Data-Driven Participant Selection and Bandwidth Allocation for Heterogeneous Federated Edge Learning

Abdullatif Albaseer , *Student Member, IEEE,* Mohamed Abdallah, *Senior Member, IEEE,* Ala Al-Fuqaha, *Senior Member, IEEE,* and Aiman Erbad, *Senior Member, IEEE*

***Abstract*—Federated edge learning (FEEL) is a fast-growing distributed learning technique for next-generation wireless edge systems. Smart systems in different application domains suffer from data heterogeneity, limited wireless resources, and device heterogeneity, necessitating the need for intelligent participants' selection schemes that accelerate the convergence rate. Hence, this paper proposes joint participants selection and bandwidth allocation schemes to address these challenges. First, we formulate an optimization problem considering communication and computation latencies and imbalanced data distribution that meets a target round deadline and bandwidth constraints. To tackle participant selection combinatorial problems, we use a relaxation method followed by a proposed priority selection algorithm to select near-optimal participants. The proposed algorithm initially prioritizes participants with more data, effective channel states, and better CPU speed. To tackle data heterogeneity, we propose a randomized deadline controlling algorithm that diversifies the updates by enabling the edge server to involve various participants with small data samples into training rounds. The proposed algorithms provide near-optimal performance compared to the brute-force method. Experiments demonstrate that our proposed scheme accelerates the convergence rate by up to 55% under extensive non-i.i.d settings compared to benchmarks. Additionally, the controlling algorithm significantly improves the performance of the high data heterogeneity levels, resulting in faster FEEL systems.**

***Index Terms*—Federated Edge Learning, Edge Computing, Participants selection, Imbalanced Data Distribution, Data Diversity, Resource Allocation**

I. INTRODUCTION

NOWADAYS, devices at the wireless network edge produce enormous amounts of data, and extracting knowledge from this data is essential to build advanced AI-based applications in order to build more reliable intelligent systems [1]. Machine Learning (ML) and Deep Learning (DL) techniques specifically are being developed rapidly [2], [3] to exploit this data and induce advanced AI services in diverse domains, such as Intrusion Detection Systems [4], Industry 4.0 [5], and Animal–vehicle Collision Avoidance systems [6]. However, most of the current ML and DL approaches are limited to centralized algorithms, where a server consolidates all raw data to train a robust model. For example, online learning, as in [7], is a centralized ML technique where the model is initially built using currently available data. Then, the model is continuously updated when new samples arrive. Such centralized approaches are becoming increasingly costly since offloading high dimensional data from end devices to the edge server in intelligent systems is often infeasible due to limited wireless resources, latency, and privacy concerns [8], [9]. Therefore, the data generated at the edge devices needs to be stored and processed locally. To satisfy these needs, an emerging paradigm called *Federated Learning* (FL) has been recently introduced. FL trains and updates a shared model collaboratively without sharing the raw data [10], [11].

Federated Learning (FL) algorithms have recently been pushed towards the network edge, and the Federated Edge Learning (FEEL) systems are being developed to enable low latency edge intelligence where the data is produced. FEEL is a cutting-edge decentralized technique that enables edge devices to train machine learning models using real-time data and then send updates back to the coordinating edge server [12]. FEEL enables intelligent systems to collaboratively train a shared machine learning model while keeping all the training data on edge devices, thereby decoupling the ability to do machine learning from the need to upload/store data in the cloud. It aims to leverage the massive amount of data collected by edge devices in real-time without compromising their privacy [13].

However, FEEL encounters two major challenges: one is data heterogeneity, i.e., imbalanced and non-i.i.d. data distribution and the other is resource heterogeneity. As for the data heterogeneity, data is massively distributed among clients, i.e., participating devices in a non-i.i.d. and imbalanced manner. Each device in the network, in particular, has its own data, which varies in size, labels, and forms. In general, ML models require a large amount of data from a variety of sources in order to be successful in decision-making which requires considerable efforts to address such a challenge. As for the resource heterogeneity, devices have varying computing and communication capabilities, while the network has limited bandwidth. For example, uploading of models updates in FEEL consumes significant bandwidth as DL models contain billions of parameters [14]. Therefore, the local updates by thousands of transmitting edge participants might simply congest the air interface, making it a bottleneck for efficient edge learning. This situation is further exacerbated by the fact that the edge server needs to wait for the arrival of all the updates. Hence, due to varying computing capabilities, the global update synchronization requirement in such a system incurs an unnecessary delay due to the idle time wasted while

Abdullatif Albaseer, Mohamed Abdallah, Ala Al-Fuqaha, and Aiman Erbad are with the Division of Information and Computing Technology, College of science and engineering, Hamad Bin Khalifa University, Doha, Qatar (e-mail: {amalbaseer, moabdallah, aalfuqaha, AErbad}@hbku.edu.qa).

waiting for the "stragglers" (i.e., the devices with slower processors, bad channels, having large data size etc.). As a solution, the server should specify the optimal deadline for synchronizing the global model.

As per the above remarks, designing a joint participant selection and bandwidth allocation for FEEL in large-scale networks is challenging. Thus, a key question is how to optimally and efficiently select participants and allocate edge resources to accelerate the convergence rate while considering all of the aforementioned challenges.

In response, many existing selection approaches [15]–[20] have been proposed to either select the participants randomly or select the participants with shorter update time and then allocate the resources accordingly. For example, the works in [15], [18]–[20] introduced greedy-based approaches trying to find the participants that provide the least updating time regardless of the size of local data samples. However, first, these works [15], [18]–[20] assumed that the data distribution is balanced and of similar data size among clients, which doesn't reflect core assumptions of FL that generally have imbalanced and massively distributed non-i.i.d data. Second, these approaches can not guarantee the optimal or even near-optimal solution. Third, the data diversity is not considered where setting a fixed deadline leads to having a biased model for the dominant participants that are always selected. Fourth, the works in [15], [18]–[20] did not account for the combinatorial nature of participant selection and its adverse effects, especially with large-scale edge network systems where a greedy based selection algorithm is used to select the participants. Therefore, it is necessary to develop novel and efficient approaches that account for all these gaps. To this end, this takes into account the aforementioned challenges and introduces novel approaches. First, we account for the imbalanced and non-i.i.d. data distribution. Second, we introduce an algorithm that ensures near-optimal solutions. Third, we propose a dynamic deadline controlling algorithm to address the data diversity. Fourth, we account for the combinatorial nature of participant selection using the relaxation method. Our specific contributions can be delineated as follows:

- Formulate an online joint optimization problem to select the optimal participants and allocate the resources considering the imbalanced data distribution among participants. In each round, participants are selected based on a priority metric; the local data size, the channel state, and the local computation speed.
- Propose a priority selection algorithm to find the list of participants with low time complexity in a polynomial time.
- Introduce a dynamic deadline controlling algorithm instead of a fixed deadline as in state-of-the-art [15], [18]–[20] to tackle the heterogeneity and non-i.i.dness. As a consequence, the edge servers can choose different participants and aggregate different updates during global training round resulting unbiased trained model.
- Reformulate the problem using a relaxation method to tackle the combinatorial nature of participant selection and its adverse effects, especially when dealing with large-scale edge systems.
- Perform a theoretical analysis to show the relationship between the convergence rate of the global model and the number of selected participants weighted to the number of local data samples.
- Assess the performance of our proposed scheme using realistic federated datasets under non-i.i.d settings. We benchmark our results with the state-of-the-art [15], [18]–[20]. Simulation experiments demonstrate that the convergence time is significantly reduced, and the performance is substantially improved. The source codes and the datasets are available at https://github.com/Abdullatif2/FL_Particpant_selection_Based_Fixed_and_dynamic_deadline.

The remainder of the paper is organized as follows: related works are discussed in Section II. The system model, federated learning preliminaries, and definitions are then introducedin Section III. Afterwards, the problem formulation is given in Section IV. The supported theory is given in detail in Section V. Section VI introduces the proposed solutions where we present the complexity, optimality, and implementation of the algorithms. We present the experimental setup, results, and discussion in Section VII alonge with most important lessons learned. Finally, we conclude our work and present the future research directions in Section VIII.

II. RELATED WORK

Previous studies [21]–[27] addressed many of the challenges associated with the use of FL over wireless channels. For instance, to address the latency problem, the broadband analog aggregation (BAA) scheme was proposed in [23]–[26] to reduce the transmission time between edge devices and the orchestrator server by utilizing the superposition property of wireless channels via over-the-air computation (AirComp) [27]. Furthermore, considering a limited bandwidth over multiple fading channels, a distributed stochastic gradient descent scheme was investigated in [30], where each device is selected opportunistically for transmission based on the channel conditions. Earlier work assumed perfect updates-uploading, representing an approach to address the communication-latency challenge in federated edge learning systems. However, the effects of wireless channels are not considered. Hence, to support low-latency federated edge learning from the communication perspective, a novel bandwidth allocation strategy was proposed in [21]. Subsequently, given limited radio resources, namely, channel bandwidth, the BAA scheme was fine-tuned for Gaussian channels in [30]. To be specific, the edge devices first determine the sparsity of the updates (gradients) and then project them to a lower-dimensional space imposed by the available channel bandwidth before transmission.

Focusing on the participant's selection scheduling policies, the work in [16] studied three scheduling policies and their effects on the convergence rate. The works in [15] and [18] proposed greedy algorithms to select the participants that provide less updating and uploading time. However, if the data size is imbalanced, the participants with large data sizes are not considered as they need more time to train their models despite

TABLE I: RELATIONSHIP BETWEEN OUR WORK AND THE RECENT LITERATURE

Ref	Imbalanced Data samples	Non-i.i.d class-distribution	Data Heterogeneity	Devices Heterogeneity	Control the deadline	Channel Uncertainty
Nishio et al. [15]	✗	✓	✗	✓	✗	✗
Shi et al. [18]	✗	✓	✗	✓	✗	✓
Chen et al. [19]	✗	✗	✗	✓	✗	✓
Xu et al. [20]	✗	✓	✗	✓	✗	✓
Wang et al. [28]	✗	✓	✗	✓	✗	✗
Anh et al. [29]	✗	✗	✗	✓	✗	✗
Our work	✓	✓	✓	✓	✓	✓

their strong effects on the convergence rate. A new approach was proposed by Chen et al. [19] to minimize the convergence time using artificial neural networks (ANN) to predict the clients' updates not involved in the learning round. In addition, they proposed that the base station stays connected with the clients provided less value of the loss function. However, the clients with few data samples will continuously produce less value of the loss function, and the clients having more data deliver a large value of the loss function as they need more local iteration to converge.

To conclude, as illustrated in Table I, these series of prior works [15], [18]–[20] assumed that the data distribution is balanced and of similar data size among clients, which does not reflect the core assumptions of FL. These assumptions do not reflect existing scenarios of FL that generally have imbalanced and massively distributed non-i.i.d data. The works in [15], [18]–[20] introduced greedy-based approaches trying to find the participants that provide the least updating time regardless of the size of local data samples. These approaches can not guarantee the optimal or even near-optimal solution. In addition, the data diversity is not considered leading to have a biased model for the dominant participants that are always selected. However, none of these approaches can guarantee an optimal or even near-optimal solution. Additionally, data diversity is not considered, resulting in a biased model for the participants who are always selected. Moreover, the works in [15], [18]–[20] did not account for the combinatorial nature of participant selection and its adverse effects, especially with large-scale edge network systems where a greedy based selection algorithm is used to select the participants. Last, some works that use a deadline constraint to choose the participants, as in [15], considered a fixed deadline where the same participants are selected during the training rounds.

III. SYSTEM MODEL

This section discusses the fundamentals of FEEL, data producers, and the computation and communication models. Table II summarizes the main notations used throughout this paper.

In this work, we consider a wireless edge network as depicted in Fig. 1 with a single edge server connected to an edge cell (i.e., base station) that wirelessly communicates with K edge devices. Each edge device $k \in \{1, 2, ..., K\}$, has its local dataset $\mathcal{D}_k$ where $\mathcal{D}_k = \{x_{k,d} \in \mathbb{R}^d, y_{k,d} \in \mathbb{R}\}$, and $|\mathcal{D}_k|$ is the number of local data samples. Here $x_{k,d}$ is the d-dimensional input data vector at k-*th* participant, and $y_{k,d}$ is

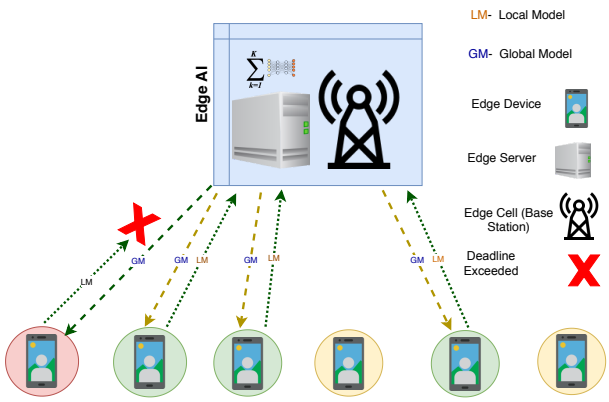


Fig. 1: Federated Edge Learning System with Participants Selection under Data and Resource Heterogeneity.

the corresponding label associated with $x_{k,d}$. The data among the participants is imbalanced depending on their activity. Each edge participant k trains its local model (**LM**) using its data $\mathcal{D}_k$ without exposing it to other participants or the edge server. To collaborate with others, the FEEL system enables edge devices to share the gradients and weights $\boldsymbol{\theta}_k$ with a central edge server (i.e., Base station) that aggregates and fuses all updates to form a new global model (**GM**) using federated averaging (i.e., FedAvg) [31]. In the FEEL system, the server can not involve all available devices in the learning round due to wireless networks' characteristics (i.e., bandwidth limitation, channel uncertainty, fading) and the deadline constraint set to avoid a long waiting time for a model update. Thus, only a subset S_r, where S_r is the indices vector that includes the selected participant's ids that are eligible to participate in a particular FEEL round. This leads to a challenge of optimally selecting the participants that accelerate the convergence rate considering the deadline requirement set by the server and available network resources. Moreover, unlike conventional decentralized ML, the FEEL system's data is unbalanced and distributed in a non-i.i.d fashion, resulting in increased data heterogeneity across the network, which necessitates the design of an efficient selection technique that addresses all these challenges.

A. Local loss function

This function captures the error of the model on the dataset $\{x_{k,d}, y_{k,d}\}$ for each k. For local updates at every round r, the loss function is expressed as follows:

$$F_k^r(\boldsymbol{\theta}) \triangleq \frac{1}{|\mathcal{D}_k|} \sum_{s \in \mathcal{D}_k} F_s(\boldsymbol{\theta}), \tag{1}$$

TABLE II: SYMBOLS AND NOTATION

D	The total dataset across devices
$\mathcal{D}_k$	The dataset of k-*th* participant
$x_{k,d}$	the d-dimensional input data vector
$y_{k,d}$	the corresponding label associated with $x_{k,d}$
b	The batch size
$\mathcal{E}$	The number of local epochs
S	The selection matrix that include all selected participants indices for all rounds
S_r	A vector representation of a subset of devices selected to join FEEL round r
$S_r^{(k)}$	A selection indicator for the k-*th* participant at r-*th* round
K	The available total number of participants
$\boldsymbol{\theta}^r$	The global model parameters at r-*th* round
$\boldsymbol{\theta}_k^r$	The local model parameters of the k-*th* participant at r-*th* round
$\boldsymbol{\theta}_k^r(i)$	The local update at local iteration i
$F_k^r(\boldsymbol{\theta})$	The local loss function of participant k at r-*th* round
$F(\boldsymbol{\theta})$	The global loss function
$F_s(\boldsymbol{\theta})$	The loss function within the local data samples that captures the local error over each sample s
T	The deadline requirement set by server
$T_{k,r}^{\text{cmp}}$	The local computation time of participant k at r-*th* round
$T_{k,r}^{\text{up}}$	The upload link time of k-*th* participant needed to send the model to the edge server at r-*th* round
τ	The Time frame
R	The number of communication rounds
h_k^r	The channel gain between the k-*th* participant and the edge server at r-*th* round
p_k^r	The transmit power of the the k-*th* participant at r-*th* round
N_0	The Noise spectral density (Gaussian Noise)
B	The total bandwidth
$\mathcal{R}_r^k$	The achievable transmission rate in bits per second (bits/s) of k-*th* participant sent to the edge server at r-*th* round
η	The learning rate
ϵ	An arbitrary constant $0 \leq \epsilon \leq 0.1$ to specify whether the model converge to the optima or not.
δ_k	The weighted number of local data samples
ξ	The size of the model parameters sent to the server in bits
$\mathbf{n}$	The number of local updates

where $F_s(\boldsymbol{\theta})$ captures the local error over each sample s. Hence, the total data across the edge network can be defined as: $D \triangleq \sum_{k=1}^{K} |\mathcal{D}_k|$, and δ_k denote the weighted number of local data samples defined as follows,

$$\delta_k = \frac{|\mathcal{D}_k|}{D}. \tag{2}$$

To train the model, each k-*th* participant runs its local solver locally, such as stochastic gradient descent (SGD) to minimize the loss function defined in (1) simultaneously for several local epochs denoted by $\mathcal{E}$ [1]. For example, for a minibatch SGD, the total number of local updates is defined as:

$$\mathbf{n} = \mathcal{E}\ \frac{|\mathcal{D}_k|}{b}, \tag{3}$$

where b is a batch-size determining a subset of the training set required for one local update. Namely, the local model parameters $\boldsymbol{\theta}_k$ are updated as:

$$\boldsymbol{\theta}_k^r(i) = \boldsymbol{\theta}_k^r(i-1) - \eta \nabla F_k^r(\boldsymbol{\theta}_k^r(i)), \tag{4}$$

where $i = 1, 2, \ldots, \mathbf{n}$ is the number of local updates performed by participant k and η is the step size (i.e., learning rate) at each round, $\boldsymbol{\theta}_k^r(0)$ are the global parameters $\boldsymbol{\theta}^{r-1}$ received from server and $\boldsymbol{\theta}_k^r(\mathbf{n})$ are the local parameters update $\boldsymbol{\theta}_k^r$ sent back to the server.

[1]GD method is logical when every participant has a small number of data samples, i.e, $|\mathcal{D}_k| << D$, $\forall\ S_r$ while for a large portion of data samples $\mathcal{D}_k$, SGD could be adopted to relieve the computation load on the participant, though the rate of convergence is different.

B. Global loss function

Once all local models $\boldsymbol{\theta}_k^r$ and local loss functions $F_k^r(\boldsymbol{\theta})$ are calculated locally using (1) and (4), and are uploaded to the server, the global loss function across participants at every round r is computed as:

$$F^r(\boldsymbol{\theta}) \triangleq \sum_{k=1}^{K} \delta_k F_k^r(\boldsymbol{\theta}). \tag{5}$$

Accordingly, the global model parameters are computed as follow:

$$\boldsymbol{\theta}^{\mathbf{r}} = \sum_{k=1}^{K} \delta_k \boldsymbol{\theta}_k^r. \tag{6}$$

It is worth mentioning that $F^r(\boldsymbol{\theta})$ and $\boldsymbol{\theta}^{\mathbf{r}}$ are sent to all selected participants at each round to be used as a reference when updating the model parameters. Thus, the aim is to find $\boldsymbol{\theta}$ so as to minimize $F(\boldsymbol{\theta})$

$$\boldsymbol{\theta}^* \triangleq \arg\min F(\boldsymbol{\theta}). \tag{7}$$

In machine learning models, it is difficult to derive a closed-form solution for (7). So, GD is used to approach the solution iteratively.

The FEEL system algorithms aim to satisfy that the iterative minimization of the loss function is closer to the optimal-value of loss function $F(\boldsymbol{\theta}^*)$ assuming that $(\boldsymbol{\theta}^*)$ is optimal model parameters. This difference is defined as an arbitrary constant $0 \leq \epsilon \leq 0.1$:

$$F^R(\boldsymbol{\theta}) - F(\boldsymbol{\theta}^*) \leq \epsilon. \tag{8}$$

C. Local Computation model

For model updates in the FEEL system, the participants are constrained by a training round deadline T based on scheduling policies to upload their models to the server. The server initiates the model parameters $\boldsymbol{\theta}_0$ and then sends them to all selected participants. Then, each k-*th* participant receives the model parameters and then updates them using its local data; then, all updates sends back to the server, which in return aggregates and fuses all updates to form new global model parameters. The new global model is being broadcasted again for further updates. These steps are iteratively repeated until the global model converges. Thus, when training identical models on participant's devices with either a small portion of data samples or extremely weak computation capacities, such devices will send undesirable updates or delay the entire collaboration cycle, thereby hindering the ability to produce a robust global model with faster convergence rate. To update the model parameters $\boldsymbol{\theta}^r$, the participant's data $\mathcal{D}_k$ is split into batches based on a predetermined batch size b , and the number of epochs $\mathcal{E}$. Let ϕ denote the number of local computation cycles required to process one data sample, and let f_k denote

the local processor's speed (cycles per second). Accordingly, the local computation time of each participant k in one round r can be defined as: $T_{k,r}^{\text{cmp}} = \frac{\phi|\mathcal{D}_k|}{f_k}$ Subsequently, each device runs the local solver for $\mathbf{n}$ iterations at each round; thus, $T_{k,r}^{\text{cmp}}$ can be rewritten as follows:

$$T_{k,r}^{\text{cmp}} = \frac{\mathbf{n}\phi b}{f_k}. \tag{9}$$

D. Communication model:

We consider Time Division Multiple Access (*TDMA*) for uploading local models with a total bandwidth of B, and the total time frame τ divided between the selected participants. We can note that TDMA isn't restrictive and other access techniques, such as orthogonal frequency-division multiple access (OFDMA), are applicable as well. For example, we can use the number of OFDMA sub-channels instead of time slots. Then each sub-channel performs Carrier-sense multiple access with collision avoidance (CSMA/CA) independently to reduce the probability of collision occurrence and enhance the system throughput. Based on Shannon's theorem, the achievable transmission rate in bits sent to the edge server can be defined as:

$$\mathcal{R}_r^k = B \ \log_2\left(1 + \frac{p_k^r||h_k^r||^2}{N_0}\right), \forall k, \tag{10}$$

where h_k is the channel gain between the edge server and k-*th* participant assuming that the channel between the participant and the BS is constant within the duration of the r-*th* round, and N_0 is the spectral noise power. Next, let ξ denote the model size in bits, hence the uplink latency for each k-*th* participant is:

$$T_{k,r}^{\text{up}} = \frac{\xi}{\mathcal{R}_r^k} \leq \tau_k, \ \forall k, \tag{11}$$

where τ_k is the time slot assigned to participant k by the edge server. It is worth noting that the inequality in (11) is to ensure that the uploading delay should be less than the allocated time slot. Throughout this paper, we only consider the uplink latency, and we assume the downlink latency is negligible due to the powerful capabilities and the sufficient bandwidth of the edge server. We note that the computation time is proportional to the collected data and CPU frequency as in (9). Also, the uplink latencies depend on the channel state, transmission power, and model size as in (11). Accordingly, we define the total latency of k-*th* participant at each round as : $T^{k,r} = T_{k,r}^{\text{cmp}} + T_{k,r}^{\text{up}}, \ \forall k$ while the total latency including TDMA is defined as: $\max(T_{k,r}^{\text{cmp}}) + \tau$ which includes the maximum computation time and time frame of TDMA. Practically, the server has to enforce a pre-determined deadline T to synchronize the updates and avoid long waiting times. Thus, the k-th participant must finish its tasks within T. Hence, the total latency should satisfy this condition:

$$(T_{cmp}^{k,r} + T_{ul}^{k,r}) \leq T, \quad \forall k, \forall r \tag{12}$$

IV. PROBLEM FORMULATION

In large-scale edge networks, selecting FEEL system participants that capture the imbalanced data distribution, round deadline and, available resources is crucial. The distribution of data across the network is typically skewed depending on the activity of each k-*th* participant, resulting in varying quality of model updates. In addition, the data is distributed in non-i.i.d., inducing the need for involving as many participants as possible to improve the global model. Moreover, the devices among participants are heterogeneous, having different computation capabilities. Consequently, a key question is how to select the participants optimally and efficiently use available edge resources and data to accelerate the convergence rate, considering the limited wireless resources, local computation capabilities, the deadline requirement, and imbalanced data samples. While the studies in [27], [18], [22], [32]–[34] showed that adding more participants improves the FEEL system performance; however, they assumed that all devices hold the same local data size, and they didn't address the impacts of imbalanced data distribution on the FEEL system performance, round deadline, and resource allocation. Following the above definition of FEEL system learning, we can narrow the question to find the optimal participants that accelerate the convergence rate of global loss, $F(\boldsymbol{\theta})$, tackle the data heterogeneity, and diversify the updates during the FEEL system rounds subject to the constraints mentioned earlier.

Having defined the system model, we can notice that only a subset of the edge devices $|S_r| \leq K$ are selected by the edge server at r-*th* round to train the global model. We aim to maximize the number of participants at every round that accelerates the convergence rate to find the minimum loss, $F(\boldsymbol{\theta})$, as well as increase the data utilization. Due to limited resource blocks, selecting a large number of participants is inapplicable; instead, we choose the optimal participants holding more data samples due to its impacts on training convergence, as explained in section V. Specifically, our goal is to select the optimal subset of participants during training round that accelerate the convergence rate while considering data heterogeneity, imbalanced-data distribution, data diversity, deadline and limited bandwidth. Let S denote the selection matrix that include all selected participants indices for all rounds where $S = \{S_r\}_{r=1,2,\ldots,R}$. It is worth noting that the number of rounds R is chosen to be a constant large enough to reach convergence. Then, we define a binary integer variable for each participant k to specify whether it is selected or not so that indices vector at r-*th* round can be defined as $S_r = [S_r^{(1)}, \ldots, S_r^{(k)}, \ldots, S_r^{(k)}]$ where:

$$S_r^{(k)} = \begin{cases} 1, & \text{if the } k\text{th participant is selected at round } r, \\ 0, & \text{Not selected.} \end{cases} \tag{13}$$

Accordingly, we can formulate the optimization problem as follows:

$$\mathbf{P1}: \max_{S} \sum_{r=1}^{R}\sum_{k=1}^{K} \delta_k S_r^{(k)}, \tag{14}$$

$$\text{s.t.:} \quad F^R(\boldsymbol{\theta}) - F(\boldsymbol{\theta}^*) \leq \epsilon, \tag{15}$$

$$\sum_{k=1}^{K} S_r^{(k)} T_{k,r}^{\text{up}} \leq \tau, \forall r, \tag{16}$$

$$S_r^{(k)}(T_{k,r}^{\text{cmp}} + T_{k,r}^{\text{up}}) \leq T, \quad \forall k, \forall r, \tag{17}$$

$$S_r^{(k)} \in \{0,1\}, \qquad \forall k, \forall r. \tag{18}$$

Constraint (15) guarantees that the trained global model converges to the optimal model $\boldsymbol{\theta}^*$. Constraint (16) is set the ensure that the bandwidth is allocated to the optimal participants. Constraint (17) is related to the deadline constraint where the selected participants should accomplish the updating and uploading tasks within T. Finally, constraint (18) is the selection binary variable. The goal of **P1** is to maximize the number of selected participants holding large data samples at all rounds that guarantee the convergence of the global loss $F(\boldsymbol{\theta})$ as well as meet the required constraints. However, solving **P1** is challenging as (15) requires to know the optimal value for the loss function $F(\boldsymbol{\theta}^*)$ which needs to have the entire datasets in a single unit and also it lacks the future information for the value of the loss function $F^R(\boldsymbol{\theta})$ at the last round. Moreover, the uploading time $T_{k,r}^{\text{up}}$ and the local computation time $T_{k,r}^{\text{cmp}}$, defined in (9) and (11) vary over rounds period. Last, some devices might be switched off or depleted of their energy. Thus, it isn't easy to proactively find participant's indices during all training rounds. Therefore, in section VI, we solve **P1** using the following steps. First, due to the difficulty of finding a closed-form solution for many DL algorithms, constraint (15) is eliminated. As a consequence, as shown in Section V, (15) is recursively solved, and the gap is reduced by finding the optimal participants who hold more local data samples. Then, for every r-*th* round, we reformulate a joint resource allocation and participant selection. Second, we address the combinatorial nature of participant selection and its computational complexity by finding a lower bound solution for the problem by employing a *relaxation* method that makes the selection constraint less restrictive followed by priority selection algorithm to utilize the relaxed-based solution and perform the FEEL process. Last, to tackle the data heterogeneity, a dynamic deadline controlling algorithm is proposed to diversify the updates through training rounds.

V. IMPACTS OF DATA AND PARTICIPANTS SELECTION ON CONVERGENCE RATE

In this section, we prove how the data and the selection of participants affect the convergence rate. To begin with, we use the following assumption for all k,

Assumption 1. *$F_s(\cdot)$ is $\mathcal{L}$-smooth $\forall s \in \mathcal{D}_k$, and $F_k(\cdot)$ is $\mathcal{L}$-smooth and β-strongly convex $\forall k$ and $\forall \boldsymbol{\theta}, \boldsymbol{\theta}^* \in \mathbb{R}^d$, respectively [21], [35], as follows:*

$$\left\|\nabla F^r(\boldsymbol{\theta}^r) - \nabla F(\boldsymbol{\theta}^*)\right\| \leq \mathcal{L}\left\|(\boldsymbol{\theta}^r - \boldsymbol{\theta}^*)\right\|. \tag{19}$$

$$F^r(\boldsymbol{\theta}^r) \leq F(\boldsymbol{\theta}^*) + \left\langle \nabla F(\boldsymbol{\theta}^*), \boldsymbol{\theta}^r - \boldsymbol{\theta}^* \right\rangle + \frac{\mathcal{L}}{2}\left\|\boldsymbol{\theta}^r - \boldsymbol{\theta}^*\right\|^2. \tag{20}$$

$$F^r(\boldsymbol{\theta}^r) \geq F(\boldsymbol{\theta}^*) + \left\langle \nabla F(\boldsymbol{\theta}^*), \boldsymbol{\theta}^r - \boldsymbol{\theta}^* \right\rangle + \frac{\beta}{2}\left\|\boldsymbol{\theta}^r - \boldsymbol{\theta}^*\right\|^2, \tag{21}$$

where $\langle\ \boldsymbol{\theta}, \boldsymbol{\theta}^*\ \rangle$ denote the inner product of vectors $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$ and $\|\cdot\|$ is the Euclidean norm. The strong convexity and smoothness in Assumption 1, has been also used in [21], [28], [36], and it exists in a variety of applications (i.e., l_2-regularized linear regression model). Given Assumption 1, and the definition of $F(\boldsymbol{\theta})$ in section III-A, we have:

$$F(\boldsymbol{\theta}^r) - F(\boldsymbol{\theta}^*) \leq \frac{\beta}{2}\left\|\boldsymbol{\theta}^r - \boldsymbol{\theta}^*\right\|^2. \tag{22}$$

Theorem 1. *For any selected participants and optimal solution $\boldsymbol{\theta}^*$, we have:*

$$\frac{\beta}{2}\left\|\boldsymbol{\theta}^r - \boldsymbol{\theta}^*\right\|^2 = \frac{\beta}{2}\frac{\left\|\sum_{k=1}^K S_r^{(k)}|\mathcal{D}_k|(\boldsymbol{\theta}_k^r - \boldsymbol{\theta}^*)\right\|^2}{\left(\sum_{k=1}^K S_r^{(k)}|\mathcal{D}_k|\right)^2}. \tag{23}$$

Proof. See Appendix A. □

We can notice that if the data is balanced among participants, we can rewrite (23) as follows:

$$\frac{\beta}{2}\left\|\boldsymbol{\theta}^r - \boldsymbol{\theta}^*\right\|^2 = \frac{\beta}{2}\frac{\left\|\sum_{k=1}^K S_r^{(k)}\mathcal{D}_c(\boldsymbol{\theta}_k^r - \boldsymbol{\theta}^*)\right\|^2}{\left(|S_r||\mathcal{D}_c|\right)^2}, \tag{24}$$

where $|\mathcal{D}_c|$ is the number of balanced data samples. In FEEL system, however, the data is imbalanced, and we must use (23) to reflect the realistic distribution of the data across the network.

Now let us recall Equ. (4), we have:

$$\boldsymbol{\theta}_k^r = \boldsymbol{\theta}^{r-1} - \eta\sum_{i=1}^{\mathbf{n}}\nabla F_k^r(\boldsymbol{\theta}_k^r(i)). \tag{25}$$

Theorem 2. *For any k-th participant performing $\boldsymbol{n}$ local updates, $F_k^r(\boldsymbol{\theta}_k^r(0))$, $F_k^r(\boldsymbol{\theta}_k^r(1))$, ..., $F_k^r(\boldsymbol{\theta}_k^r(\boldsymbol{n}))$ is a decreasing function and its value is inverse proportional to the data size.*

$$\begin{aligned} F_k^r(\boldsymbol{\theta}_k^r(\boldsymbol{n})) - F(\boldsymbol{\theta}^*) \leq F_k^r(\boldsymbol{\theta}_k^r(\boldsymbol{n}-1)) - F(\boldsymbol{\theta}^*) \\ \leq F_k^r(\boldsymbol{\theta}_k^r(\boldsymbol{n}-2)) - F(\boldsymbol{\theta}^*) \cdots \leq F_k^r(\boldsymbol{\theta}_k^r(0)) - F(\boldsymbol{\theta}^*), \end{aligned} \tag{26}$$

$$F_k^r(\boldsymbol{\theta}_k^r) - \nabla F_k(\boldsymbol{\theta}^*) := \frac{\left\|\boldsymbol{\theta}_k^r(0) - \boldsymbol{\theta}^*\right\|_2^2}{2\eta\mathbf{n}}. \tag{27}$$

Proof. See Appendix B. □

From (26), we can notice that as the number of local data points increases, the divergence between $\boldsymbol{\theta}_k^r$ and $\boldsymbol{\theta}^*$ decreases.

Lemma 1. *The convergence of the global loss function at round r is given by [21]:*

$$\begin{aligned} \mathbb{E}\left[F\left(\boldsymbol{\theta}^{r+1}\right) - F\left(\boldsymbol{\theta}^*\right)\right] \leq & \frac{2c_1}{LD}\sum_{k=1}^K |\mathcal{D}_k|\left(1 - S_r^{(k)}\right) + (1 - \\ & \frac{\beta}{\mathcal{L}} + \frac{4\beta c_2}{LD}\sum_{k=1}^K |\mathcal{D}_k|\left(1 - S_r^{(k)}\right)) \\ & \mathbb{E}\left(F\left(\boldsymbol{\theta}_k^r\right) - F\left(\boldsymbol{\theta}^*\right)\right). \end{aligned} \tag{28}$$

Proof. See Appendix C. □

We can notice that the upper bound of the gap between the left-hand side and the right-hand side in (28) is

$\frac{2c_1}{LD}\sum_{k=1}^{K}|\mathcal{D}_k|\left(1-S_r^{(k)}\right)$. This gap can be reduced by pushing more data and selecting more participants and hence, accelerating the convergence rate as the gap between the optimal model $\boldsymbol{\theta}^*$ and the trained model $\boldsymbol{\theta}^r$ is inversely proportional to the number of participants weighted to local data samples as in (28). From (26), (27), and (28) in the above Theorems, we note that selecting the participants with large datasets increases the number of local updates, reduces the upper bound gap and in turn, accelerates the convergence towards the optimal model $\boldsymbol{\theta}$*. Hence, maximizing the number of participants weighted to the local data samples can minimize the value of the loss function of the global model.

VI. PARTICIPANTS SELECTION AND RESOURCE ALLOCATION FOR FAST CONVERGENCE RATE

As shown in Section V, using more data or increasing the number of participants accelerate the convergence rate. However, due to limited wireless bandwidth, the latter is infeasible. To this end, we aim to maximize the number of participants at each round r weighted to the number of local data samples aiming to push more data to increase the number of local updates and accelerate the convergence rate. Each k-*th* participant is weighted based on its local data volume as in (2). We take into account the local computation latency, uplink latency, and available bandwidth at every round. Thus, **P1** is solved iterativly at each round to select the optimal participants that have more data samples and provide less updating and uploading time. The problem can be formulated as follows:

$$\textbf{P2:} \quad \max_{S_r} \quad \sum_{k=1}^{K}\delta_k S_r^{(k)}, \tag{29}$$

$$\text{s.t.:} \quad \sum_{k=1}^{K} S_r^{(k)} T_{k,r}^{\text{up}} \leq \tau, \tag{30}$$

$$S_r^{(k)}(T_{k,r}^{\text{cmp}} + T_{k,r}^{\text{up}}) \leq T, \quad \forall k, \tag{31}$$

$$S_r^{(k)} \in \{0,1\}, \qquad \forall k. \tag{32}$$

In particular, **(29)** aims to select the participants having more data samples while respecting constraints (30) and (31). In particular, constraint (32), $S_r^{(k)} \in \{0,1\}$, is relaxed as fractional real-value constraint $0 \leq S_r^{(k)} \leq 1$. We can note that the fractional real-value of $S_r^{(k)}$ can be seen as a selection priority for each participant k. Mathematically, **(29)** after relaxation can be rewritten as follows:

$$\textbf{P3:} \quad \max_{S_r} \quad \sum_{k=1}^{K}\delta_k S_r^{(k)}, \tag{33}$$

$$\text{s.t.:} \quad \sum_{k=1}^{K} S_r^{(k)} T_{k,r}^{\text{up}} \leq \tau, \tag{34}$$

$$S_r^{(k)}(T_{k,r}^{\text{cmp}} + T_{k,r}^{\text{up}}) \leq T, \quad \forall k, \tag{35}$$

$$S_r^{(k)} \in [0,1], \qquad \forall k. \tag{36}$$

We can note that **P3** is a convex problem and it can be solved using a numerical method. In this work, we use Gurobi optimizer (i.e., a suite of solvers for mathematical programming) [37] on the server-side to solve **P3**. Gurobi optimizer has many advanced algorithms that are more efficient for joint optimization and large-scale inputs than conventional techniques (i.e., Hungarian algorithm). We proposed a priority selection algorithm called *priority selection algorithm* that utilizes the solutions of **P3** for efficient selection and allocation in FEEL. Particularly, we utilize the outcomes of the **P3** solution to select the optimal participants as shown in Algorithm 1. Algorithm 1 selects the participants at each round based on their priority (i.e., fractional real value $S_r^{(k)}$). In Algorithm 1, step 1, the server initializes the parameters of the global model and then starts the training rounds as in step 2. In step 3, the server collects all required information from available devices to select the optimal participants to update the model. Steps 5 and 6 initialize the selection vector and estimate the computation and communications latencies. Steps 7-8, solve **P3** to select the optimal participants that carry out the model parameters update and then sort them in a descending order based on their fractional real-values resulting from the solution of **P3**. In steps 9-15, the server starts with a minimum number of participants as in (39) and iterates to check the possibility of adding more participants if the available resources are not exhausted. Finally, in steps 16-19, the server broadcasts the current model parameters to the selected participants to update the parameters locally and send it back to the server. The server then averages all incoming updates to reform new model parameters. This procedure is repeated until the model converges. These steps are summarized in Algorithm 1.

Algorithm 1: : FEEL Priority Selection Algorithm

1 Initialize $\boldsymbol{\theta}^0$, as random vector with size ξ, and determine $\mathcal{E}$ and b;
2 **for** *round* $r = 1$ *To* R **do**
3 **Input:** Set of available clients K, T, $|\mathcal{D}_k|$, f_k, ϕ, and p_k, $\forall k$;
4 **Output:** Optimal Participants that meet the constraints, and accelerate the convergence rate ;
5 Set $S_r = \{\}$;
6 Server **estimates** $T_{k,r}^{\text{cmp}}$, $T_{k,r}^{\text{up}}$ using (9) and (11) ;
7 Server **solves P3** to obtain the priority of each k ;
8 Server **sorts** the clients based on relaxed $S_r^{(k)}$ in descending order ;
9 Server **selects** $L = (S_r)_{min}$ and update $S_r = S_r \cup \{k : k = 1, 2, \dots L\}$;
10 **if** $\sum_{k=1}^{|S_r|} T_{k,r}^{\text{up}} + T_{ul}^{L+1} \geq \tau$ **then**
11 Set S_r as optimal participants;
12 **else**
13 **while** $T_{k,r}^{\text{cmp}} + T_{k,r}^{\text{up}} \leq T$ *and* $\sum_{k=1}^{|S_r|} T_{k,r}^{\text{up}} + T_{ul}^{L+1} \leq \tau$ **do**
14 Add {L+1} to S_r i.e. $S_r = S_r \cup \{k : k = L+1\}$;
15 L=L+1;
16 Server **broadcasts** $\boldsymbol{\theta}^{r-1}$ and $\nabla F^{r-1}(\boldsymbol{\theta})$ to all selected participants;
17 Each participant k in S_r **receives** $\boldsymbol{\theta}^{r-1}$ and $\nabla F^{r-1}(\boldsymbol{\theta})$ from the server then **trains** its local FEEL model locally to obtain $\boldsymbol{\theta}_k^r$ and $F_k^r(\boldsymbol{\theta})$;
18 Each participant k **sends** its local updates to the server;
19 The server **aggregates** and **averages** all updates and form a new global model

A. The properties of proposed solutions

In this section, we outline the complexity analysis, the optimality gap, and the quality of the proposed solutions. From the aforementioned discussion, one can observe that $T_{k,r}^{\mathrm{up}}$ and B limit the number of participants involved at each round. This observation can be further discussed as follows:

Lemma 2. *Let $(S_r)_{\min}$ denote the minimum number of participants and $(S_r)_{\max}$ denote the maximum number of participants. Then, the number of participants that can join a FEEL training model is as follows:*

$$(S_r)_{\min} \leq |S_r| \leq (S_r)_{\max}, \tag{37}$$

where

$$(S_r)_{\max} = \left\lfloor \frac{B}{\frac{\xi}{B \log_2 \ (1+\frac{p_{max}||h^*||^2}{N_0})}} \right\rfloor, \tag{38}$$

and

$$(S_r)_{\min} = \left\lfloor \frac{B}{\frac{\xi}{B \log_2 \ (1+\frac{p_{\min}||h^{'}||^2}{N_0})}} \right\rfloor, \tag{39}$$

where $\lfloor . \rfloor$ denote the floor function.

Proof. Based on (11), $T_{k,r}^{\mathrm{up}}$ is a decreasing function of p_k and h_k. Hence, the minimum uploading time is attained with maximum p_k and h_k. Now, let $P_k = p_{max}$, maximum transmit power, and $h_k = h^*$, ideal channel state, the maximum number of participants $(S_r)_{\max}$ can be defined as in (38). In contrast, let $P_k = P_{min}$, minimum transmit power, and $h_k = h^{'}$, the worst channel state, the minimum number of participants $(S_r)_{\min}$ can be defined as in (39). We define $(S_r)_{\max}$ as the best-case selection scenario where the edge server can select more participants allocated minimum bandwidth. We also define $(S_r)_{\min}$ as the worst-case selection scenario. Whereas, by contrast, the server can only select fewer participants allocated maximum bandwidth. □

More specifically, the server aggregate all prior information from all available clients in the network, then the server solves **P3** using Gurobi solver to obtain the optimal participants that increase the bandwidth utilization and accelerate the convergence rate, which has a time complexity of $\mathcal{O}(K^2)$. After solving **P3**, the major complexity lies in the recursive testing of the computational and communication constraints. Let L denote the initial number of selected participants. In the beginning, we can find the initial L by selecting the participants that have the highest amount of data as in (38) as worst-case scenario. Thus, Algorithm 1 reduces the complexity of finding the good candidates by utilizing a pre-estimate of the minimum participants $(S_r)_{\min}$ that can join the FEEL round. This can be pre-determined using (38), which has a time complexity of $\mathcal{O}(1)$ because the minimum number of participants determined by (38) can be accepted without checking the constraints. Then, Algorithm 1 checks the possibility of adding more participants, which takes $L - (S_r)_{\min}$ where $(S_r)_{\min}$ is calculated in $\mathcal{O}(1)$ time complexity. For the worst-case scenario, Algorithm 1 has a worst-case optimality gap of $\mathcal{O}(L/K)$ as the first L participants directly. Moreover, it is worth noting that this optimality gap is achieved in polynomial time.

B. Data diversity and dynamic deadline algorithm

As discussed in Section VI, **P3** aims to maximize the weighted number of participants; however, participants having a large number of data samples can be iteratively selected over training rounds. This will not precisely characterize the data heterogeneity and data diversity across the network especially with a high degree of heterogeneity and high non-i.i.d. level, as the neural network's loss functions are nonconvex and do not fully satisfy all theorems mentioned above in section V (i.e., the convexity assumption).

As shown in [31], FEEL system with non-i.i.d data and several tens of clients require more rounds to attain the same accuracy as that for i.i.d data. However, increasing the number of global training rounds leads to slower execution and higher operational costs. Thus, apart from existing works [17], [28], [32], [36], [38], we propose an extension for **P3** and Algorithm 1 to ensure the diversity of the data and allow more participants with heterogeneous and more diverse data samples to join the training rounds, especially the participants with small data samples. The proposed algorithm adopts a dynamic deadline where different T is occupied every $\frac{R}{t}$ rounds where $t \in [1, R]$ is an integer value to determine how many times the deadline is changed. We define a deadline vector ζ to hold the round index through which the deadline is changed. When T is large enough, the participants having much larger data samples are selected. Consequently, the models' accuracy is enhanced due to pushing much more data samples into training rounds. Thus, the server will indisputably alternate between different participants during global training rounds and capture more features to improve the model's performance. The steps of this algorithm are illustrated in Algorithm 2. Step 1 initiates a vector ζ that includes all rounds indices in which the deadline has to be changed. Steps 2-6 iteratively change the deadline when $R = \zeta[i]$ where $i = 1, 2, \ldots, t$ to $T = T * C$ where C is a uniformly distributed between 0.5 and 1.5 to specify the deadline expanding or dropping percentage for the current deadline (i.e. $C = 1.1$ then T increases by 10%, $C = 0.9$ then T decreases by 10%). The average deadline during all rounds is T to make the total time of all rounds similar. At step-7, Algorithm 1 is carried out to perform model training.

Algorithm 2: : Deadline Controlling Algorithm

1 **Determine** the round's index through which the deadline is changed i.e. $\zeta = [\frac{R}{t}.n : \forall\ n = 1, 2, \ldots t]$;
2 **Set** $i = 1$;
3 **for** $r = 1$ *To* R **do**
4 **if** $r == \zeta[i]$ **then**
5 Set $T = T.C$ where $C \sim U(0.5, 1.5)$;
6 $i = i + 1$;
7 The server **invokes** Algorithm 1 to train the global model;

VII. PERFORMANCE EVALUATION

In this section, we develop an experimental setup to demonstrate the effectiveness of our proposed approaches to improve FEEL system performance.

A. Experimental Setup

*1) **Wireless Networks and local computations***

We consider TDMA with a total bandwidth of $B = 1$ MHz. Unless specified otherwise, we model a random wireless channel gain h_k for each participant with a path loss ($\mu = g_0(\frac{d_0}{d})^4$) where $g_0 = -35$ dB and the reference distance $d_0 = 2$ m. We assume that the channel state is changing every r-*th* round. The distances between edge devices and the edge server are distributed uniformly between 5 and 100 m. Also, Additive White Gaussian Noise (AWGN) power is set to $N_0 = 10^{-6}$ watt. The transmit power p_k^r is randomly distributed between $p_{min} = -10$ dBm and $p_{max} = 20$ dBm. For the local computation, each participant's processing speed is randomly generated between minimum CPU frequency, 1 GHz, and maximum CPU frequency, 9 GHz. It is worth noting that, the CPU frequency is changing every r-*th* round.

*2) **Simulation Environment and Datasets***

To simulate our scenarios, we leverage the Tensorflow framework [39]. We use an MNIST dataset for hand-written digit classification with 10 classes (0-9), 69000 samples, and 1000 workers using multi-class logistic regression. Also, we use FEMNIST with 62 classes (digits from 1-9, A-Z, and a-z characters), $80,5263$ samples, and $3,550$ workers. In FEMNIST, each edge device represents a writer of the digits/characters with multi-class logistic regression. To verify the results, we extend our experiments by adding CIFAR10, which consists of 60000, 32x32 colored images with 10 classes, and it has 50000 for training and 10000 for testing. All datasets are used under the FEEL system setting, and Non-i.i.d distribution where the MNIST and CIFAR10 datasets are first to split into 10 partitions each (one partition for each label), and each user is assigned batches of two classes only. We use the same distribution for FEMNIST where the datasets are first split into 62 partitions (one partition for each label), and then each user is assigned batches of two labels only. We use multilayer perceptron (MLP) for MNIST and the convolutional neural network (CNN) for FEMNIST and CIFAR10 to train the models. Furthermore, the learning rate, batch size, and the number of epochs are homogeneous. We split the data on each device into a training set (80%) and a testing set (20%) at each round; a unique seed is set to enable reproducible experiments. We adopt mini-batch SGD as a local solver with $\eta = 0.01$ for MNIST and FEMNIST and 0.001 for CIFAR10 and $b = 20$ for all experiments. The performance is evaluated every round. More details, including the simulation parameters, the models, and the datasets, are listed in appendix D.

*3) **Benchmarks:***

The proposed approaches have been compared with the following baselines:

- **Random selection [16], [31]**: The participants are selected randomly in every round, and the bandwidth allocation is assigned to each k based on its transmit power and channel gain.

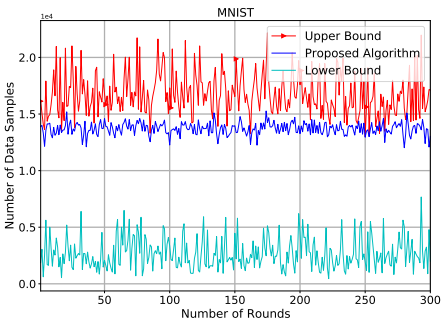


Fig. 2: The Performance of priority selection algorithm in term of data utilization vs upper and lower bounds ($R = 300$, MNIST).

- **Greedy Selection approaches [15], [18]**: This approach mainly selects the participants that provide minimum updating and uploading time regardless of their impacts on the global model or the number of local samples.
- **Fixed deadline and dynamic deadline**: Even though this approach shows outstanding performance with a fixed deadline, it lacks the flexibility of selecting different participants over time, especially with the ideal channel state. To address this challenge, we benchmark the extended proposed approach to the main proposed algorithm, Algorithm 2, considering dynamic deadlines as in Algorithm 2 during the global training rounds.

B. NUMERICAL RESULTS

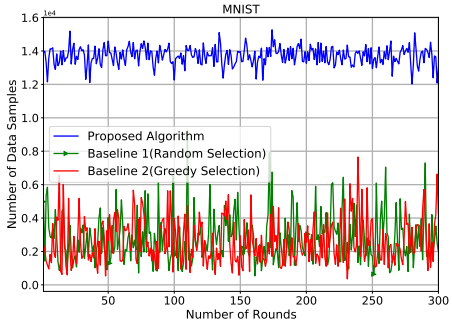
This section presents the numerical results that we carry out to evaluate the proposed algorithms' performance in terms of data utilization, testing loss, and testing accuracy. We present a thorough evaluation under extensive simulations using Proximal gradient descent (FedProx) algorithm [40]. FedProx is an enhanced algorithm of FedAvg [31] that controls the local updates to be closer to the global model received from the edge server. We employ FedProx as a local solver to train all local models

*1) **Performance gain in terms of data utilization***

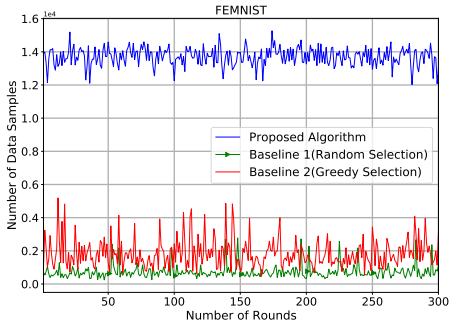
First, we present the impact of the proposed algorithms on the performance gain of data utilization by computing the number of data samples injected into model training at each round. In Fig. 2, we show the performance of the proposed priority selection algorithm compared to Gurobi optimizer solution. The upper bound illustrated in this figure is obtained from the solution of **(P2)** before relaxation. The lower bound also is obtained from the heuristic greedy solution where all participants are sorted w.r.t data size and selects one by one considering the bandwidth and deadline constraints. We can notice that Algorithm 1 approximates the upper bound over global training rounds because the fractional values resulting from the solution of **P3** gives more priority to participants having more local data samples, better computation and communication capabilities, and complying with resource and deadline constraints. It is worth mentioning that all results are averaged over five trials.

Figs. 3a and 3b shows the attained data utilization when the number of rounds is 300 using MNIST and FEMNIST

datasets. Comparing the proposed algorithm with benchmarks, it is clear that the proposed algorithm significantly improves data utilization. These gains stem from the fact that the proposed algorithm provides joint data and resource optimization; hence, including more participants with better channel states, high-performance CPUs, and more data samples. Overall, the proposed selection algorithms can significantly increase the convergence rate. This improvement stems from the fact that the training algorithms use minibatch-SGD; thus, establishing a symbiotic relationship between the total number of data samples across participants and the number of local model updates.



(a) (MNIST)



(b) (FEMNIST)

Fig. 3: Instantaneous data utilization vs the global training rounds ($R = 300$, MNIST and FEMNIST).

*2) **Performance gain in terms of testing loss and accuracy***

The proposed algorithms reduce the required number of communication rounds as the total number of local data samples increasesdue to the increase in the number of local updates. Therefore, by reasonably increasing the number of local iterations, we can save the overall communication costs by reducing the number of total communication rounds required while, at the same time, improving the quality of the global model. Since the proposed scheme can increase data utilization, computational capabilities are also utilized.

In particular, our experiments illustrate that when the number of participants and local updates are selected appropriately, a high convergence rate can be accomplished in fewer communication rounds. Fig. 4 shows the identification accuracy of handwritten digits (MNIST) when the number of global rounds is 300. From these figures, it is evident that the proposed algorithm provides a faster convergence rate compared to the benchmarks as the performance gets saturated in less than $\frac{R}{2}$

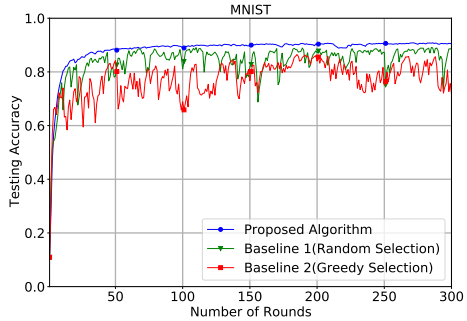


Fig. 4: Instantaneous testing accuracy vs global training rounds ($R = 300$, MNIST).

rounds while achieving better identification accuracy. This gain stems from using more data and, as a sequence performing more local updates during the learning process.

We repeat similar experiments by using FEMNIST. Fig. 5 demonstrates that our proposed selection mechanism outperforms benchmark approaches. We can see that random selection provides the worst performance. This is due to the randomized nature of such an approach, as some participants may have a bad channel state or minimum transmission power leading to an increase in the upload time, which consumes more bandwidth. However, it can be seen that the convergence rate on the MNIST dataset is much faster than the convergence rate on the FEMNIST dataset. This is because we utilize MLP for MNIST while utilizing CNN for FEMNIST and CIFAR10.

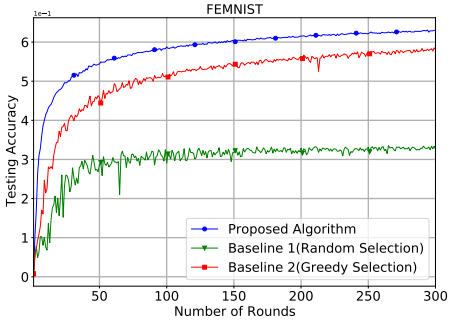


Fig. 5: Instantaneous testing accuracy vs. global training rounds ($R = 300$, FEMNIST) .

Fig. 6 presents the impact of using dynamic deadlines algorithm on convergence rate through 300 global rounds. As can be seen, there is a little acceleration in the convergence rate. This stems from the nature of MNIST datasets which have fewer classes than FEMNIST, and as a consequence, it has less diverse data amongst participants. Further, Fig. 7 shows the impacts of using dynamic deadlines on the testing accuracy during global training rounds using FEMNIST. Note that the dynamic deadline algorithm accelerates the convergence rate despite the value of t as larger and more diverse data samples are used to train the local models. However, setting a larger integer value for t improves the model performance faster. Further, we notice that the impact of the dynamic deadlines algorithm on the FEMNIST dataset model is obviously better than the model of MNIST. The reason is that FEMNIST has

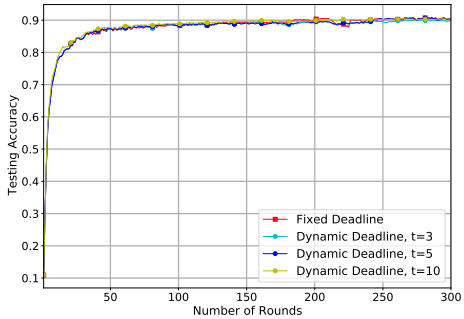


Fig. 6: Instantaneous testing accuracy of dynamic and fixed deadlines vs the global training rounds ($R = 300$, MNIST).

a high level of non-i.i.d, more diverse data samples, and a much larger number of classes (i.e., 62-class v.s. 10-class for MNIST) which means that changing the deadline can diversify the updates and enable the FEEL system to converge faster, showing that a fixed deadline cannot account for all local data distributions with the high level of non-i.i.d. among participants. More results that show the testing loss and other experiment scenarios are listed in appendix E.

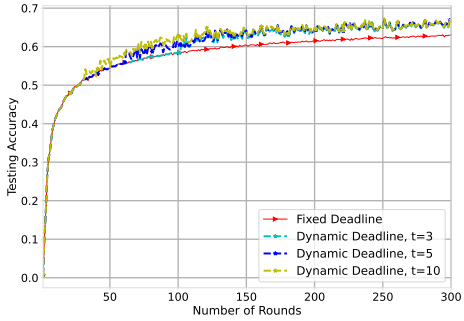


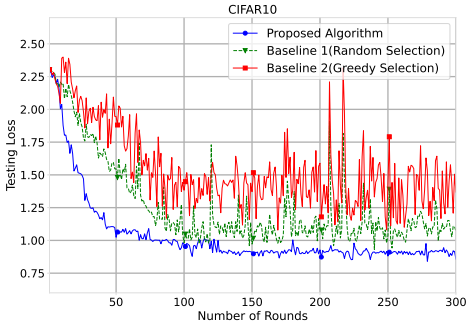
Fig. 7: Instantaneous testing accuracy of dynamic and fixed deadlines vs the training rounds ($R = 300$, FEMNIST).

To further verify our proposed approaches, we have extended our experiment by adding CIFAR10 with 1000 users as a more complex training task as seen in Figs. 8a and 8b. All results are averaged over five trails. We also repeat the experiments with CIFAR10 to validate the performance of the dynamic deadline algorithm, as illustrated in Fig. 9. In summary, one can see that the proposed approaches improves the performance significantly even when the learning task, i.e., CIFAR10, is complex, as in algorithm 1, where the optimal participants are selected on a regular basis. Furthermore, because of the adaptive deadline selection in Algorithm 2, we ensure that the training is performed by a large number of participants.

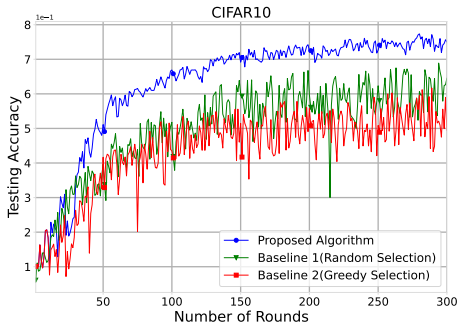
C. Insights and lessons learned

We can remark the following insights and lessons learned:

- Participants selection is challenging and plays a significant role in reducing the communication costs and enhancing the global model's performance, especially with large-scale edge networks. Selecting the participants having more data accelerates the convergence rate and strongly influences the global model.



(a) $R = 500$



(b) $R = 500$

Fig. 8: Instantaneous results of testing loss and accuracy vs global training rounds ($R = 300$, CIFAR10).

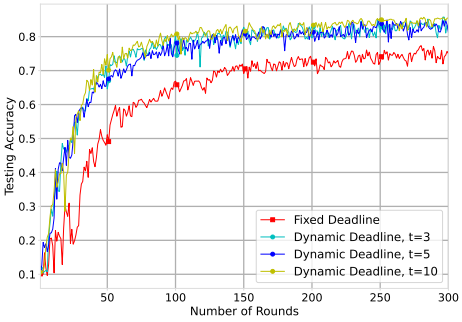


Fig. 9: Instantaneous accuracy of dynamic and fixed deadlines vs the global training rounds (CIFAR10).

- The proposed algorithms can significantly accelerate the convergence rate and increase the utilization of data by order of magnitude within the same deadline as the baseline algorithms.
- The model size significantly influences the convergence rate, limiting the number of participants involved in global training rounds. It is easy to see that the convergence time needed for MNIST is much lower than the convergence time required for FEMNIST and CIFAR10 where the latter's model size is much larger especially the CIFAR10's model, leading to involve fewer participants.
- The dynamic deadlines algorithm provides a controlling method to alternate between different participants and tackle the problem of data heterogeneity and ensure the data diversity during the training rounds

VIII. CONCLUSION

This paper proposed novel selection and allocation algorithms that tackle imbalanced data distribution, data diversity, and the limited resources as well as the deadline constraint at the network edge. Specifically, we formulated a joint communication and computation resource allocation problem aiming to enhance the data utilization as well as accelerate the convergence rate. A relaxation method was utilized to tackle the combinatorial nature of participants selection to make the complicating constraints less restrictive. Then, we developed a priority selection algorithm to select the optimal participants with low time complexity utilizing the relaxed-based solution. Finally, we extended our approach to utilize dynamic deadlines so as to address the data heterogeneity and diversify more features during the global training rounds. Extensive systematic experimentation has been carried out, and the results demonstrated that the proposed algorithms provide much better performance than state-of-the-art baselines. The dynamic deadline algorithm improves the intelligent system model's performance and ensures data diversity across the network. For future work, investigating this problem considering the energy consumption and using empirical experiments can be an interesting direction.

REFERENCES

[1] W. G. Hatcher and W. Yu, "A survey of deep learning: Platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24 411–24 432, 2018.

[2] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Towards an intelligent edge: Wireless communication meets machine learning," *arXiv preprint arXiv:1809.00343*, 2018.

[3] A. Alwarafy, A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha, "Ai-based radio resource allocation in support of the massive heterogeneity of 6g networks," in *2021 IEEE 4th 5G World Forum (5GWF)*, DOI 10.1109/5GWF52925.2021.00088, pp. 464–469, 2021.

[4] K. Huang, Q. Zhang, C. Zhou, N. Xiong, and Y. Qin, "An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2704–2713, 2017.

[5] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 1, pp. 11–20, 2017.

[6] A. Mammeri, D. Zhou, and A. Boukerche, "Animal-vehicle collision mitigation system for automated vehicles," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1287–1299, 2016.

[7] C. Gautam, A. Tiwari, S. Suresh, and K. Ahuja, "Adaptive online learning with regularized kernel for one-class classification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[8] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.

[9] A. Alwarafy, M. Abdallah, B. S. Ciftler, A. Al-Fuqaha, and M. Hamdi, "The frontiers of deep reinforcement learning for resource management in future wireless hetnets: Techniques, challenges, and research directions," *IEEE Open Journal of the Communications Society*, 2022.

[10] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[11] A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha, "Exploiting unlabeled data in smart cities using federated edge learning," pp. 1666–1671, 2020.

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

[13] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[14] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[15] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE, 2019.

[16] 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, 2019.

[17] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6. IEEE, 2020.

[18] 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, 2020.

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

[20] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Transactions on Wireless Communications*, 2020.

[21] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[22] 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*, Jul. 2021.

[23] G. Zhu, Y. Du, D. Gunduz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *arXiv preprint arXiv:2001.05713*, 2020.

[24] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: Compressed optimisation for non-convex problems," *arXiv preprint arXiv:1802.04434*, 2018.

[25] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, 2019.

[26] M. M. Amiri and D. Gunduz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *[Online]. Available: https://arxiv.org/abs/1901.00844*, 2019.

[27] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

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

[29] T. T. Anh, N. C. Luong, D. Niyato, D. I. Kim, and L.-C. Wang, "Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1345–1348, 2019.

[30] 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.

[31] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[32] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6. IEEE, 2020.

[33] A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Erbad, "Threshold-based data exclusion approach for energy-efficient federated edge learning," pp. 1–6, 2021.

[34] 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 (GLOBECOM)*, pp. 1–6, 2021.

[35] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, 2020.

[36] C. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *arXiv preprint arXiv:1910.13067*, 2019.

[37] G. Optimization, "Gurobi optimization-the state-of-the-art mathematical programming solver," 2018.

[38] Y. Liu, W. Xu, G. Wu, Z. Tian, and Q. Ling, "Communication-censored admm for decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2565–2579, 2019.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[40] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. S. Talwalkar, and V. Smith, "Federated optimization for heterogeneous networks," 2018.

Abdullatif Albaseer is a Ph.D. student at the Smart Communication Networks & System lab, Hamad Bin Khalifa University, Doha, Qatar. He received a B.Sc in computer science from Thamar University, Thamar, Yemen, in 2009, and received an M.Sc. degree in Computer Networks from King Fahd University of PetroleumMinerals (KFUPM), Dhahran, Saudi Arabia, in 2017. His current research interests include wireless communication, Federated Learning, Deep learning, IoT, and Smart cities.

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.

A IMAN ERBAD (Senior Member, IEEE) received the M.C.S. degree in embedded systems and robotics from the University of Essex, U.K., and the Ph.D. degree in computer science from The University of British Columbia, Canada. He is currently an Associate Professor with the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). His research interests include cloud computing, edge computing, the IoT, private and secure networks, and multimedia systems. He received the Platinum award from H. H. 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. He is an Editor of KSII *Transactions on Internet and Information Systems* and was a Guest Editor of *IEEE Network*

APPENDIX A
PROOF OF THEOREM 1

Given Assumption 1, and the definition of $F(\boldsymbol{\theta})$ in section III-A, we have:

$$F(\boldsymbol{\theta}^r) - F(\boldsymbol{\theta}^*) \leq \frac{\beta}{2} \left\| \boldsymbol{\theta}^r - \boldsymbol{\theta}^* \right\|^2. \tag{40}$$

From (6), by the substitution of

$$\boldsymbol{\theta}^r = \frac{\sum_{k=1}^{|S_r|} |\mathcal{D}_k| \boldsymbol{\theta}_k^r}{\sum_{k \in S_r} |\mathcal{D}_k|}$$

into the right-hand side of (40), we have the following:

$$\begin{aligned}
\frac{\beta}{2} \left\| \boldsymbol{\theta}^r - \boldsymbol{\theta}^* \right\|^2 &= \frac{\beta}{2} \left\| \frac{\sum_{k=1}^{|S_r|} |\mathcal{D}_k| \boldsymbol{\theta}_k^r}{\sum_{k=1}^{|S_r|} |\mathcal{D}_k|} - \boldsymbol{\theta}^* \right\|^2 \\
&= \frac{\beta}{2} \left\| \frac{\sum_{k=1}^{|S_r|} |\mathcal{D}_k| (\boldsymbol{\theta}_k^r - \boldsymbol{\theta}^*)}{\sum_{k=1}^{|S_r|} |\mathcal{D}_k|} \right\|^2 \\
&= \frac{\beta}{2} \left\| \frac{\sum_{k=1}^{K} S_r^{(k)} |\mathcal{D}_k| (\boldsymbol{\theta}_k^r - \boldsymbol{\theta}^*)}{\sum_{k=1}^{K} S_r^{(k)} |\mathcal{D}_k|} \right\|^2 \\
&= \frac{\beta}{2} \frac{\left\| \sum_{k=1}^{K} S_r^{(k)} |\mathcal{D}_k| (\boldsymbol{\theta}_k^r - \boldsymbol{\theta}^*) \right\|^2}{\left(\sum_{k=1}^{K} S_r^{(k)} |\mathcal{D}_k| \right)^2}
\end{aligned} \tag{41}$$

where $S_r^{(k)}$ is the indicator defined in (13).

APPENDIX B
PROOF OF THEOREM 2

Since $F(\cdot)$ is also L-Lipschitz (i.e., $\|\nabla F(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta}^*)\| \leq \sum_{k=1}^{K} \delta_k \|\nabla F_k^r(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta}^*)\| \leq L \|\boldsymbol{\theta}^r - \boldsymbol{\theta}^*\|, \forall \boldsymbol{\theta}, \boldsymbol{\theta}^*$, by using Jensen's inequality and L-smoothness, respectively), we have

$$\begin{aligned}
&F_k^r(\boldsymbol{\theta}_k^r) - F(\boldsymbol{\theta}^*) \\
&\leq \langle \nabla F(\boldsymbol{\theta}^*), \boldsymbol{\theta}_k^r - \boldsymbol{\theta}^* \rangle + \frac{L}{2} \left\| \boldsymbol{\theta}_k^r - \boldsymbol{\theta}^* \right\|^2 \\
&= \langle \nabla F(\boldsymbol{\theta}^*) - \nabla \bar{F}^{r-1}, \boldsymbol{\theta}_k^r - \boldsymbol{\theta}^* \rangle + \frac{L}{2} \left\| \boldsymbol{\theta}_k^r - \boldsymbol{\theta}^* \right\|^2 \\
&\quad + \langle \nabla \bar{F}^{r-1}, \boldsymbol{\theta}_k^r - \boldsymbol{\theta}^* \rangle \qquad (42) \\
&\leq \left\| \nabla F(\boldsymbol{\theta}^*) - \nabla \bar{F}^{r-1} \right\| \left\| \boldsymbol{\theta}_k^r - \boldsymbol{\theta}^* \right\| + \frac{L}{2} \left\| \boldsymbol{\theta}_k^r - \boldsymbol{\theta}^* \right\|^2 \\
&\quad + \langle \nabla \bar{F}^{r-1}, \boldsymbol{\theta}_k^r - \boldsymbol{\theta}^* \rangle \qquad (43)
\end{aligned}$$

Then, we use the fact that $F(.)$ is a decreasing function due to the convexity and smoothness; thus, for each individual participant, we can have:

$$\begin{aligned}
F_k(\boldsymbol{\theta}_k^r) - \nabla F_k(\boldsymbol{\theta}^*) &\leq \frac{1}{\mathbf{n}} \sum_{i=1}^{\mathbf{n}} \nabla F_k^r(\boldsymbol{\theta}_k^r(i)) - \nabla F_k(\boldsymbol{\theta}^*) \\
&\leq \frac{\|\boldsymbol{\theta}_k^r(0) - \boldsymbol{\theta}^*\|_2^2}{2\eta \mathbf{n}}
\end{aligned} \tag{44}$$

where

$$\mathbf{n} = \mathcal{E} \ \frac{|\mathcal{D}_k|}{b}$$

is the number of local updates performed by participant k. Further, from (4), we have:

$$\boldsymbol{\theta}_k^r = \boldsymbol{\theta}^{r-1} - \eta \sum_{i=1}^{\mathbf{n}} \nabla F_k^r(\boldsymbol{\theta}_k^r(i)), \tag{45}$$

Thus, $F_k^r(\boldsymbol{\theta}_k^r(0))$, $F_k^r(\boldsymbol{\theta}_k^r(1))$, ..., $F_k^r(\boldsymbol{\theta}_k^r(\mathbf{n}))$ decreases with $\mathbf{n}$ and $\boldsymbol{\theta}_k^r(.)$ due to its smoothness and convexity and its value decreases as the the number of data samples increases (the number of updates is proportional to the number of batches which depends on the data size assuming that the batch size is homogeneous among all participants). Therefore,

$$\begin{aligned}
&F_k^r(\boldsymbol{\theta}_k^r(\mathbf{n})) - F(\boldsymbol{\theta}^*) \leq F_k^r(\boldsymbol{\theta}_k^r(\mathbf{n}-1)) - F(\boldsymbol{\theta}^*) \ldots \\
&\leq F_k^r(\boldsymbol{\theta}_k^r(\mathbf{n}-2)) - F(\boldsymbol{\theta}^*) \cdots \leq F_k^r(\boldsymbol{\theta}_k^r(0)) - F(\boldsymbol{\theta}^*)
\end{aligned}$$

As a result from (42), (43), (44) and (45), we notice that as the number of local data points increases, the divergence between $\boldsymbol{\theta}_k^r$ and $\boldsymbol{\theta}^*$ decreases

APPENDIX C
PROOF OF THEOREM 1

In the following, show the relationship between the convergence rate of global model and the number of selected participants weighted to the number of local data samples. Let $S_1(r) = \{k \in \{1, \ldots, K\} \mid S_r^{(k)} = 1\}$ is the selected participant at round r and $S_2(r) = \{k \in \{1, \ldots, K\} \mid S_r^{(k)} = 0\}$ is the unselected participant at round r. we first rewrite $F\left(\boldsymbol{\theta}^{r+1}\right)$ using the second-order Taylor expansion, which can be expressed as

$$\begin{aligned} F\left(\boldsymbol{\theta}^{r+1}\right) =& F\left(\boldsymbol{\theta}^{r}\right) + \left(\boldsymbol{\theta}^{r+1} - \boldsymbol{\theta}^{r}\right)^T \nabla F\left(\boldsymbol{\theta}^{r}\right) + \frac{1}{2}\left(\boldsymbol{\theta}^{r+1} - \boldsymbol{\theta}^{r}\right)^T \\ & \nabla^2 F\left(\boldsymbol{g}\right)\left(\boldsymbol{\theta}^{r+1} - \boldsymbol{\theta}^{r}\right), \qquad (46) \\ \leq & F\left(\boldsymbol{\theta}^{r}\right) + \left(\boldsymbol{\theta}^{r+1} - \boldsymbol{\theta}^{r}\right)^T \nabla F\left(\boldsymbol{\theta}^{r}\right) + \frac{\mathcal{L}}{2}\|\boldsymbol{\theta}^{r+1} - \boldsymbol{\theta}^{r}\|^2, \end{aligned}$$

where the inequality in (46) is resulting from assumption (1). Let's learning rate $\eta = \frac{1}{\mathcal{L}}$, based on (20) and (21), the following expectation $\mathbb{E}\left(F\left(\boldsymbol{\theta}^{r+1}\right)\right)$ for the global loss holds as:

$$\begin{aligned} \mathbb{E}\left(F\left(\boldsymbol{\theta}^{r+1}\right)\right) \leq & \mathbb{E}\Bigg(F\left(\boldsymbol{\theta}^{r}\right) - \eta(\nabla F\left(\boldsymbol{\theta}^{r}\right) - \boldsymbol{\Psi})^T \nabla F\left(\boldsymbol{\theta}^{r}\right) + \\ & \frac{L\eta^2}{2}\|\nabla F\left(\boldsymbol{\theta}^{r}\right) - \boldsymbol{\Psi}\|^2\Bigg), \qquad (47) \\ = & \mathbb{E}\left(F\left(\boldsymbol{\theta}^{r}\right)\right) - \frac{1}{2\mathcal{L}}\|\nabla F\left(\boldsymbol{\theta}^{r}\right)\|^2 + \frac{1}{2\mathcal{L}}\mathbb{E}\left(\|\boldsymbol{\Psi}\|^2\right), \end{aligned}$$

where

$$\begin{aligned} \mathbb{E}\left(\|\boldsymbol{\Psi}\|^2\right) &= \mathbb{E}\left(\left\|\nabla F\left(\boldsymbol{\theta}^{r}\right) - \frac{\sum_{k=1}^{K}\sum_{s=1}^{|\mathcal{D}_k|} S_r^{(k)} \nabla F_s(\boldsymbol{\theta})}{\sum_{k=1}^{K} |\mathcal{D}_k| S_r^{(k)}}\right\|^2\right) \\ &= \mathbb{E}\left(\left\|\frac{\sum_{k\in S_1(r)}\sum_{s=1}^{|\mathcal{D}_k|} \nabla F_s(\boldsymbol{\theta})}{D} - \frac{(D - \sum_{k=1}^{K} |\mathcal{D}_k| S_r^{(k)}) \sum_{k\in S_2(r)}\sum_{s=1}^{|\mathcal{D}_k|} \nabla F_s(\boldsymbol{\theta})}{D \sum_{k=1}^{K} |\mathcal{D}_k| S_r^{(k)}}\right\|^2\right) \qquad (48) \\ &\leq \mathbb{E}\left(\frac{\sum_{k\in S_1(r)}\sum_{s=1}^{|\mathcal{D}_k|} \|\nabla F_s(\boldsymbol{\theta})\|}{D} - \frac{(D - \sum_{k=1}^{K} |\mathcal{D}_k| S_r^{(k)}) \sum_{k\in S_2(r)}\sum_{s=1}^{|\mathcal{D}_k|} \|\nabla F_s(\boldsymbol{\theta})\|}{D \sum_{k=1}^{K} |\mathcal{D}_k| S_r^{(k)}}\right)^2. \end{aligned}$$

Hence, based on Assumption 1 and triangle-inequality as in [21], [28] we have:

$$\mathbb{E}\left[F\left(\boldsymbol{\theta}^{r+1}\right) - F\left(\boldsymbol{\theta}^{*}\right)\right] \leq \frac{2c_1}{LD}\sum_{k=1}^{K} |\mathcal{D}_k| \left(1 - S_r^{(k)}\right) + (1 - \frac{\beta}{\mathcal{L}} + \frac{4\beta c_2}{LD}\sum_{k=1}^{K} |\mathcal{D}_k| \left(1 - S_r^{(k)}\right))\mathbb{E}\left(F\left(\boldsymbol{\theta}_k^{r}\right) - F\left(\boldsymbol{\theta}^{*}\right)\right). \qquad (49)$$

where c1, and c2 are obtained from second-order Taylor expansion. We can notice that the upper bound of the gap between the left-hand side and the right-hand side in (49) is $\frac{2c_1}{LD}\sum_{k=1}^{K} |\mathcal{D}_k| \left(1 - S_r^{(k)}\right)$. This gap can be reduced by pushing more data and selecting more participants, which leads to accelerating the convergence rate as the gap between the optimal model $\boldsymbol{\theta}^*$ and the trained model $\boldsymbol{\theta}^r$ is shrinking as we increase the number of participants weighted to local data samples as in (49). Hence Proved.

APPENDIX D
EXPERIMENTS DETAILS

This appendix contains the simulation parameters as in Table III. Also, we provide details on the used datasets and models. Both datasets are used under FEEL setting and Non-i.i.d distribution.

- **MNIST**: The image classification of handwritten digits 0-9 is studied in MNIST using multinomial logistic regression. To conduct experiments under non-i.i.d settings, the data was distributed among 1000 users in such a way that each user has imbalanced samples of just 2 digits and the number of samples per user follows a power law to ensure the imbalanced

TABLE III: EXPERIMENTAL SETUP PARAMETERS

Parameter	**Value**
Library	TensorFlow
Simulation Environment	Anaconda
learning rate η	0.01 and 0.001
Batch Size	20
Number of rounds	300 and 800
Local solver	Minibatch-SGD
Evaluation Period	Per round

data distribution. The model input is a flattened 784-dimensional (28 × 28) image, and the output is a class label between 0 and 9.

- **FEMNIST**: An image classification problem is studied in more realistic datasets on the 62-class FEMNIST dataset using multinomial logistic regression. Each user represents a writer of the digits/characters in Extended MNIST, **EMNIST**. This version of the datasets is called Federated FEMNIST. The model input is a flattened 784-dimensional (28 × 28) image, and the output is a class label between 0 and 61. To conduct experiments under non-i.i.d settings, each user was assigned only 2 labels and the number of samples per user is randomly assigned to ensure the imbalanced data samples among clients.
- **CIFAR10**: An consists of 60000, 32x32 colored images with 10 classes, and it has 50000 for training and 10000 for testing. The images are divided into ten categories: airplane, automobile (but not truck or pickup truck), bird, cat, deer, dog, frog, horse, ship, and truck (but not pickup truck). Each class contains 6000 photographs, comprising 5000 training and 1000 assessment images. The data was distributed among 1000 users in such a way that each user has imbalanced samples of just 2 digits and the number of samples per user follows a power law to ensure the non-i.i.d and imbalanced data distribution.

Table IV illustrates the statistics of the utilized datasets under FEEL settings.

TABLE IV: STATISTICS OF THE UTILIZED DATASETS

Dataset	Total number of devices	Total number of data samples	Number of classes
MNIST	1000	69000	10
FEMNIST	3,550	80,5263	62
CIFAR10	1000	60000	10