Fair Selection of Edge Nodes to Participate in Clustered Federated Multitask Learning

Abdullatif Albaseer, *Member, IEEE,* Mohamed Abdallah, *Senior Member, IEEE,* Ala Al-Fuqaha, *Senior Member, IEEE,* Abegaz Mohammed, *Member, IEEE,* Aiman Erbad, *Senior Member, IEEE,* Octavia A. Dobre, *Fellow, IEEE*

Abstract—Clustered federated Multitask learning is introduced as an efficient technique when data is unbalanced and distributed amongst clients in a non-independent and identically distributed manner. While a similarity metric can provide client groups with specialized models according to their data distribution, this process can be time-consuming because the server needs to capture all data distribution first from all clients to perform the correct clustering. Due to resource and time constraints at the network edge, only a fraction of devices is selected every round, necessitating the need for an efficient scheduling technique to address these issues. Thus, this paper introduces a two-phased client selection and scheduling approach to improve the convergence speed while capturing all data distributions. This approach ensures correct clustering and fairness between clients by leveraging bandwidth reuse for participants spent a longer time training their models and exploiting the heterogeneity in the devices to schedule the participants according to their delay. The server then performs the clustering depending on predetermined thresholds and stopping criteria. When a specified cluster approximates a stopping point, the server employs a greedy selection for that cluster by picking the devices with lower delay and better resources. The convergence analysis is provided, showing the relationship between the proposed scheduling approach and the convergence rate of the specialized models to obtain convergence bounds under non-i.i.d. data distribution. We carry out extensive simulations, and the results demonstrate that the proposed algorithms reduce training time and improve the convergence speed by up to 50% while equipping every user with a customized model tailored to its data distribution.

Index Terms—Distributed Learning, CFL, participants scheduling, resource allocation, non-i.i.d. and incongruent data distribution.

I. INTRODUCTION

Internet-of-Things (IoT) devices on wireless edge networks generate a vast amount of heterogeneous data that could be utilized to interpret the current behavior of the system or anticipate its future states.

Mobile Edge Computing (MEC) takes advantage of having edge devices and access points equipped with unrivaled capabilities to conduct complicated tasks and support intelligent services closest to these devices [2], [3]. On MEC, *Artificial Intelligence* (AI), and *Machine Learning* (ML), in particular,

Abdullatif Albaseer, Mohamed Abdallah, Ala Al-Fuqaha, Abegaz Mohammed, and Aiman Erbad are with the Division of Information and Computing Technology, College of Science and Engineering, Doha, Qatar (e-mail:{aalbaseer, moabdallah, aalfuqaha,mabegaz, aerbad}@hbku.edu.qa).

Octavia A. Dobre is with the Faculty of Engineering and Applied Science, Memorial University, Canada. (e-mail: odobre@mun.ca).

* Preliminary results in this work are presented at the IEEE GLOBE-COM Conference, 2021 [1]

have seen rapid advancements and have begun to deliver intelligent services that have the potential to revolutionize our lives. Several recent studies have looked towards the use of ML techniques for IoT-edge applications, serving as an enabler for this vision [4]. Traditional ML techniques, on the other hand, require data to be outsourced and processed in a central spot, which poses a serious privacy threat, boosts the data size transferred by edge nodes, and exacerbates communication delays caused by limited resources [5].

Federated Learning (FL) is a promising decentralized ML solution that copes with these issues while preserving the data in place. Only model parameters (i.e., weights and biases) are shared with the server while the learning process takes place on edge devices [6]–[10]. The server handles the process of developing the global model by collecting and averaging all updates performed by different edge devices. In each global round, the server regularly publishes the most recent global model for further updates. The server repeats these procedures until convergence of the global model to the optimum solution is attained. Our work focuses primarily on applying FL to edge networks, so we refer to FL as Federated Edge Learning (FEEL) [11]. The main difference is that in FL, the cloud server can engage many different clients from different locations, whereas in FEEL, the training is conducted near the clients over wireless links.

When deploying FEEL, statistical and resource heterogeneity are considered the key challenges. For the statistical heterogeneity, the data amongst edge devices is distributed in a non-independent and identically distributed (non-i.i.d.) and unbalanced fashion [5], [8], [12]. Regarding the resource heterogeneity challenge, the devices are heterogeneous, with different computation and communication capabilities. Also, bandwidth is restricted, limiting the number of devices willing to participate in a given FEEL round. This emphasizes the importance of practical and effective selection and scheduling algorithms that improve the convergence rate and produce an unbiased model that can optimally fit all incongruent data distributions, particularly in large-scale edge networks.

To tackle the statistical challenges, considerable research efforts have been devoted to studying and tackling this issue under different FEEL settings [13]–[19]. Among all these solutions, Clustered Multi-tasks Federated Learning (CFL) [16], [20] demonstrated exceptional performance by striking a balance between learning and cost. The geometry of the FEEL loss surface is utilized by CFL to cluster service users into groups using data distributions that can be trained at

the federal level. It is worth mentioning that conventional FL implies that just a single global model is always being trained for all clients regardless of the discrepancies in their data distribution. On the other hand, Sattler *et al.* [16] proved that these assumptions are commonly violated in realistic applications. In particular, it is demonstrated that at each stationary point of the FL main objective, the similarities between the local models of different participants could be measured using cosine similarity to deduce the groupings of participants with contradictory distributions (i.e., incongruent distributions).

The key advantage of CFL, in to multi-task FEEL [21] which learns several models for several related tasks and personalized FEEL [22] which provides each participant (i.e., participating client) with a customized model, is that CFL does not necessitate any change to the underlying FEEL communication protocol even though clustering occurs only after the conventional FL model reaches to a stationary point. CFL is interpreted as a post-processing scheme that can strengthen the FEEL performance by properly clustering all clients and effectively producing a customized model for each cluster. Recently, the investigations in [23]–[25] followed up on the CFL-based methodology and verified that CFL is more reliable for attaining higher accuracy than the traditional FEEL when the data is heterogeneous and non-i.i.d.

Several studies [26]–[30] have sought to solve the challenge of participants' scheduling and resource allocation (i.e., resource heterogeneity challenges) while taking non-i.i.d. data distribution (i.e., statistical challenges). For example, the authors in [26]–[30] proposed several techniques to select participants during each round of federated averaging, prioritizing those with lower training latency, better communication and computation resources, or lower energy consumption, under the assumption that each client holds the same amount of data (i.e., balanced). Yet, neither of these works [26]–[30] consider the scheduling problem for the CFL technique, which requires capturing all data distributions from all clients to perform the correct clustering based on their data distributions. This means that the scheduling approaches in the literature for traditional FL are not applicable to CFL. If devices with better channels or lower latency are regularly selected to participate in the training, the resulting models will be biased since other devices not engaged in training may have different data distributions. The challenge of scheduling in CFL is a crucial issue that remains unresolved and requires further investigation.

In response to all these remarks, this work proposes a new client scheduling framework for CFL over wireless edge networks to minimize training time and expedite the rate of convergence while equipping every group of devices with the optimal model that closely fits their data distribution. We account for the insufficient resources at the edge network (i.e., bandwidth) and the deadline constraints that the server sets to prevent a longer waiting time for updates to start a new training round. We can summarize our key contributions as follows:

- Propose a novel client scheduling algorithm for CFL to cope with the problem of limited resources while performing efficient clustering to tackle the non-i.i.d. and unbalanced data distribution problems. The proposed

algorithm is based on the fairness between the clients across the network in such a way that all clients have equal chances of being selected to participate in the training phase, despite their channel states and data sizes. This will enable the edge system to imbue the clients with more specialized models rather than biased models.
- Formulate a joint optimization problem for resource allocation and scheduling aiming to reduce the training latency and improve the convergence speed, taking into account unbalanced data distribution and non-i.i.d., device heterogeneity, and insufficient resources. Due to the NP-hardness of the problem, we propose a heuristic-based solution depending on device heterogeneity and bandwidth reuse to fairly aggregate all updates from all devices.
- Bound the impacts of the proposed approach on the convergence of the group's models concerning the number of scheduled clients, and incongruent and congruent data distribution.
- Perform experimental evaluation using two federated datasets, FEMNIST and CIFAR-10, under non-i.i.d and unbalanced data distribution. The results confirm that our proposed solutions effectively minimize the training latency and improve the convergence speed while attaining a satisfying performance.

The rest of the paper is structured as follows: we present the related work in Section II. The system, learning, computation, and communication models are introduced in Section III. Next, we formulate the minimization problem in Section IV. The proposed scheduling framework is introduced in Section V. The convergence analysis is given in Section VI, where we derive the relationships between the proposed algorithm and the convergence rate. We evaluate the proposed scheduling framework in Section VII where the experimental setup is outlined, and the numerical results are discussed. Finally, we summarize this paper and provide directions for future extensions in Section VIII.

II. RELATED WORK

The study of FL deployment via edge networks from various perspectives has gained a lot of interest in the literature. A decentralized stochastic gradient descent approach was studied in [31] considering the context of a bandwidth limit in several channel conditions, where every client is chosen opportunistically for transmission according to its channel state. Earlier works considered perfect updates-upload tasks to address the delay aspects' challenges in FEEL. Yet, the impacts of wireless channels are ignored, especially the ability to leverage the characteristics of the channels (e.g., fading, multi-access, and broadcasting) to reduce the latency. Therefore, the broadband analogue aggregation technique is fine-tuned for probabilistic channels due to restricted radio spectrum, particularly channel capacity [31]. To be more precise, prior to transmitting the models, the edge devices assess the sparsity of the gradients and then transfer them to a lower-dimensional realm constrained by the limited channel capacity. In addition, the works in [32] investigated the convergence of the FL
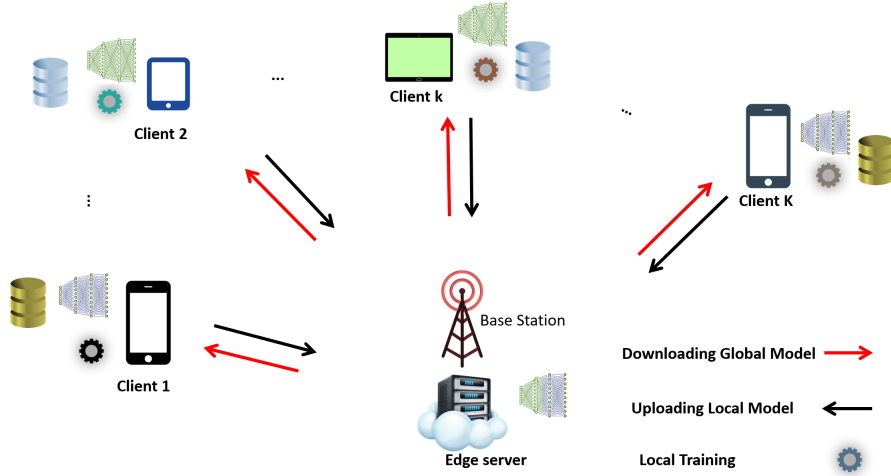
Fig. 1: System overview where the CFL is performed at edge networks.

algorithm across wireless links, taking into account the effect of unstable connections on the global model convergence.

Recently, the work in [33] introduced a solution to optimize the accuracy and cost under unreliable wireless channels. However, all analytical expressions are accounted for in the convex ML setting (not addressing the non-convexity of the deep learning algorithm).

Focusing on scheduling policies and participants' selection, to minimize the latency during the FL training process, several client scheduling techniques were proposed in [1], [27], [28], [34]–[37]. The aim is to improve the speed of the convergence rate as well as address the limitations of communication and communication resources. For example, Amiri *et al.*, [38] evaluated four alternative update quantization-based scheduling strategies. However, the authors first assumed a data symmetry among all the clients; second, they utilized stochastic gradient descent (SGD), which needs more rounds to converge on the small data. Third, data heterogeneity was not considered, so the clients with the best channels and significant L2-norm differences control the global model, resulting in a biased model. Moreover, the researchers in [39] developed a new client strategy depending on the direction of the updated gradient. This strategy, however, would almost certainly result in a skewed model, particularly for a high degree of non-i.i.d. The models with various gradient directions will not be reflected in the global model version. Lately, the work in [40], studied the client selection for hierarchical FL where the connections keep changing while the works in [30], [41] proposed approaches either to minimize the energy or to apply semi-supervised learning. Huang *et al.* [42] investigated the client selection problem, considering a volatile context scenario in such a way that some selected clients may not succeed in finishing their training and uploading tasks.

To conclude, although considerable works were devoted to optimally selecting and scheduling the participants over a wireless edge network, they only work for the traditional FL where all clients aim to train only a single model. However, CFL is entirely different from a theoretical point of view, where all incongruent data distributions must be captured

in order to produce more reliable and efficient models that optimally fit the distributions of the data amongst clients. For that, a more reliable scheduling framework should be considered to enable efficient CFL systems at the network edge.

III. SYSTEM MODEL

As Fig. 1 depicts, this work considers a set of $\mathcal{K} = \{1, \ldots, K\}$ heterogeneous edge devices, $K = |\mathcal{K}|$, associated with and coordinated by an edge server via a base station (BS). Each individual $k \in \mathcal{K}$ device owns a local dataset $\mathcal{D}_k$, with $D_k = |\mathcal{D}_k|$ number of samples, and the total number of samples amongst all devices is $D = \sum_{k=1}^{K} D_k$. Each local dataset, $\mathcal{D}_k$, comprises a number of sample data with input-output pairings as follows: $\{\mathbf{x}_{i,d}^{(k)}, y_i^{(k)}\}_{i=1}^{D_k}$, where $\mathbf{x}_{i,d}^{(k)} \in \mathbb{R}^d$ is an input with d features, and $y_i^{(k)} \in \mathbb{R}$ is the accompanying class-labeled output. Each device behaves differently based on its activities, generating different sizes of local data; thus, $\mathcal{D}_k$ is non-i.i.d. and unbalanced. As previously stated, the connections between the devices and the server are made via an unreliable wireless link. Also, the devices themselves are resource-constrained, posing the challenge of scheduling and selecting participants in order to improve the convergence speed and minimize training time, particularly for CFL, where the main objective is to train a set of models rather than a single model as in conventional training. For the reader's convenience, the main symbols utilized in this work are listed in Table I.

A. FEEL model

In FEEL, the aim is to develop a collaborative global model to be used across the network. To do that, the server initiates the global model $\boldsymbol{W}^0$. Then, at every r-*th* round, the server selects only a subset of devices Ω_r every round due to the limited resources (i.e., bandwidth sub-channels). Then, all participants receive a copy of the global model $\boldsymbol{W}_{r-1}$ in a multicast scheme. Each k-*th* selected device employs its local solver, such as SGD, to train the local model by minimizing the

TABLE I: List of important symbols.

Abbrev.	Description
K	number of clients (i.e., available devices)
k	client index k where $k \in \mathcal{K}$
$\mathcal{D}_k$	k-th client's local data
$\mathcal{D}$	Total Data samples amongst clients
$\{\mathbf{x}_{i,d}^{(k)}, y_i^{(k)}\}$	The input and the accompanying class-labeled output
$\boldsymbol{W}_r$	The parameters of the global model at r-th round
$F_r(\boldsymbol{W})$	The loss function of the global model at r-th round
f_i	The local loss function for each data sample i
$\boldsymbol{W}_k$	The model parameters of the k-th client
$\mathcal{T}$	Number of updates performed locally to update the model
η	Learning rate
E	Number of epochs
b	Batch size
T_k^{cmp}	Local computation time of k-th client
$\mathcal{T}_k$	Number of local updates
T^{tot}	The time budget for the whole training process
f_k	The used CPU speed at k-th client
Φ_k	The required CPU cycles to process one local sample
T_r	The round deadline determined by the server
T_k^{trans}	The upload time (i.e., uploading latency)
r_k^r	Transmit data rate achieved by the k-th client
B	The bandwidth size
$\lambda_r^k B$	The allocated bandwidth for client k at round r
N	Number of OFDMA sub-channels
Ω_r	Participant's selection set at r-th round
S_r^k	The binary variable to specify whether the client is selected 1 or not 0
h_r^k	The uplink channel gain between the k-th client and the BS at r-th round
P_k^r	The k-th client transmit power
ξ	Model size
$F_k(\boldsymbol{W}_r)$	Local loss function at k-th client
$\nabla F_k(\boldsymbol{W})$	Local gradient at k-th client
$\varepsilon_1, \varepsilon_2$	Hyper-parameters to control the clustering
$I(k)$	The data distribution of client k

loss function over the number of local epochs denoted by E. For example, let us assume that each device uses a mini-batch SGD; thus, at every epoch, the data is divided into batches in which the local solver performs an update on every single batch. As a result, the number of local updates performed by each participant during each global round is defined as follows:

$$\mathcal{T}_k = E\,\frac{D_k}{b}, \qquad (1)$$

where b is the batch size to determine the number of samples used for one local update. After finishing the local training, each selected device, $k \in \Omega_r$, uploads its updated model to the edge server, which in return collects and fuses all updates to create a new global model. Regularly, the server coordinates the learning process to seamlessly find the optimal model parameters $\boldsymbol{W}_r \in \mathbb{R}^d$ that are able to learn the linked output patterns y_i by repeatedly minimizing the corresponding loss function as:

$$f_i(\boldsymbol{W}_r) = \ell(\mathbf{x}_{i,d}^{(k)}, y_i^{(k)}; \boldsymbol{W}_r), \qquad (2)$$

in every r-th global round where the local loss function is defined as:

$$F_k(\boldsymbol{W}_r, \mathcal{D}_k) := \frac{1}{D_k} \sum_{i \in \mathcal{D}_k} f_i(\boldsymbol{W}_r). \qquad (3)$$

Consequently, given datasets $\mathcal{D}_1, .., \mathcal{D}_K$ amongst a set of edge devices, the global objective is to find the minimum reciprocal value of the loss function for a set of local's data $\mathcal{D} = \cup_k \mathcal{D}_k$

as follows:

$$\min_{\boldsymbol{W}} F(\boldsymbol{W}, D) = \sum_{k=1}^{K} \frac{D_k}{D} \underbrace{F_k(\boldsymbol{W}_r, \mathcal{D}_k)}_{\text{local loss}}. \qquad (4)$$

B. Local Computation and Communication Models

As previously indicated, the BS is unable to serve all existing devices due to a limited number of bandwidth sub-channels. Therefore, in every FEEL round, only a part of these devices, Ω_r, can upload their updates. In particular, the selected set is defined as:

$$\Omega_r = \{k \mid s_r^k = 1, k = 1, 2, \ldots, K\}, \qquad (5)$$

where $s_r^k = 1$ indicates that client k is on the participating set Ω_r, otherwise $s_r^k = 0$. The delay amongst the selected set includes two parts, i.e., uploading delay and computing delay. For the uploading latency, all clients are assumed to hold a similar model architecture, $\boldsymbol{W}_r$, determined by the system administrator, which has a size of ξ. We employ the orthogonal frequency division multiple access (OFDMA) technique for the communication between the BS and the devices, where each k-th participant (i.e., selected device) is given bandwidth of size $\lambda_r^k B$. To be more specific, assuming that the bandwidth is split into N sub-channels depending on the model size, we can define the number of sub-channels as follows:

$$N = \frac{B}{\xi}, \qquad (6)$$

where each 1-th sub-channel of size $\lambda_r^k B$ is allocated for each participant. Hence, each k-th device can achieve a data rate defined as:

$$r_k^r = \lambda_r^k B \ln\left(1 + \frac{P_r^k |h_r^k|^2}{N_0}\right), \qquad (7)$$

where h_r^k is the channel gain of the link between client k and the server, P_r^k is the k-th client's transmission power to upload the local model, and N_0 denotes background noise. Hence, the uploading delay could be estimated as:

$$T_{trans}^k = \frac{\xi}{r_k^r}. \qquad (8)$$

For the computing latency, each device takes a time defined as:

$$T_{cmp}^k = E\frac{\phi_k D_k}{f_k}, \qquad (9)$$

to train its local model, where f_k (cycles/second) is the central processing unit (CPU) frequency; and ϕ_k (cycles/data point) denotes CPU cycles to process one data point. Therefore, the total computing and uploading delay for each k-th participant at the r-th round is defined as:

$$T_{tot}^k = T_{trans}^k + T_{cmp}^k. \qquad (10)$$

In a real FEEL scenario, the system administrator or the coordinating server determines a time constraint (i.e., a round deadline constraint) in such a way that each participating device has to finish its tasks within this time. Specifically,
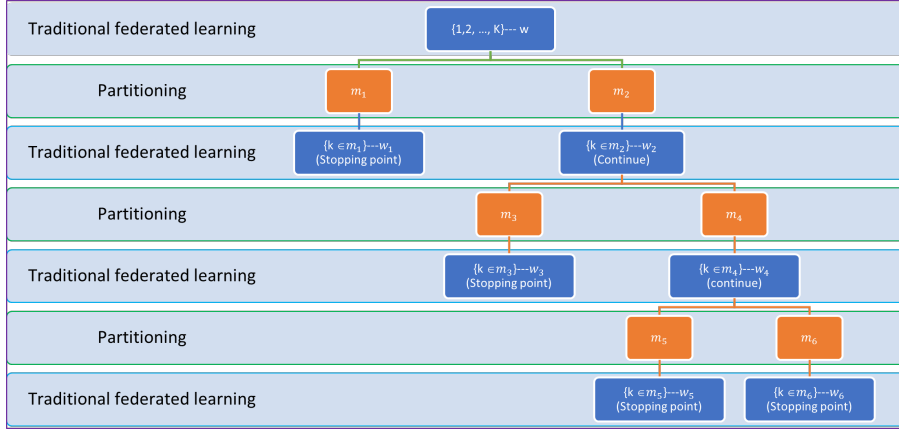
Fig. 2: CFL representation as a parameter tree where the root is the traditional FL [16].

this time could be set to the latency of the slowest participant $k \in \Omega_r$, which is defined as:

$$T_r = \max\{s_r^k(T_{trans}^k + T_{cmp}^k)\}. \tag{11}$$

C. Clustered Federated Learning (CFL)

Unlike traditional FL, whereby all clients train a single model collaboratively, the main goal of CFL is to cope with incongruent data distribution and provide a group of clients with a specialized model tailored to their local data distribution. More precisely, CFL [16] intends to generalize the traditional FEEL assumption to a set of participating devices holding a similar data distribution so as to consolidate the aforementioned challenges.

Assumption 1. *("CFL") [16]: The client population can be partitioned as $\mathcal{M} = \{c_1, \ldots, c_m\}$, $\bigcup_{i=1}^{M} c_i = \{1, \ldots, K\}$ in which every subgroup $c \in \mathcal{M}$ meets the conventional FL assumption.*

Here, $\mathcal{M}$ is the clusters' set, and $M = |\mathcal{M}|$ is the number of clusters whereby the participating devices with similar data distribution (i.e., congruent) are grouped into a single cluster. The FEEL system can be regarded as a representative parameter tree in CFL, as shown in Fig. 2 [16]. At the root node remains the conventional FEEL model, approaching the stationary point $\boldsymbol{W}^*$. In the subsequent layer, the client population is divided into two groups based on their cosine similarities, and every subgroup should reach a stationary point $\boldsymbol{W}_1^*$ and $\boldsymbol{W}_2^*$, respectively. The recursive branching continues until no further partitions are possible. It is worth mentioning that the *cosine similarity*, sim, between any two devices' updates, device k and device k', is calculated by:

$$sim_{k,k'} := sim(\nabla F_k(\boldsymbol{W}), \nabla F'_k(\boldsymbol{W})) := \frac{\langle \nabla F_k(\boldsymbol{W}), \nabla F_{k'}(\boldsymbol{W}) \rangle}{\|\nabla F_k(\boldsymbol{W})\| \|\nabla F_{k'}(\boldsymbol{W})\|}$$
$$= \begin{cases} 1 & \text{if } I(k) = I(k') \\ -1 & \text{if } I(k) \neq I(k'), \end{cases} \tag{12}$$

where both $I(k)$ and $I(k')$ denote the data distributions of k and k', respectively. Accordingly, the correct bi-partitioning is obtained by:

$$m_1 = \{k | sim_{k,0} = 1\}, \quad m_2 = \{k | sim_{k,0} = -1\}. \tag{13}$$

As in [16], the separation is only executed if the following two conditions are fulfilled:

$$0 \leq \left\| \sum_{k=1,\ldots,K} \frac{D_k}{D} \nabla_{\boldsymbol{W}} F_k(\boldsymbol{W}^*) \right\| < \varepsilon_1. \tag{14}$$

We can note that (14) retains that the achieved solution is approaching the stationary point of the conventional FL objective. In contrast, the participants are away from the stationary point of their local loss if it is aligned with the following condition:

$$\max_{k=1,\ldots,K} \|\nabla_{\boldsymbol{W}} F_k(\boldsymbol{W}^*)\| > \varepsilon_2 > 0, \tag{15}$$

where ε_1 and ε_2 control hyperparameters to manage the clustering task.

Remark 1. *The split will not be performed if all clients have the same data distribution. The reason for this is that both of the aforementioned conditions cannot be fulfilled for the same data distribution. As a result, we revert to the traditional FL with a single model.*

IV. PROBLEM FORMULATION

As in [23]–[25], all participants in CFL can be either included when the training process starts, or just a random subset is selected every round. Due to limited resources, the former assumption is impracticable for deploying CFL at the network edge (i.e., the number of sub-channels with respect to the model size). On the other hand, the latter will not catch all incongruent data distributions amongst participants, posing the challenge of effectively developing a client selection and scheduling strategy while maintaining edge wireless network restrictions and attaining required performance (i.e., efficient specialized models for all clusters). To be more specific, to properly theorize the setting of conventional FL, it is essential to partition clients of incongruent data distributions as early as

possible to expedite the convergence rate and minimize associated costs in light of bandwidth as well as data and device heterogeneity. Hence, improving the convergence speed and reducing the total training latency for all resulting specialized models (i.e., the M group models and one conventional FL model) depends on how we optimally select and schedule the client to compute and upload their updates and how to allocate the resources for the uploading task. Let R be the number of global rounds completed throughout the training time budget of T_{tot}. The goal is to find the optimal selecting and scheduling strategy that obtains the optimal set of M models, $\boldsymbol{W}^* = \{\boldsymbol{W}_m | m = 1, 2, \ldots, M\}$ that minimizes the global loss function for every group within T_{tot}. Mathematically speaking, the main objective is to find the optimal model parameters for each group as follows:

$$\boldsymbol{W}_m \triangleq \underset{\boldsymbol{W}_m \in \{\boldsymbol{W}_r^{\Omega_r} : r = 1, 2, \ldots, R\}}{\arg \min} F(\boldsymbol{W}_m). \tag{16}$$

Hence, the minimization problem for all clusters can be posed as follows:

$$\min_{\boldsymbol{W}^*, R, \Omega_r, T_{[R]}} \sum_{m=1}^{M} F(\boldsymbol{W}_m) \tag{P1}$$

$$\text{s.t.} \quad F(\boldsymbol{W}_m) - F(\boldsymbol{W}_m^*) \le \epsilon, \quad \forall m \in \mathcal{M}, \tag{P1.0}$$

$$\sum_{r=1}^{R} T_r(\Omega_r) \le T_{tot}, \quad \forall r \in [R], \tag{P1.1}$$

$$s_r^k T_{tot}^k \le T_r(\Omega_r), \quad \forall r \in [R], \forall k \in \mathcal{K}, \tag{P1.2}$$

$$T_r = \max\{s_r^k (T_{trans}^k + T_{cmp}^k)\}, \quad \forall r, \forall k, \tag{P1.3}$$

$$|\Omega_r| \le N, \quad \forall r \in [R], \tag{P1.4}$$

$$s_r^k \in \{0, 1\}, \tag{P1.5}$$

where $\Omega_r = [\Omega_1, \Omega_2, \ldots, \Omega_R]$ denotes the selected scheduling sets during the training rounds, and $T_{[R]} = [T_1, T_2, \ldots, T_R]$ is the maximum latency of every round (i.e., the deadline). Constraint (P1.0) is intended to guarantee that every subgroup model converges to the optimal model . We can see that this constraint ensures fairness across the groups where each group has optimal model parameters. The constraint (P1.1) indicates that the total training time during the FEEL process does not exceed the assigned time budget. Furthermore, constraint (P1.2) is related to the deadline constraint as well as the training and upload latency for each cooperating client. Constraint (P1.3) specifies the deadline at every round. In constraint (P1.4), the selected clients should not surpass the system bandwidth sub-channels. Last, constraint (P1.5) is a binary variable to determine whether the client is selected $s_r^k = 1$ or not $s_r^k = 0$. We can notice that P1 is a Mixed-Integer Nonlinear Programming (MINLP) and NP-hard the problem as the impacts of R and $\Omega_{[R]}$ on the weight vector of each cluster model, i.e., $F(\boldsymbol{W}_m)$, should first be found. It is worth mentioning that it is difficult to obtain an explicit expression for $F(\boldsymbol{W}_m)$ when considering R and $\Omega[R]$. Indeed, finding the optimal $\boldsymbol{W}_m$ depends on the correctness of the clustering and the corresponding congruent data distribution. Thus, $F(\boldsymbol{W}_m)$ is solved iteratively, as we see later in Section
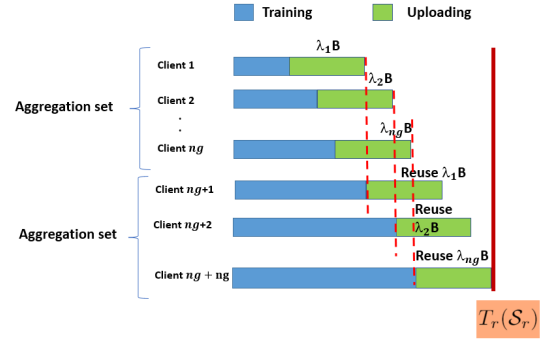

Fig. 3: Aggregation sets and bandwidth reuse every r-th round.

V. Furthermore, because of the fluctuation in the computation delay, T_{cmp}^k, and wireless channel conditions h_r^k throughout the rounds R, obtaining the optimal client scheduling technique is complicated and might even be non-stationary. In practice, the CFL mechanism is recursive, relying on the scheduled clients to assist in capturing all incongruent data distributions and provide all clients with the congruent data distribution with the optimal model parameters. Hence, the problem in P1 is solved iteratively later on as follows. First, to address the difficulties encountered by requiring to schedule all clients in each round to prevent biased models, we propose an efficient scheduling algorithm that accounts for the challenges mentioned above. Then, we extend the CFL algorithm, taking into account computing and communication resources and the deadline constraints. Last, we analytically find the relationship between the selected participants, the round deadline, and the convergence rate for the congruent and incongruent data distributions.

V. PROPOSED SOLUTION

This section presents our proposed approach including both scheduling mechanisms and the related resource allocation. We take advantage of employing bandwidth reuse and devices' heterogeneity to increase the number of participating clients to capture all data distributions. In this algorithm, the scheduling and selection procedure includes two phases. In the first phase, the server copes equally with all available clients to perform the correct clustering at the early stages of the training. Thus, all existing clients are selected to carry out the global model updates until a specific cluster reaches the stopping point (i.e., all cluster members have the same data distribution). In the second phase, the server uses the greedy approach for each cluster, reaching the stopping point where the clients with less latency are only selected to carry out further model updates. Let us assume that Ω_r includes all clients for the clusters not reaching the stopping point. The selected clients for other clusters can be added as follows:

$$\Omega_{grdy} = \{i \leftarrow \underset{i \in m}{\arg \min}\{T_i^{tot} \forall i \in m \tag{17}$$
$$|\forall m| \max_{k \in m} \|\nabla_{\boldsymbol{W}} F_k(\boldsymbol{W}^*)\| < \varepsilon_2\}\}$$

$$\Omega_r = \Omega_r \cup \Omega_{grdy}. \tag{18}$$

We can see that in (17), the greedy selection is applied only to the clusters that reach the stopping point of splitting, where the stopping point is used to decide to use greedy selection for any cluster and added it to the selection set as in (18).

Specifically, the server initially collects prior information from all clients, such as the size of data, local processing capacity, and channel condition; subsequently, the server assesses the anticipated delays of all clients to arrange model uploading depending on their completion time. The server sorts the clients in ascending order according to their estimated latencies to collect the updates effectively. At this point, the number of aggregations performed by the server to gather all updates per round could be expressed as follows:

$$ng = \frac{|\Omega_r|}{N}. \tag{19}$$

As shown in Fig. 3, the server can exploit the heterogeneity of the devices to enable a more efficient training process where the devices with a small data size can increase their local CPU speed and transmit power to finish their computing and uploading tasks. At the same time, the other clients having large data sizes can adjust their local CPU speed and the transmit power based on the completion time of the previous aggregation set. We can infer that as the first participant in the aggregation set j completes its uploading task, the first participant in the aggregation set $j + 1$ will reuse the same bandwidth frequency, i.e., sub-channel, to upload its update. This procedure is subsequently repeated for participants in all aggregation sets. Formally, we can define the aggregation set as follows:

$$\mathcal{G} = \{\{1 + (N(j-1)), \ldots, N + (N(j-1))\}, j = 1, 2, \ldots, ng\}. \tag{20}$$

To find the resources for both the computation and communication tasks, each client needs to adjust its local CPU speed and the transmit power, relying on the pre-determined deadline. It is worth highlighting that the server is anticipated to have considerably more computing capabilities than the clients; therefore, the computation cost on the server side is negligible. We can see that there is a trade-off in the scheduling problem between the number of participants chosen in a given round and the entire number of rounds, which could be handled by T_r. Assuming we started with a short T_r, the number of engaging participants in Ω_r can decrease, which is unsuitable for CFL leading to slowly partitioning the clients into congruent groups. This brings out unwanted communication costs due to increasing the number of training rounds needed to converge. Hence, we propose to initiate a long T_r at the beginning of the training; then, this time is reduced as the correct clustering is performed where only the best clients that require less latency will be selected from the cluster reaching the stopping point as follows:

$$\max_{k \in m} \|\nabla_{\boldsymbol{W}} F_k(\boldsymbol{W}^*)\| < \varepsilon_2. \tag{21}$$

The server uses (21) to check if all clients within cluster m have congruent data distribution. If so, the optimal solution $\boldsymbol{W}^*$ is returned, and the CFL is stopped for that group. Otherwise, the server partitions and clusters the clients using

---

Algorithm 1 : The detailed steps of our proposed approach for efficient CFL.

In: K, w_0, $\varepsilon_1, \varepsilon_2 > 0$, E, b
Out: M specialized models, one conventional FL model
Init: Start with $\mathcal{M} = \{\{1, \ldots, K\}\}$ as initial clustering, set $w_k \leftarrow w_0$ $\forall k$, and $r = 1$
1: **while** all M does not reach the stopping point **do**
          // Server Side - Before local updates start
2:    **if** $M > 1$ **then**
3:       Server **chooses**
4:       $\{m \in \mathcal{M} | \max_{k \in m} \|\nabla_{\boldsymbol{W}} F_k(\boldsymbol{W}^*)\| < \varepsilon_2 \geq 0\}$
5:    **else**
6:       **schedule** all participants willing to take part in the training process
7:    **end if**
8:    Server **measures** the approximate delay of $\mathcal{K}$ and sort them in ascending
9:    Server **organizes** the aggregation set using (19) and (20)
10:   Server **sends** w_{r-1} in multi-cast manner
          // Participants Simultaneously perform
11:   **for** $k = 1$ to $|\Omega_r|$ **do**
12:      get w_{r-1}
13:      **Carry out** local training $w_k = \boldsymbol{w}_{r-1} - \eta_r \sum_{t=1}^{\mathcal{T}} \nabla F_k(\boldsymbol{w}_k(t))$
14:   **end for**
15:   Server **collects** models as in (20)
          // Server Performs the Post-processing Steps
16:   $\mathcal{M}_{tmp} \leftarrow \mathcal{M}$
17:   Server **finds** $F(w^r) = \sum_{k=1}^{K} \frac{D_k}{D} F_k(w^r)$ and $w^r = \sum_{k=1}^{K} \frac{D_k}{D} w^r$
18:   **for** $c \in \mathcal{M}$ **do**
19:     • Server **obtains** $\Delta W_m \leftarrow \frac{1}{|c|} \sum_{k \in c} \Delta w_k$
20:     **if** $\|\Delta w_c\| < \varepsilon_1$ and $\max_{k \in c} \|\Delta w_k\| > \varepsilon_2$ **then**
21:       • $sim_{k,k'} \leftarrow \frac{\langle \Delta w_k, \Delta w_{k'} \rangle}{\|\Delta w_k\| \|\Delta w_{k'}\|}$
22:       • $c_1, c_2 \leftarrow \arg\min_{c_1 \cup c_2 = c}(\max_{k \in c_1, k' \in c_2} sim_{k,k'})$
23:       • $sim_{cross}^{max} \leftarrow \max_{k \in c_1, k' \in c_2} sim_{k,k'}$
24:       • $\gamma_k := \frac{\|\nabla F_{I(k)}(\boldsymbol{W}^*) - \nabla F_k(\boldsymbol{W}^*)\|}{\|\nabla F_{I(k)}(\boldsymbol{W}^*)\|}$
25:      **if** $max(\gamma_k) < \sqrt{\frac{1 - sim_{cross}^{max}}{2}}$ **then**
26:        • $\mathcal{M}_{tmp} \leftarrow (\mathcal{M}_{tmp} \setminus c) \cup c_1 \cup c_2$
27:      **end if**
28:     **end if**
29:   **end for**
30:   • $\mathcal{M} \leftarrow \mathcal{M}_{tmp}$
31:   $r = r + 1$
32: **end while**
33: **Server** ***Return*** M specialized models, and one conventional FL model.

---

(12), (14), and (15) at every round. In this context, the similarities between cooperating participants k and k' within a group could be bounded as follows:

$$sim_{within}^{min} = \min_{\substack{k,k' \\ I(k) = I(k')}} sim(\nabla_w r_k(\boldsymbol{W}^*), \nabla_w r_{k'}(\boldsymbol{W}^*)), \tag{22}$$

At the same time, we can bound the similarities between participants k and k' belonging to two different clusters as follows:

$$sim_{cross}^{max} = \max_{k \in c_1^*, k' \in c_2^*} sim(\nabla_{\boldsymbol{W}} r_k(\boldsymbol{W}^*), \nabla_{\boldsymbol{W}} r_{k'}(\boldsymbol{W}^*)). \tag{23}$$

It is worth highlighting that (22) and (23) are employed to measure the separation gap which can be expressed as follow:

$$g(sim) := sim_{intra}^{min} - sim_{cross}^{max}. \tag{24}$$

All steps of our proposed framework, including the client scheduling algorithm and the CFL training algorithm, are listed in Algorithm 1 where the algorithm has input parameters K, $\varepsilon_1, \varepsilon_2 > 0$, E, and b. Algorithm 1 starts with an initialization step by clustering all clients in one group. Then,

the server aggregates prior information such as the data size, channel state, and computational capabilities from all existing clients (line 2). In lines 3-4, the algorithm checks whether the clustering was performed before or not. If achieved, the server checks the stopping condition (21), and selects the best clients having less latency to take part in the training from that group, while all clients are selected for others. Otherwise, the server determines all clients to participate in the FEEL round. In Algorithm. 1, we observe that whenever a given cluster approaches the stationary point of its model (lines 2–7), the server shifts to employing a greedy selection algorithm in which the clients with less training time and better resources are selected to conduct subsequent updates. This is due to the fact that clients with similar data distribution (i.e., congruent data distribution) behave likewise. For each cluster not reaching a stationary point, the server estimates all clients' latency (line 8) to schedule uploading the local models' updates and find the aggregation set (line 9) due to the limited bandwidth. In line 10, the server broadcasts the latest updated models to the selected clients. In lines (11–14), all chosen clients perform conventional FEEL updates and return them to the coordinating, which collects (line 15) all updates based on the aggregation sets. In lines 16-29, the server runs the clustering algorithm to cluster the participants according to their local data distribution if the model of the last cluster reaches the stationary points. Otherwise, the server will keep using the traditional FEEL algorithm. All these steps are repeated until all models reach the stationary points and all incongruent data distributions are captured as in Algorithm. 1. It is worth mentioning that since the server depends on the uploaded models, clustering the participants isn't affected by the mobility of the devices as long as they are within the server's coverage area.

VI. ANALYSIS OF THE CONVERGENCE UNDER THE PROPOSED ALGORITHM

This section analyzes the relationship between our proposed approach and the convergence rate of the specialized models that optimally fit M incongruent data distributions. To begin with, let us define $\boldsymbol{W}_m^* = \arg\min_{\boldsymbol{W}} F(\boldsymbol{W}_m^*)$ as an optimal parameters of the specialized model for cluster m that is related to the minimum loss $F(\boldsymbol{W}_m^*)$ across all group's devices. We also define $\boldsymbol{W}_k^* = \arg\min_{\boldsymbol{W}_k} F(\boldsymbol{W}_k^*)$ as an optimal local model parameters at client k. Accordingly, the optimal gap between the global of a cluster m and local loss can be expressed as:

$$g(F_m) \triangleq F(\boldsymbol{W}_m^*) - \frac{1}{|\Omega_r^m|} \sum_{k \in [m]}^{|\Omega_r^m|} F_k^*, \quad (25)$$

where $g(F_m) \geq 0$ indicates that the cluster's model $\boldsymbol{W_m}$ does not reach the optimal solution yet. For a given cluster when reaching the optimal solution, $g(F_m)$ approaches zero.

To begin with, we apply the commonly used assumptions in FEEL [38], [43], [44] listed as follow:

Assumption 2. *Local loss functions $F_1, \dots, F_K$ amongst clients are all β-smooth; that is, $\forall \boldsymbol{W}', \boldsymbol{W} \in \mathbb{R}^d$,*

$$F_k(\boldsymbol{W}') - F_k(\boldsymbol{W}) \leq \langle \boldsymbol{W}' - \boldsymbol{W}, \nabla F_k(\boldsymbol{W}) \rangle + \frac{\beta}{2} \|\boldsymbol{W}' - \boldsymbol{W}\|_2^2,$$
$$\forall k \in \mathcal{K}. \quad (26)$$

Assumption 3. *Local loss functions $F_1, \dots, F_K$ amongst clients are all α-strongly convex; that is, $\forall \boldsymbol{W}', \boldsymbol{W} \in \mathbb{R}^d$,*

$$F_k(\boldsymbol{W}') - F_k(\boldsymbol{W}) \geq \langle \boldsymbol{W}' - \boldsymbol{W}, \nabla F_k(\boldsymbol{W}) \rangle + \frac{\alpha}{2} \|\boldsymbol{W}' - \boldsymbol{W}\|_2^2,$$
$$\forall k \in \mathcal{K}. \quad (27)$$

Assumption 4. *The expected squared l_2-norm of the local gradients is bounded as [38]:*

$$\mathbb{E}_{\mathfrak{D}} \left[\|\nabla F_k(\boldsymbol{W}_k(r), \mathfrak{D}_k(r))\|_2^2 \right] \leq \varrho^2, \quad \forall k \in \mathcal{K}, \forall r. \quad (28)$$

where $\mathfrak{D}_k(r)$ is a mini-batch sample at each local iteration. It is worth mentioning that all those assumptions, when combined, allow for the analysis of the learning algorithm's convergence rate, which is an important measure of its efficiency as seen next in Section VI-A. They also aid in determining the conditions under which the proposed algorithm is guaranteed to reach a solution.

A. Convergence Details

As seen in Section V, our proposed approach has two scheduling phases. First, all active and available clients have equal chances to take part in the global model training task. Second, for a particular cluster that reaches the stopping point, only one unique participant providing the lowest latency is selected to perform further updates for the FEEL model. Hence, for the first phase, the probability of selecting any client in every r-*th* round is $\frac{|\Omega_r^m|}{K} = 1$. Now, we find the relationship between the participating clients, the number of epochs and data samples, and the convergence rate. We follow the steps similar to [38]. However, in our work, we consider the CFL as model training and the mini-batch SGD as a local solver where the number of updates performed locally is proportional to the number of data samples. It is worth noting that we consider unbalanced data distribution where the number of examples (i.e., data points) is randomly distributed following the power law. Consequently, the number of local iterations can be calculated as in (1).

Theorem 1. *Given a learning rate $\eta_r = \frac{1}{\alpha \mathcal{T}}$, $\forall r$, we have:*

$$\mathbb{E}\left[\|\boldsymbol{W}(r) - \boldsymbol{W}^*\|_2^2\right] \leq \left(\prod_{t=0}^{r-1} \zeta_1(t)\right) \|\boldsymbol{W}_0 - \boldsymbol{W}^*\|_2^2$$
$$+ \sum_{t'=0}^{r-1} \zeta_2(t') \prod_{t=t'+1}^{r-1} \zeta_1(t), \quad (29a)$$

where

$$\zeta_1(t) \triangleq 1 - \alpha \eta_t (\mathcal{T} - \eta_t(\mathcal{T} - 1)), \quad (29b)$$
$$\zeta_2(t) \triangleq (1 + \alpha(1 - \eta\zeta_1(t))) \eta^2(t)$$

$$\varrho^2 \frac{\mathcal{T}(\mathcal{T}-1)(2\mathcal{T}-1)}{6} + \eta^2(t)(\mathcal{T}^2 + \mathcal{T} - 1)\varrho^2$$
$$+ 2\eta\zeta_1(t)(\mathcal{T}-1)\mathfrak{F}. \tag{29c}$$

Proof. See Appendix A. □

Corollary 1. *From the β-smoothness of the loss function $F(\cdot)$, after R global rounds, for each cluster's model, we have:*

$$\mathbb{E}\left[F(\boldsymbol{W}(R))\right] - F^* \leq \frac{\beta}{2}\mathbb{E}\left[\|\boldsymbol{W}(R) - \boldsymbol{W}^*\|_2^2\right]$$
$$\leq \frac{\beta}{2}\prod_{t=0}^{R-1}\zeta_1(t)\|\boldsymbol{W}_0 - \boldsymbol{W}^*\|_2^2$$
$$+ \frac{\beta}{2}\sum_{t'=0}^{R-1}\zeta_2(t')\prod_{t=t'+1}^{R-1}\zeta_1(t). \tag{30}$$

We can notice that the last inequality results from Theorem 1 [38]. If the learning rate is continuously decreasing, $\lim_{r\to\infty}\eta_r = 0$, and we can simply infer that $\lim_{R\to\infty}\mathbb{E}\left[F(\boldsymbol{W}(R))\right] - F^* = 0$.

Remark 2. *For the client scheduling in our proposed approach, the client scheduling is not random, which brings out the advantage of capturing the data distribution amongst clients and avoiding unbiased models.*

Remark 3. *For the dominant cluster that has more similarities, the average loss within its group depends on many other parameters such as β, α, K, $|\Omega_r^m|$, ε_1 and ε_2.*

VII. NUMERICAL EXPERIMENTS

In this section, we perform extensive simulations to evaluate the proposed algorithms. The system models described in Section III are properly considered with the following experiment details.

A. Experimental Setup

In all experiments, we use a bandwidth of $B = 10$ MHz, with each sub-channel having a bandwidth of 1 MHz. The channel gain of each device, h_r^k, is randomly modeled with a path loss ($\alpha = g_0(\frac{d_0}{d})^4$) , where $g_0 = -35$ dB and the baseline distance $d_0 = 2$ m. We assume that the distances between the devices and the coordinating server are uniformly distributed between 20 and 100 m. In addition, the power of AWGN is defined as $N_0 = 10^{-6}$ watts. The transmission power P_r^k is distributed at random between $p_{min} = -10$ dBm and $p_{max} = 20$ dBm. The device's CPU frequency f_k is generated randomly between 1 GHz and 9 GHz, and the required CPU cycles per data point, ϕ, is set to 20 and it is assumed to be homogeneous for all devices. We use two federated datasets, FEMNIST and CIFAR-10 [45], for handwriting classification and object recognition, respectively. Specifically, FEMNIST is utilized for handwriting classifications of both letters and digits (A-Z, a-z, and 0-9), and it has 305,654, 28 x 28, images while CIFAR-10 consists of 60,000 32x32 colored images. We split both datasets for each device into 80% for training and 2% for testing. Further, we utilize both datasets in a non-i.i.d. manner, whereas we split each dataset into $\mathcal{I}$ fragments, and then we assign each device only two random classes. For the model architecture, the convolutional neural network (CNN) classifier is adopted for FEMNIST, and Alexnet—a deep neural network (DNN) is adopted for CIFAR-10. In our models, we employed 2 hidden layers in the FEMNIST model and 13 hidden layers in the CIFAR-10 model. For both learning tasks, we utilized the ReLU activation function for the hidden layers and the Softmax activation function for the output layer. To simplify the presentation, we list all simulation parameters in Table II.

TABLE II: Simulation parameters

Sym.	Parameter	Value(s)
K	# of participating devices	(20, 50, 100, 200)
R	# of training rounds	200 for FEMNIST and 500 for CIFAR-10
E	# of local training epochs	10
b	Batch size	32
η	Learning rate	0.01, 0.009
B	Bandwidth	10 MHz
P_{min}	Minimum transmission power	-10 dBm
P_{max}	Maximum transmission power	20 dBm
N_0	Background noise	-10 dBm
P_k^{min}	Minimum CPU frequency	1 GHz
P_k^{max}	Maximum CPU frequency	9 GHz

B. Benchmarks

In this paper, we evaluate our proposed algorithms against the following state-of-the-art algorithms:

- **Random approach** [16], [46]: In this approach, the clients are scheduled randomly in each round regardless of the number of local samples, the computation and communication latencies, or the quality of the update.
- **Best channel scheduling** [38]: This approach aims to select the clients with the best channel states without taking into account the impacts of the local updates on the global model.
- **Best local updates** [38]: This approach seeks to determine the participating devices with the maximum L2-norm where the participants first calculate their gradients, then find the Euclidean distance between the local parameters and the parameters of the last received model from the server. After that, they notify the server with their status, which in turn keeps the connection with the participants' having maximum L2-norm.
- **Maximum number of data points**: In this algorithm, the coordinating server chooses the participants having the largest data sizes and keeps connecting with them until the end of the training. We also extend this approach to select the maximum number of data examples amongst clients every round, assuming that the channel of each participant is not stable and diverse participants may be involved in different rounds.

C. Numerical Results

To evaluate the feasibility and performance of our proposed approach, we present and compare the findings against the benchmark algorithms. To ensure a fair comparison, we utilize

similar experimental setups and the hyper-parameters for all scheduling approaches regarding the number of local epochs, model structure, learning rate, and data distribution (i.e., we employed non-i.i.d. and unbalanced data distribution for both). For all experiments, we adjust R, the global training rounds, to 200 and 500 for FEMNIST and CIFAR-10, respectively. We also set $E = 10$, and the batch-size $b = 32$. First, we carried out the experiments and reported the results during each global round. Then, we test theresulting models after finishing the training process to showcase the effectiveness of the proposed approach to equipping each device with the best-fitting model. It is important to emphasize that all findings are averaged over five trials.

1) Performance Gain on Clustering Speed and Convergence Rate: To evaluate the proposed algorithms on the splitting and correct clustering, we start conducting the experiments using FEMNIST. We use the random selection as a baseline in this stage as it shows the best performance among benchmarks. Fig. 4 exhibits the performance of the proposed algorithm and the random selection algorithm in terms of convergence rate and clustering speed. We use the bordered accuracy with the confidence threshold at each round (Figs. 4a and 4b, left) and the gradients' norm among all participants to showcase the acceleration of the convergence rate (Figs. 4a and 4b, right). From both figures, it is clear to note that our proposed approach (fairness followed by greedy approach) gains a much faster convergence rate regarding the speed of clustering since the partitioning starts at the training round of index 37, whereas the partitioning using the random scheduling algorithm starts at the training round of index 83. This confirms that the acceleration rate of the proposed algorithm is 2x faster than the benchmarks. It is noted that in our approach, the partitioning actually takes place again for all participants with incongruent distributions at global training rounds of index 45 and 63, respectively until no further clustering is possible. All trained models achieve the stopping criteria as in (21) at round 190, proving that clustering is correctly performed, according to Fig. 4a right. On the other hand, the baseline algorithm, Fig. 4b right, shows that the CFL still requires many more rounds to reach the stopping point at round 200. This is owing to the random client scheduling behavior, which may choose clients who have previously been grouped, bringing out the need for many more rounds to capture the data amongst clients to perform the correct clustering. For example, the clustering becomes slow if clients with different data distributions are randomly selected at the latter rounds.

2) Performance Gain in Terms of Testing Accuracy and Performance Gap: Now, we present the results of the testing accuracy, focusing on the fairness between clients. We note that the training procedure for both all the algorithms, the proposed and benchmarks, has produced multiple models based on clustering, including a conventional FEEL model and a more tailored model for every group, as depicted in Table III. We test all models amongst 15 clients after finishing all the training rounds to show the effectiveness of each model depending on the resulting accuracy. One can see that our proposed approach yields three better-tailored models. As shown in Table IIIa, it is clear that all participating

devices attain well-fitting accuracy (outlined in green color), and inconsistency in performance (maximum accuracy and minimum accuracy in the last row of Table IIIa) across all is approximately 10% only when our approach is used. In comparison, as depicted in Table IIIb, almost more than $\frac{1}{3}$ of the tested participants(i.e., P 1, P 2, P 3, P 4, and P 5) achieve undesired accuracy (outlined in red color). The inconsistency in performance can reach up to 31.4%, demonstrating that some resulted models are still biased to the repeatedly selected devices which dominate the clustering.

Furthermore, to validate the proposed algorithms, we perform other simulation experiments using the CIFAR-10 dataset as a complex learning task under non-i.i.d. federated settings. This assists in generalizing the performance of the proposed algorithms for different application domains. Figs. 5a–5d show the minimum, maximum, and average testing accuracy amongst clients where the minimum accuracy represents the lowest accuracy can be achieved by the client and vise versa. It is worth mentioning that the proposed approach balances the accuracy between all clients despite the number of participating clients where the gap between the maximum and minimum accuracy amongst clients reaches up to 10%, meaning that the resulting models optimally fit all local data distribution. At the same time, we notice that the best channel, the best l2-norm, and the max-samples approaches increase the accuracy gap between the clients, reaching up to 70%, especially if the number of clients increases. This stems from the fact that the server stays connected with the same clients during all global rounds in those approaches, particularly when the channel is not changing rapidly, leading to biased models that can only fit the participants' clients. In addition, random scheduling provides less accuracy gaps due to the randomness of participant selection. Nevertheless, the proposed scheduling approach substantially outperforms all benchmarks and provides much more stable accuracy regardless of the distribution of the data and the number of clients in the network.

D. Lessons Learned

The key takeaways from the experiments outlined in this paper can be listed as follows.

- The proposed scheduling algorithms can accelerate the convergence rate compared to the baseline scheduling algorithms due to the fairness between clients at the beginning of the training.
- The proposed approach is effectively suited to deal with CFL when compared with the scheduling algorithms of traditional FL, seeking to avoid biased models to frequently selected clients.
- Selecting all clients at the beginning of the training leads to performing the correct clustering and capturing all data distributions amongst clients while increasing the clustering speed and accelerating the convergence rate.
- The proposed algorithms also reduce the resource consumption as only a single client can take part in the model training once a specific cluster converges to the stopping point.
- Overall, the proposed approach surpasses the benchmarks in terms of agglomeration quality with more tailored
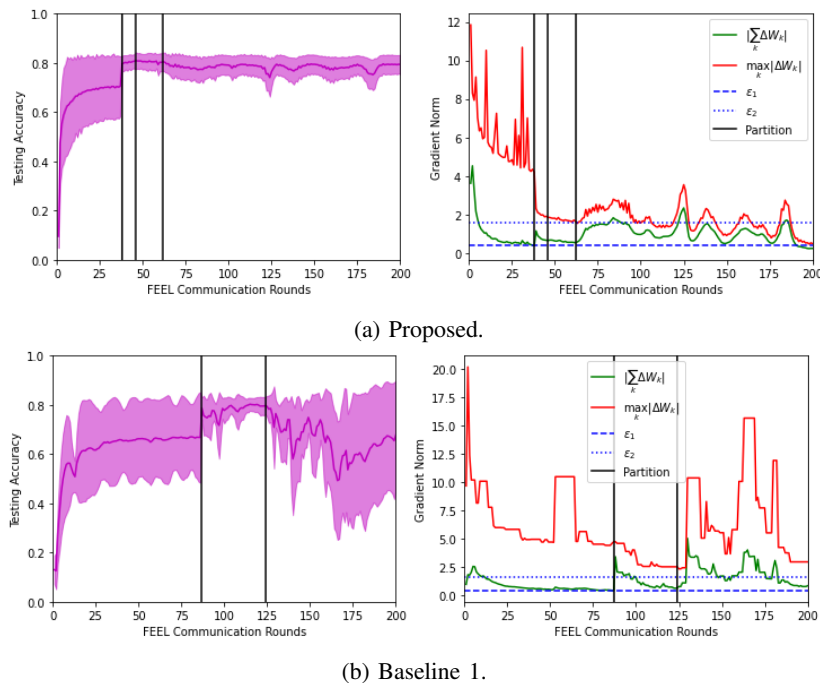
(a) Proposed.

(b) Baseline 1.

Fig. 4: Average testing accuracy of the clusters' models during the global training rounds for both our proposed approach, (a: left), and the benchmarks (b: left). In (a: right and b: right) the gradient norm of global and local loss functions during the global training rounds for both the proposed algorithms and benchmarks, respectively.

TABLE III: Models' testing accuracy after finishing all global training rounds: a conventional FL model and the tailored models of all groups (the proposed and benchmark algorithms), P*=Participant, M*=Model. (FEMNIST)

(a) Proposed Approach.

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11	P 12	P 13	P 14	P 15
Conventional FL M	45.8	38	46	45	36.9	76	81	82.9	78.7	78.9	76.95	77	78	77	76
M 1	0	0	0	0	0	77.1	83.3	86	81.6	82	81.8	81.5	82.5	79.6	80.9
M 2	0	0	81.6	76	77.6	0	83.8	85.5	0	82.7	80.1	0	84.5	82	81.8
M 3	83.3	0	77	67	77.9	76.8	0	0	0	82.7	0	0	0	0	0
M 4	0	74.5	0	0	0	0	82.5	83	75.2	0	0	0	0	0	0
M 5	83.3	78.6	0	0	0	76.8	0	0	75.2	0	0	76.8	0	0	0
M 6	83.3	74.5	76.9	67.2	77.9	0	0	0	0	0	0	0	0	0	0
Max Acc	83.3	78.6	81.6	76	77.9	77.1	83.8	86	81.6	82.7	81.8	81.5	84.5	82	81.8

(b) Benchmark 1 (Random Scheduling Approach).

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11	P 12	P 13	P 14	P 15
Conventional FL M	50.7	41.6	49.2	46.4	41.5	73.1	80.1	82.1	76.5	77.8	77	77.3	76.7	76.1	77.7
M 1	0	0	0	0	0	78.5	84.5	86.5	82.1	81.4	81.5	79.9	81	79.9	81.1
M 2	0	0	0	52.4	56.5	0	78.3	75.8	77.9	72.7	0	0	78.7	78.2	67.4
M 3	59.7	61.1	62.5	0	0	82	0	0	0	0	82.8	67.9	0	0	0
M 4	59.7	61.1	62.5	52.4	56.5	0	0	0	0	0	0	0	0	0	0
Max Acc	59.7	61.1	62.5	52.4	56.5	82	84.5	86.5	82.1	81.4	82.8	79.9	81	79.9	81.1

models, expediting convergence and minimizing training costs.

VIII. CONCLUSION

In this paper, we have proposed novel client scheduling and selection algorithms for clustered federated multitask learning to reduce the training latency and speed up the convergence rate, which improves the resulting model for each cluster. The proposed algorithms are based on the fairness between the clients across the network, where all available clients have equal chances of being selected to take part in the training process regardless of the channel state or the size of their local data. This enables the edge system to imbue the clients with more specialized models rather than having biased models. Given non-i.i.d. and unbalanced data distribution, clients' heterogeneity, and restricted bandwidth, we have first formulated an optimization problem to obtain the best client scheduling that minimizes the training cost and improves the convergence speed. We have analyzed the relationship between the proposed scheduling approach and the convergence rate of the specialized models. We have conducted extensive simulation experiments using realistic federated datasets, FEMNIST and CIFAR-10. The findings demonstrate that the proposed approach efficiently reduces the training latency and acceler-

(a) $K = 20$

(b) $K = 50$
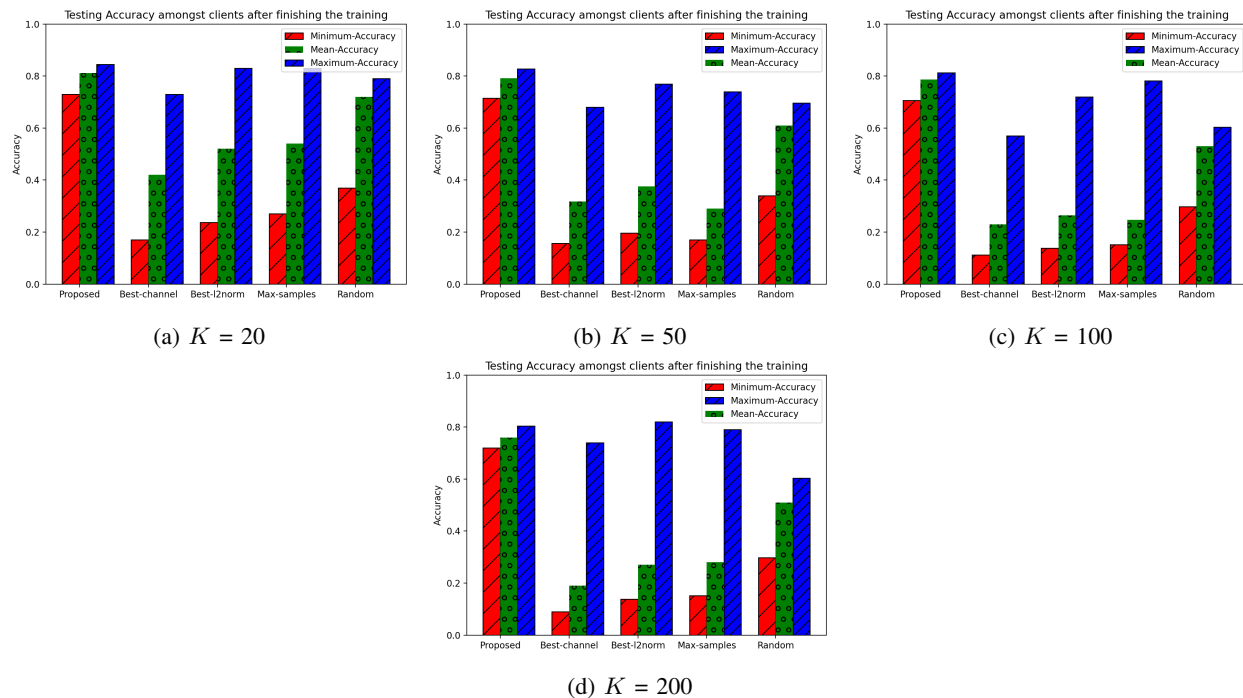

(c) $K = 100$

(d) $K = 200$

Fig. 5: Minimum, average, and maximum accuracy overall testing clients (CIFAR-10) when the data is distributed over 20, 50, 100, and 200 clients.

ates convergence while attaining a satisfying performance. For future research, it will be interesting to adapt the age of the updates for the missing clients' updates and find the optimal thresholds for the splitting conditions.

REFERENCES

[1] A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Erbad, "Client selection approach in support of clustered federated learning over wireless edge networks," in *2021 IEEE Global Communications Conference (GLOBE-COM)*, 2021, pp. 1–6.

[2] Y. Chen, Y. Sun, B. Yang, and T. Taleb, "Joint caching and computing service placement for edge-enabled iot based on deep reinforcement learning," *IEEE Internet of Things Journal*, 2022.

[3] Y. Chen, Y. Sun, C. Wang, and T. Taleb, "Dynamic task allocation and service migration in edge-cloud iot system based on deep reinforcement learning," *IEEE Internet of Things Journal*, 2022.

[4] Y. Luo, J. Yang, W. Xu, K. Wang, and M. Di Renzo, "Power consumption optimization using gradient boosting aided deep q-network in c-rans," *IEEE Access*, vol. 8, pp. 46 811–46 823, March 2020.

[5] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1342–1397, 2021.

[6] B. S. Ciftler, A. Albaseer, N. Lasla, and M. Abdallah, "Federated learning for rss fingerprint-based localization: A privacy-preserving crowdsourcing method," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 2112–2117.

[7] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476–5497, 2021.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[9] A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha, "Exploiting unlabeled data in smart cities using federated edge learning," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 1666–1671.

[10] A. Albaseer and M. Abdallah, "Privacy-preserving honeypot-based detector in smart grid networks: A new design for quality-assurance and fair incentives federated learning framework," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, 2023, pp. 722–727.

[11] X. Wang, C. Wang, X. Li, V. C. Leung, and T. Taleb, "Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9441–9455, 2020.

[12] K. D. Stergiou and K. E. Psannis, "Federated learning approach decouples clients from training a local model and with the communication with the server," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022.

[13] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," *arXiv preprint arXiv:2105.10056*, 2021.

[14] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the heterogeneity: A self-organized federated learning framework for iot," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3088–3098, July 2020.

[15] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," *arXiv preprint arXiv:2002.07948*, 2020.

[16] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy con-

straints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710–3722, August 2020.

[17] A. Albaseer, M. Abdallah, A. Al-Fuqaha *et al.*, "Data-driven participant selection and bandwidth allocation for heterogeneous federated edge learning," 2022.

[18] T. Subramanya and R. Riggio, "Centralized and federated learning for predictive vnf autoscaling in multi-domain 5g networks and beyond," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 63–78, 2021.

[19] M. H. Mahmoud, A. Albaseer, M. Abdallah, and N. Al-Dhahir, "Federated learning resource optimization and client selection for total energy minimization under outage, latency, and bandwidth constraints with partial or no csi," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 936–953, 2023.

[20] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, "On the byzantine robustness of clustered federated learning," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8861–8865.

[21] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems*, December 2017, pp. 4424–4434.

[22] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," *arXiv preprint arXiv:2012.08565*, 2020.

[23] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning," *arXiv preprint arXiv:2005.01026*, 2020.

[24] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[25] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *Proc. of International Joint Conference on Neural Networks (IJCNN)*, July 2020, pp. 1–9.

[26] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *arXiv preprint arXiv:2004.04314*, 2020.

[27] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 453–467, September 2020.

[28] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, September 2019.

[29] X. Chen, G. Zhu, Y. Deng, and Y. Fang, "Federated learning over multi-hop wireless networks with in-network aggregation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4622–4634, 2022.

[30] A. Albaseer, M. Abdallah, A. Al-Fuqaha, A. Erbad, and O. A. Dobre, "Semi-supervised federated learning over heterogeneous wireless iot edge networks: Framework and algorithms," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 626–25 642, 2022.

[31] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3546–3557, 2020.

[32] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time minimization of federated learning over wireless networks," in *Proc. of IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[33] C. Feng, Z. Zhao, Y. Wang, T. Q. Quek, and M. Peng, "On the design of federated learning in the mobile edge computing systems," *IEEE Transactions on Communications*, 2021.

[34] A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Erbad, "Threshold-based data exclusion approach for energy-efficient federated edge learning," in *Proc. of IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2021.

[35] A. M. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Erbad, "Fine-grained data selection for improved energy efficiency of federated edge learning," *IEEE Transactions on Network Science and Engineering*, July 2021.

[36] M. H. u. Rehman, A. M. Dirir, K. Salah, E. Damiani, and D. Svetinovic, "Trustfed: A framework for fair and trustworthy cross-device federated learning in iiot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8485–8494, 2021.

[37] A. Hammoud, H. Otrok, A. Mourad, and Z. Dziong, "On demand fog federations for horizontal federated learning in iov," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022.

[38] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3643–3658, Jun. 2021.

[39] H. Wu and P. Wang, "Node selection toward faster convergence for federated learning on non-iid data," *arXiv preprint arXiv:2105.07066*, 2021.

[40] Z. Qu, R. Duan, L. Chen, J. Xu, Z. Lu, and Y. Liu, "Context-aware online client selection for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1–15, 2022.

[41] A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Erbad, "Balanced energy consumption based on historical participation of resource-constrained devices in federated edge learning," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, 2022, pp. 300–305.

[42] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya, "Stochastic client selection for federated learning with volatile clients," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

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

[44] N. H. Tran, W. Bao, A. Zomaya, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. of IEEE INFOCOM*, 2019, pp. 1387–1395.

[45] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[46] M. Duan, D. Liu, X. Ji, R. Liu, L. Liang, X. Chen, and Y. Tan, "Fedgroup: Ternary cosine similarity-based clustered federated learning framework toward high accuracy in heterogeneous data," *CoRR*, vol. abs/2010.06870, 2020. [Online]. Available: https://arxiv.org/abs/2010.06870

Abdullatif Albaseer (Member, IEEE) received an M.Sc. degree in computer networks from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2017 and a Ph.D. degree in computer science and engineering from Hamad Bin Khalifa University, Doha, Qatar, in 2022. He is a Postdoctoral Research Fellow with the Smart Cities and IoT Lab at Hamad Bin Khalifa University. He has authored and co-authored more than twenty conference and journal papers in IEEE ICC, IEEE Globecom, IEEE CCNC, and IEEE Transactions. He also has six US patents in the area of the wireless network edge. His current research interests include semantic communication, federated learning over network edge, industrial IoT, and cybersecurity.

Mohamed Abdallah (Senior Member, IEEE) received the B.Sc. degree from Cairo University, in 1996, and the M.Sc. and Ph.D. degrees from the University of Maryland at College Park, in 2001 and 2006, respectively. From 2006 to 2016, he held academic and research positions at Cairo University and Texas A&M University at Qatar. He is currently a Founding Faculty Member with the rank of Associate Professor with the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). His current research interests include wireless networks, wireless security, smart grids, optical wireless communication, and blockchain applications for emerging networks. He has published more than 150 journals and conferences and four book chapters, and co-invented four patents. He was a recipient of the Research Fellow Excellence Award at Texas A&M University at Qatar, in 2016, the Best Paper Award in multiple IEEE conferences including the IEEE BlackSeaCom 2019, the IEEE First Workshop on Smart Grid and Renewable Energym in 2015, and the Nortel Networks Industrial Fellowship for five consecutive years, from 1999 to 2003. His professional activities include an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE OPEN ACCESS JOURNAL OF COMMUNICATIONS, a Track Co-Chair of the IEEE VTC Fall 2019 conference, a Technical Program Chair of the 10th International Conference on Cognitive Radio Oriented Wireless Networks, and a Technical Program Committee Member of several major IEEE conferences.

Ala Al-Fuqaha (Senior Member, IEEE) received Ph.D. degree in Computer Engineering and Networking from the University of Missouri-Kansas City, Kansas City, MO, USA, in 2004. He is currently a professor at Hamad Bin Khalifa University (HBKU). His research interests include the use of machine learning in general and deep learning in particular in support of the data-driven and self-driven management of large-scale deployments of IoT and smart city infrastructure and services, Wireless Vehicular Networks (VANETs), cooperation, and spectrum access etiquette in cognitive radio networks, and management and planning of software defined networks (SDN). He is a senior member of the IEEE and an ABET Program Evaluator (PEV). He serves on editorial boards of multiple journals including IEEE Communications Letter and IEEE Network Magazine. He also served as chair, co-chair, and technical program committee member of multiple international conferences including IEEE VTC, IEEE Globecom, IEEE ICC, and IWCMC.

Abegaz Mohammed Seid (memeber, IEEE) received his B.Sc. and M.Sc. degrees in Computer Science from Ambo University and Addis Ababa University, Ethiopia, in 2010 and 2015, respectively. He received a Ph.D. degree in Computer Science and Technology from the University of Electronic Science and Technology of China (UESTC) in 2021. He is currently a post-doctoral fellow with the College of Science and Engineering at Hamad Bin Khalifa University, Doha, Qatar. He served as a graduate assistant and lecturer, as well as a member of the academic committee and an associate registrar at Dilla University, Ethiopia, from 2010 to 2016. Dr. Abegaz has published more than thirteen scientific conferences and journal papers. His research interests include a wireless network, mobile edge computing, blockchain, machine learning, Vehicular network, IoT, machine learning, UAV Network, IoT, and 5G/6G wireless network.

Aiman Erbad (Senior Member, IEEE) is an Associate Professor and ICT Division Head in the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). Prior to this, he was an Associate Professor at the Computer Science and Engineering (CSE) Department and the Director of Research Planning and Development at Qatar University until May 2020. He also served as the Director of Research Support responsible for all grants and contracts (2016–2018) and as the Computer Engineering Program Coordinator (2014–2016). Dr. Erbad obtained a Ph.D. in Computer Science from the University of British Columbia (Canada) in 2012, a Master of Computer Science in embedded systems and robotics from the University of Essex (UK) in 2005, and a B.Sc. in Computer Engineering from the University of Washington, Seattle in 2004. He received the Platinum award from H.H. The Emir Sheikh Tamim bin Hamad Al Thani at the Education Excellence Day 2013 (Ph.D. category). He also received the 2020 Best Research Paper Award from Computer Communications, the IWCMC 2019 Best Paper Award, and the IEEE CCWC 2017 Best Paper Award. His research received funding from the Qatar National Research Fund, and his research outcomes were published in respected international conferences and journals. He is an editor for KSII Transactions on Internet and Information Systems, an editor for the International Journal of Sensor Networks (IJSNet), and a guest editor for IEEE Network. He also served as a Program Chair of the International Wireless Communications Mobile Computing Conference (IWCMC 2019), as a Publicity chair of the ACM MoVid Workshop 2015, as a Local Arrangement Chair of NOSSDAV 2011, and as a Technical Program Committee (TPC) member in various IEEE and ACM international conferences (GlobeCom, NOSSDAV, MMSys, ACMMM, IC2E, and ICNC). His research interests span cloud computing, edge intelligence, Internet of Things (IoT), private and secure networks, and multimedia systems. He is a senior member of IEEE and ACM.

Octavia A. Dobre (Fellow, IEEE) received the Dipl. Ing. and Ph.D. degrees from the Polytechnic Institute of Bucharest, Romania, in 1991 and 2000, respectively. Between 2002 and 2005, she was with New Jersey Institute of Technology, USA. In 2005, she joined Memorial University, Canada, where she is currently a Professor and Research Chair. She was a Visiting Professor with Massachusetts Institute of Technology, USA and Universite de Bretagne Occidentale, France. Her ´ research interests encompass wireless, optical and underwater communication technologies. She has (co-)authored over 400 refereed papers in these areas. Dr. Dobre serves as the Editor-in-Chief (EiC) of the IEEE Open Journal of the Communications Society. She was the EiC of the IEEE Communications Letters, Senior Editor, Editor, and Guest Editor for various prestigious journals and magazines. She also served as General Chair, Technical Program CoChair, Tutorial Co-Chair, and Technical Co-Chair of symposia at numerous conferences. Dr. Dobre was a Fulbright Scholar, Royal Society Scholar, and Distinguished Lecturer of the IEEE Communications Society. She obtained Best Paper Awards at various conferences, including IEEE ICC, IEEE Globecom, IEEE WCNC, and IEEE PIMRC. Dr. Dobre is a Fellow of the Engineering Institute of Canada and a Fellow of the Canadian Academy of Engineering.

APPENDIX A
PROOF OF THEOREM 1

We recall the updated global model as follows:

$$\boldsymbol{W}(r+1) = \boldsymbol{W}(r) + \frac{1}{K} \sum_{k \in \Omega_r^m} \widehat{\boldsymbol{\mathcal{L}}}_k(r), \tag{31}$$

Let us define the following aux variable as in [38]:

$$\boldsymbol{W}'(r+1) = \boldsymbol{W}(r) + \frac{1}{K} \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{L}}}_k(r). \tag{32}$$

We can have:

$$\begin{aligned}
\|\boldsymbol{W}(r+1) - \boldsymbol{W}^*\|_2^2 &= \|\boldsymbol{W}(r+1) - \boldsymbol{W}'(r+1) + \boldsymbol{W}'(r+1) - \boldsymbol{W}^*\|_2^2 \\
&= \|\boldsymbol{W}(r+1) - \boldsymbol{W}'(r+1)\|_2^2 + \|\boldsymbol{W}'(r+1) - \boldsymbol{W}^*\|_2^2 + 2\langle \boldsymbol{W}(r+1) - \boldsymbol{W}'(r+1), \boldsymbol{W}'(r+1) - \boldsymbol{W}^* \rangle.
\end{aligned} \tag{33}$$

Now, we bound the average of the right hand side of (33).

Lemma 1. *We have the following optimally gap for the fairness scheduling algorithm:*

$$\mathbb{E}\left[\|\boldsymbol{W}(r+1) - \boldsymbol{W}'(r+1)\|_2^2 \right] \le \epsilon. \tag{34}$$

Proof. See Appendix B. □

Lemma 2. *Let $\mathbb{E}_{\mathcal{K}(r)}$ denote expectation over the client scheduling fairness at round r. We have*

$$\mathbb{E}_{\mathcal{K}(r)} [\boldsymbol{W}(r+1)] = \boldsymbol{W}'(r+1), \tag{35}$$

In light of this, it follows

$$\mathbb{E}_{\mathcal{K}(r)} [\langle \boldsymbol{W}(r+1) - \boldsymbol{W}'(r+1), \boldsymbol{W}'(r+1) - \boldsymbol{W}^* \rangle] = 0. \tag{36}$$

Proof. Since the client scheduling policy in our proposed algorithm is definite, it follows that

$$\mathbb{E}_{\mathcal{K}(r)} \left[\frac{1}{K} \sum_{k \in \mathcal{K}(r)} \widehat{\boldsymbol{\mathcal{L}}}_k(r) \right] \stackrel{\text{(a)}}{=} 1 \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{L}}}_k(r) = \frac{1}{K} \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{L}}}_k(r). \tag{37}$$

The proof of Lemma 2 is concluded from (37). □

Depending on the results of Lemmas 1 and 2, we have

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{W}(r+1) - \boldsymbol{W}^*\|_2^2 \right] &\le (1 - \alpha\eta_r (\mathcal{T} - \eta_r(\mathcal{T} - 1))) \mathbb{E}\left[\|\boldsymbol{W}(r) - \boldsymbol{W}^*\|_2^2 \right] \\
&\quad + \eta^2(r) (\mathcal{T}^2 + \mathcal{T} - 1) \varrho^2 \\
&\quad + (1 + \alpha(1 - \eta_r)) \eta^2(r)\varrho^2 \frac{\mathcal{T}(\mathcal{T} - 1)(2\mathcal{T} - 1)}{6} + 2\eta_r(\mathcal{T} - 1)\mathfrak{F} \\
&\quad + 2\eta_r \frac{1}{K} \sum_{k=1}^{K} \sum_{t=2}^{\mathcal{T}} \left(F_k^* - \mathbb{E}\left[F_k(\boldsymbol{W}_k^t(r)) \right] \right) + 2\eta_r \left(F^* - \mathbb{E}\left[F(\boldsymbol{W}(r)) \right] \right) \\
&\stackrel{\text{(a)}}{\le} (1 - \alpha\eta_r (\mathcal{T} - \eta_r(\mathcal{T} - 1))) \mathbb{E}\left[\|\boldsymbol{W}(r) - \boldsymbol{W}^*\|_2^2 \right] \\
&\quad + \eta^2(r) (\mathcal{T}^2 + \mathcal{T} - 1) \varrho^2 \\
&\quad + (1 + \alpha(1 - \eta_r)) \eta^2(r)\varrho^2 \frac{\mathcal{T}(\mathcal{T} - 1)(2\mathcal{T} - 1)}{6} + 2\eta_r(\mathcal{T} - 1)\mathfrak{F}.
\end{aligned} \tag{38}$$

In this case, (a) follows because $F^* - F(\boldsymbol{W}(r)) \le 0$, $\forall r$, and $F_k^* - F_k(\boldsymbol{W}_k^r) \le 0$, $\forall k, r$. According to (38), we conclude Theorem 1.

APPENDIX B
PROOF OF LEMMA 1

To prove Lemma 1, we take similar steps as [38, Appendix B.4]. We have

$$\mathbb{E}\left[\|\boldsymbol{W}(r+1) - \boldsymbol{W}'(r+1)\|_2^2 \right] = \mathbb{E}\left[\left\| \frac{1}{K} \sum_{k \in \mathcal{K}(r)} \widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r) \right\|_2^2 \right], \tag{39}$$

where

$$\widehat{\boldsymbol{\mathcal{L}}}(r) \triangleq \frac{1}{K} \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{L}}}_k(r). \tag{40}$$

We notice that the indicator $1(k \in \mathcal{K}(r)) = 1$ and $1(k' \in \mathcal{K}(r)) = 1$ due to the fairness amongst clients. As such, we eliminate its effects in the following proofs. We have

$$\mathbb{E}\left[\|\boldsymbol{W}(r+1) - \boldsymbol{W}'(r+1)\|_2^2\right] = \mathbb{E}\left[\left\|\frac{1}{K}\sum_{k=1}^{K}\left(\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\right)\right\|_2^2\right]$$
$$= \frac{1}{K^2}\mathbb{E}\left[\sum_{k=1}^{K}\left\|\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\right\|_2^2 \right.$$
$$\left. + \sum_{k=1}^{K}\sum_{k'=1,k'\neq k}^{K}\langle\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r), \widehat{\boldsymbol{\mathcal{L}}}_{k'}(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\rangle\right]. \tag{41}$$

Based on the symmetry, we can conclude that

$$\mathbb{E}_{\mathcal{K}(r)}\left[\sum_{k=1}^{K}\left\|\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\right\|_2^2\right] \overset{(a)}{=} \frac{\binom{K-1}{|\Omega_r^m|-1}}{\binom{K}{|\Omega_r^m|}}\sum_{k=1}^{K}\left\|\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\right\|_2^2$$
$$= \frac{K}{|\Omega_r^m|}\sum_{k=1}^{K}\left\|\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\right\|_2^2, \tag{42}$$

where (a) follows the fact that in the proposed scheduling approach every client index k, for $k \in \mathcal{K}$, appears r times before splitting, and

$$\mathbb{E}_{\mathcal{K}(r)}\left[\sum_{k=1}^{K}\sum_{k'=1,k'\neq k}^{K}\langle\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r), \widehat{\boldsymbol{\mathcal{L}}}_{k'}(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\rangle\right]$$
$$\overset{(b)}{=} \frac{\binom{K-2}{|\Omega_r^m|-2}}{\binom{K}{|\Omega_r^m|}}\sum_{k=1}^{K}\sum_{k'=1,k'\neq k}^{K}\langle\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r), \widehat{\boldsymbol{\mathcal{L}}}_{k'}(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\rangle$$
$$= \frac{|\Omega_r^m|(|\Omega_r^m| - 1)}{K(K-1)}\sum_{k=1}^{K}\sum_{k'=1,k'\neq k}^{K}\langle\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r), \widehat{\boldsymbol{\mathcal{L}}}_{k'}(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\rangle. \tag{43}$$

Due to the fairness scheduling:

$$\frac{|\Omega_r^m|(|\Omega_r^m| - 1)}{K(K-1)} = 1 \tag{44}$$

As a result, we substitute (42) and (43) into (41) which yields

$$\mathbb{E}\left[\|\boldsymbol{W}(r+1) - \boldsymbol{W}'(r+1)\|_2^2\right] = \frac{1}{K^2}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\right\|_2^2\right]$$
$$+ \frac{1}{K^2}\sum_{k=1}^{K}\sum_{k'=1,k'\neq k}^{K}\mathbb{E}\left[\langle\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r), \widehat{\boldsymbol{\mathcal{L}}}_{k'}(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\rangle\right]$$
$$\overset{(c)}{=} \frac{1}{K^2}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\right\|_2^2\right]$$
$$= \frac{1}{K^2}\left(\sum_{k=1}^{K}\mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{L}}}_k(r)\right\|_2^2\right] - \mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{L}}}(r)\right\|_2^2\right]\right)$$
$$\leq \frac{1}{K^2}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{L}}}_k(r)\right\|_2^2\right]$$
$$\overset{(d)}{=} \frac{1}{K^2}\sum_{k=1}^{K}\mathbb{E}\left[\|\boldsymbol{\mathcal{L}}_k(r)\|_2^2\right]$$
$$= \frac{1\eta^2(r)}{K^2}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\sum_{t=1}^{\mathcal{T}}\nabla F_k\left(\boldsymbol{W}_k^t(r), \mathfrak{D}_k^t(r)\right)\right\|_2^2\right]$$
$$\overset{(e)}{\leq} \frac{\eta^2(r)\mathcal{T}}{K^2}\sum_{k=1}^{K}\sum_{t=1}^{\mathcal{T}}\mathbb{E}\left[\left\|\nabla F_k\left(\boldsymbol{W}_k^t(r), \mathfrak{D}_k^t(r)\right)\right\|_2^2\right]$$
$$\overset{(f)}{\leq} \frac{\eta^2(r)\mathcal{T}^2\varrho^2}{K(K-1)}, \tag{45}$$

where

$$\left\|\sum_{k=1}^{K}\left(\widehat{\boldsymbol{\mathcal{L}}}_k(r) - \widehat{\boldsymbol{\mathcal{L}}}(r)\right)\right\|_2^2 = \epsilon, \epsilon \approx 0 \tag{46}$$

This is due to the convexity of the loss function, which proves that the proposed scheduling algorithm ensures the convergence to the optimal without any gap.