma = 0.5$. This clearly demonstrates that the average successful training rate does not directly correlate with training performance. Furthermore, while E3CS may achieve suboptimal performance, it fails to further enhance performance or reduce wasted energy due to its lack of consideration for the impact of energy consumption on training performance.
- *Comparison of convergence speed:* We present a summary of the convergence speed and best accuracy achieved by different algorithms in Table III. To account for the local training instability, we use the data under Accuracy@*number* to indicate the communication round at which a target accuracy is achieved for the first time. Upon inspecting Table III, we can observe that the performance achieved by different schemes is comparable in the initial training phase, especially when the data distribution is relatively uniform (i.e., $\gamma = 5$). However, as

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

the training process progresses and the performance improves, the differences between other algorithms become more pronounced. In particular, for the more challenging task 2, all other algorithms failed to achieve the target accuracy of 0.64 throughout the training process when $\gamma = 0.5$ and $\gamma = 1$. On the other hand, our proposed DCSE scheme achieves the target accuracy of 0.64 with a faster convergence speed 38.71% E3CS algorithm when $\gamma = 5$. These results illustrate the effectiveness of our scheme in accelerating model training.

- *Comparison of wasted energy:* Fig. 6 illustrates the wasted energy in task 1 and task 2. As shown, our algorithm has the lowest waste of energy, while the wasted energy of E3CS is slightly lower than that other algorithms. Overall, the order of wasted energy is consistent with test accuracy. Moreover, our findings demonstrate that DCSE is the most energy-efficient scheme for clients compared to other algorithms. Specifically, in task 1, when $\gamma = 0.5$, the wasted energy of E3CS and EADS is respectively 81.25% and 65.62% higher than that of DCSE. These results provide strong evidence for the energy-saving capability of our proposed scheme.

VIII. Conclusion

This paper investigated an over-the-air federated learning system with volatile clients, which have limited energy and may unexpectedly drop out during local training. The research focuses on the impact of such volatile clients on the model training process and convergence performance. The study revealed that maximizing the aggregation number can boost the FL training. Building on this finding, we proposed DCSE, a dynamic client selection strategy based on Exp3 with multiple plays and energy constraint, to allocate the selection probability of each client. Theoretical analysis has proven that DCSE is theoretically feasible as its regret has a strictly upper bound. Our extensive experiments further demonstrate that DCSE can significantly enhance FL training by accelerating the convergence speed, enhancing model accuracy, and reducing wasted energy.

References

[1] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[2] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, “Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing,” *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 9–39, Jan. 2023.

[3] J. Chen and X. Ran, “Deep learning with edge computing: A review,” in *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[4] G. Shi, Y. Xiao, Y. Li, and X. Xie, “From semantic communication to semantic-aware networking: Model, architecture, and open problems,” *IEEE Commun. Mag.*, vol. 59, no. 8, pp. 44–50, Aug. 2021.

[5] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[7] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-efficient sgd via gradient quantization and encoding,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1707–1718.

[8] J. Wangni, J. Wang, J. Liu, and T. Zhang, “Gradient sparsification for communication-efficient distributed optimization,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018.

[9] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, “FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization,” in *Proc. Int. Conf. Artif. Intell. Statist.*, PMLR, 2020, pp. 2021–2031.

[10] Z. Qu, K. Lin, J. Kalagnanam, Z. Li, J. Zhou, and Z. Zhou, “Federated learning’s blessing: Fedavg has linear speedup,” 2020, *arXiv: 2007.05690*.

[11] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.

[12] J. Xu and H. Wang, “Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2021.

[13] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, “An efficiency-boosting client selection scheme for federated learning with fairness guarantee,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.

[14] S. Zou, M. Xiao, Y. Xu, B. An, and J. Zheng, “FedDCS: Federated learning framework based on dynamic client selection,” in *Proc. IEEE 18th Int. Conf. Mobile Ad Hoc Smart Syst.*, 2021, pp. 627–632.

[15] G. Zhu, J. Xu, K. Huang, and S. Cui, “Over-the-air computing for wireless data aggregation in massive IoT,” *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 57–65, Aug. 2021.

[16] D. Wen, G. Zhu, and K. Huang, “Reduced-dimension design of MIMO over-the-air computing for data aggregation in clustered IoT networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5255–5268, Nov. 2019.

[17] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, “Over-the-air computation for IoT networks: Computing multiple functions with antenna arrays,” *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5296–5306, Dec. 2018.

[18] X. Cao, G. Zhu, J. Xu, and K. Huang, “Optimized power control for over-the-air computation in fading channels,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7498–7513, Nov. 2020.

[19] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, 2020.

[20] W. Guo, R. Li, C. Huang, X. Qin, K. Shen, and W. Zhang, “Joint device selection and power control for wireless federated learning,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2395–2410, Aug. 2022.

[21] L. L. d. Oliveira, G. H. Eisenkraemer, E. A. Carara, J. B. Martins, and J. Monteiro, “Mobile localization techniques for wireless sensor networks: Survey and recommendations,” *ACM Trans. Sensor Netw.*, vol. 19, no. 2, pp. 1–39, 2023.

[22] M. N. Hussain, M. A. Halim, M. Y. A. Khan, S. Ibrahim, and A. Haque, “A comprehensive review on techniques and challenges of energy harvesting from distributed renewable energy sources for wireless sensor networks,” *Control Syst. Optim. Lett.*, vol. 2, no. 1, pp. 15–22, 2024.

[23] R. Priyadarshi, “Exploring machine learning solutions for overcoming challenges in IoT-based wireless sensor network routing: A comprehensive review,” *Wireless Netw.*, vol. 30, pp. 2647–2673, 2024.

[24] Y. Sun, S. Zhou, Z. Niu, and D. Gündüz, “Dynamic scheduling for over-the-air federated edge learning with energy constraints,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 227–242, Jan. 2022.

[25] C. Battiloro, P. Di Lorenzo, M. Merluzzi, and S. Barbarossa, “Lyapunov-based optimization of edge resources for energy-efficient adaptive federated learning,” *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 265–280, Mar. 2023.

[26] Q. An, Y. Zhou, Z. Wang, H. Shan, Y. Shi, and M. Bennis, “Online optimization for over-the-air federated learning with energy harvesting,” *IEEE Trans. Wireless Commun.*, vol. 23, no. 7, pp. 7291–7306, Jul. 2024.

[27] H. Wang and J. Xu, “Friends to help: Saving federated learning from client dropout,” 2022, *arXiv:2205.13222*.

[28] K. Pillutla, S. M. Kakade, and Z. Harchaoui, “Robust aggregation for federated learning,” *IEEE Trans. Signal Process.*, vol. 70, pp. 1142–1154, 2022.

[29] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya, “Stochastic client selection for federated learning with volatile clients,” *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20055–20070, Oct. 2022.

[30] N. Zhang and M. Tao, “Gradient statistics aware power control for over-the-air federated learning,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5115–5128, Aug. 2021.

[31] H. Yang, P. Qiu, J. Liu, and A. Yener, “Over-the-air federated learning with joint adaptive computation and power control,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2022, pp. 1259–1264.

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.

[32] W. Ni, Y. Liu, Y. C. Eldar, Z. Yang, and H. Tian, "Star-RIS integrated nonorthogonal multiple access and over-the-air federated learning: Framework, analysis, and optimization," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17136–17156, Sep. 2022.

[33] W. Ni, Y. Liu, Z. Yang, H. Tian, and X. Shen, "Integrating over-the-air federated learning and non-orthogonal multiple access: What role can RIS play?," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10083–10099, Dec. 2022.

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

[35] Y. Hu et al., "Energy minimization for federated learning with irs-assisted over-the-air computation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3105–3109.

[36] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2020, pp. 1–6.

[37] W. Xia, W. Wen, K.-K. Wong, T. Q. Quek, J. Zhang, and H. Zhu, "Federated-learning-based client scheduling for low-latency wireless communications," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 32–38, Apr. 2021.

[38] F. Shi, C. Hu, W. Lin, L. Fan, T. Huang, and W. Wu, "Vfedcs: Optimizing client selection for volatile federated learning," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 24995–25010, Dec. 2022.

[39] F. Shi, W. Lin, L. Fan, X. Lai, and X. Wang, "Efficient client selection based on contextual combinatorial multi-arm bandits," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5265–5277, Aug. 2023.

[40] X. An, R. Fan, S. Zuo, H. Hu, H. Jiang, and N. Zhang, "Joint power control and data size selection for over-the-air computation aided federated learning," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14031–14046, Apr. 2024.

[41] Y. Sun, S. Zhou, and D. Gündüz, "Energy-aware analog aggregation for federated learning with redundant data," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–7.

[42] J. Konečny, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[43] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.

[44] W. Lin, Y. Xu, B. Liu, D. Li, T. Huang, and F. Shi, "Contribution-based federated learning client selection," *Int. J. Intell. Syst.*, vol. 37, no. 10, pp. 7235–7260, 2022.

[45] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending mnist to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 2921–2926.

[46] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," pp. 1–60, 2009.

[47] C. Chen, Y.-H. Chiang, H. Lin, J. C. Lui, and Y. Ji, "Joint client selection and receive beamforming for over-the-air federated learning with energy harvesting," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1127–1140, 2023.

[48] X. Ma, H. Sun, Q. Wang, and R. Q. Hu, "User scheduling for federated learning through over-the-air computation," in *Proc. IEEE 94th Veh. Technol. Conf.*, 2021, pp. 1–5.

Fang Shi received the MS degree from Guangzhou University, in 2019, and the PhD degree from the South China University of Technology, in 2023. She is now an associate professor with the College of Mathematics and Informatics (Software Enginering), South China Agricultural University, Guangzhou, China. Her research interests focus on federated learning, distributed systems, edge intelligence and wireless cooperative communications.

Weiwei Lin (Senior Member, IEEE) received the BS and MS degrees from Nanchang University, in 2001 and 2004, respectively, and the PhD degree in computer application from the South China University of Technology, in 2007. He has been a visiting scholar with Clemson University from 2016 to 2017. Currently, he is a professor with the School of Computer Science and Engineering at South China University of Technology. His research interests include distributed systems, cloud computing, and AI application technologies. He has published more than 150 papers in refereed journals and conference proceedings. He has been a reviewer for many international journals, including *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Computers*, *IEEE Transactions on Cybernetics*, etc. He is a distinguished member of CCF.

Xiumin Wang (Member, IEEE) received the BS degree from the Department of Computer Science, Anhui Normal University, China, in 2006, and the PhD degree from both the Department of Computer Science, University of Science and Technology of China, Hefei, China, and the Department of Computer Science, City University of Hong Kong, under a joint PhD program. She is now an associate professor with the School of Computer Science and Engineering, South China University of Technology, China. Her research interests include mobile computing, algorithm design and optimization.

Keqin Li (Fellow, IEEE) is a SUNY distinguished professor of computer science with the State University of New York. He is also a distinguished professor with Hunan University, China. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber-physical systems, heterogeneous computing systems, Big Data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has published more than 740 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*.

Albert Y. Zomaya (Fellow, IEEE) is currently the chair professor of High Performance Computing & Networking with the School of Information Technologies, The University of Sydney. He is also the director in the Centre for Distributed and High Performance Computing, which was established in late 2009. He published more than 500 scientific papers and articles and is an author, co-author or editor of more than 20 books. He served as the editor-in-chief of *IEEE Transactions on Computers* (2011–2014). He serves as an associate editor for 22 leading journals, such as, the *ACM Computing Surveys*, *IEEE Transactions on Computational Social Systems*, *IEEE Transactions on Cloud Computing*, and *Journal of Parallel and Distributed Computing*. He delivered more than 150 keynote addresses, invited seminars, and media briefings and has been actively involved, in a variety of capacities, in the organization of more than 600 national and international conferences. He received the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014). He is a chartered engineer, a fellow of AAAS, and IET (United Kingdom). His research interests include areas of parallel and distributed computing and complex systems.

Authorized licensed use limited to: SOUTH CHINA UNIVERSITY OF TECHNOLOGY. Downloaded on February 20,2025 at 11:40:59 UTC from IEEE Xplore. Restrictions apply.