ABSTRACT  A variety of modern AI products essentially require raw user data for training diverse machine learning models. With the increasing concern on data privacy, federated learning, a decentralized learning framework, enables privacy-preserving training of models by iteratively aggregating model updates from participants, instead of aggregating raw data. Since all the participants, i.e., mobile devices, need to transfer their local model updates concurrently and iteratively over mobile edge networks, the network is easily overloaded, leading to a high risk of transmission failures. Although previous works on transmission protocols have already tried their best to avoid transmission collisions, the number of iterative concurrent transmissions should be fundamentally decreased. Inspired by the fact that raw data are often generated unevenly among devices, those devices with a small proportion of data could be properly excluded since they have little effect on the convergence of models. To further guarantee the accuracy of models, we propose to properly select a subset of devices as participants to ensure the given proportion of involved data. Correspondingly, we propose to minimize the risk against the transmission failures during model updates. Afterwards, we design a randomized algorithm (ranRFL) to choose suitable participants by using a series of delicately calculated probabilities, and prove that the result is concentrated on its optimum with high probability. Extensive simulations show that through delicate participant selection, ranRFL decreases the maximal error rate of model updates by up to 38.3% compared with the state-of-the-art schemas.

INDEX TERMS Concurrent transmissions, federated learning, mobile edge networks.

I. INTRODUCTION
Various mobile devices [1], like smart phones and portable tablets, continuously produce diverse user data during users’ daily usages of applications [2], including the ClickLogs [3], the user trajectories from GPS and so on. In order to provide better services to users, raw user data are widely collected and used to train related machine learning models [4] for a variety of modern AI products [5], guiding some commercial decisions [6], such as commodity recommendation for users, route guidance [7], etc. However, with the growing concern on data privacy [8], users are unwilling to upload their raw data for training models centrally. As a result, equipped with increasingly advanced computing capabilities [9], mobile devices are encouraged to train their models locally by using their own raw data, and then, they only upload related model updates iteratively to the central place, like a nearby edge server [10], for global model aggregation, i.e., the federated learning framework [11].

Unfortunately, iterative model updates with too many concurrent transmissions may easily overload shared mobile edge networks and incur transmission failures. In order to verify this, firstly, we conduct preliminary studies on the training process of federated learning. More technically, we implement the training of KMeans by using Hadoop, which is a typical clustering method in machine learning upon a typical distributed data analytics framework. As illustrated in Fig. 1, both iterative training setups and iterative model updates lead to a large number of concurrent transmissions within a short time duration. Secondly, as shown in Fig. 1
further, with the growth of concurrent transmissions, local area network (LAN, shared networks, etc.) is more likely to have a higher risk of transmission failures [12]. Due to the collisions over shared links or wireless channels [13], concurrent transmissions easily result in transmission failures during the establishment of connections. In brief, iterative model updates easily incur iterative transmission failures over shared mobile edge networks, which should be avoided as much as possible in federated learning.

Although some works have already studied collision/error based re-transmission mechanisms [14]–[16], as well as utilization based transmission protocols [13], [17]–[20] to make best efforts to transfer data upon shared links or channels, either shared links or network devices are easily overloaded, also resulting in a high risk of failures [21] during model updates. To fundamentally avoid too many transmissions, inspired by the fact that raw user data are often generated unevenly among mobile devices [2], [10], [11], in this paper, we propose to select suitable participants for federated learning to ensure both convergence and accuracy of machine learning models. Firstly, since reasonably reducing the number of participants has little effect on the convergence of models [22], i.e., the model still gets closer to its optimum, those devices with a small proportion of raw data could be properly excluded. Secondly, the accuracy of models in federated learning could also be guaranteed [23] by ensuring the given ratio of involved data to the whole dataset as well as the number of iterations. As a result, the number of concurrent model updates may decrease so as to minimize the high risk caused by concurrent transmission failures.

Consequently, under the consideration of both participant selection and the number of concurrent transmissions, we propose to minimize the maximal error rate of transmissions during model updates. Then, we design a randomized algorithm to select participants with a series of delicately calculated probabilities, and prove that the result of our proposed algorithm ranRFL is concentrated on its optimum with high probability through our rigorous theoretical analysis. Numerical experiments show that through delicate participant selection, ranRFL decreases the error rate of model updates by up to 38.3% compared with the state-of-the-art schemas.

- To decrease the high risk of transmission failures over shared mobile networks for model updates, we propose to select suitable devices as participants in federated learning, considering the number of concurrent transmissions as well as the proportion of involved data.
- We design a randomized algorithm (ranRFL) to choose suitable participants by using a series of delicately calculated probabilities with rigorous theoretical analysis.
- Extensive simulations show that ranRFL decreases the error rate of model updates by up to 38.3% compared with the state-of-the-art schemas.

The rest of this paper is organized as follows. Section II briefly summarizes and reviews the related works. Section III illustrates our preliminary case studies and motivation. Section IV presents the system model and problem formulation. Sections V and VI develop the design of our randomized algorithm and the theoretical analysis, respectively. Section VII evaluates the performance through extensive simulations. Finally, Section VIII concludes the paper.

II. RELATED WORKS

We summarize prior research in two categories, then highlight their drawbacks compared to our work, respectively.

A. MODEL UPDATES IN FEDERATED LEARNING

Yang et al. [24] analyzed the performance of federated learning when communication resources were limited and the transmissions were subjected to the interference, and found that transmission failures could be severe. Although they analyzed three existing scheduling policies, they ignored to propose a systematical schema with theoretical analysis. Keith et al. [11] introduced the federated learning framework for mobile devices, supporting the unreliable connectivity of devices as well as interrupted scenarios. However, they failed to consider the data sizes generated within the devices and related model accuracy during the participant selection. Nguyen et al. [23] studied the computation and communication characteristics for federated learning, but ignored to further explore the risk of transmission failures on communications. Rulin et al. [25] used stochastic-channel based approach to avoid inverse-model attacks in federated learning.

These works mainly ignore to model, formulate and analyze the risk of transmission failures during the model updates for federated learning in mobile edge networks, especially when the number of concurrent updates actually affects the error rate of transmissions.

B. CONCURRENT TRANSMISSIONS

Some works [15], [16] also focused on collision-aware re-transmission mechanisms upon shared links or channels. Xiaoyu et al. [14] proposed STAIRS, a time and energy efficient collision resolution mechanism for wireless sensor networks based on CSMA/CA. Others [20], [26]–[28] mainly studied on the fully use of transmission capability, making...
best efforts to transfer data. Injong et al. [21] switched protocols based on the contention status in wireless networks. Works like [13], [17]–[19] supported cross-technology data forwarding to transfer data by using hybrid protocols and devices.

Those works have tried their best to avoid collisions and errors by re-transferring data or fully utilizing the transmission capability of shared links and channels. In order to essentially decrease the number of concurrent model updates in federated learning, as well as to release the heavy burden on the mobile edge networks, we prefer to select suitable participants with guaranteed ratio of involved training data to the whole dataset, fundamentally avoiding too many transmissions concurrently during model updates.

C. PRIVACY-PRESERVING COMPUTING AT EDGES

The basic idea of federated learning framework, i.e., aggregating the updated model instead of raw user data, actually protects the data privacy. Based on the framework of federated learning, previous works have already focused on several privacy issues, proposing many privacy-based schemas like [29]–[32]. Furthermore, other works focused on the security of edge computing, designing a series of systems and algorithms to protect the vulnerable edges from attacks, like [33]–[36].

These works mainly focus on the security issues to protect both the edges and related federated learning frameworks from attacks. Although the federated learning framework is secured, the efficiency may be influenced if the error risk is high regarding the concurrent transmissions.

III. PRELIMINARY CASE STUDIES AND MOTIVATION

In this section, we illustrate our preliminary case studies, which guide the motivation for participant selection to decrease the concurrent transmissions in federated learning.

A. ITERATIVE MODEL UPDATES

We first investigate the characteristics of model updates within decentralized learning framework. To do so, we investigate the training process of KMeans, a typical clustering method in machine learning based on a typical distributed data analytics framework. More specifically, we build the testbed by using Hadoop 3.0.0 upon Inspur SN5160M4 with Intel E5-2680V4, 40 cores and 125GB memory. Note that each participant is implemented as a task in Hadoop. More specifically, for each case, KMeans uses 32 participants for iterative model training, whose iteration number is 20.

As shown in Fig. 2, iterative model updates within decentralized learning framework easily incur concurrent transmissions during a short time lag. Firstly, for each iteration, KMeans triggers model downloads before training and aggregates model updates after training, respectively, for all of the participants. Correspondingly, we measure the time durations for both startups and update aggregations. With the growth of average input size for each KMeans participant, from 2MB to 16MB in uniform distribution, the time durations for startups are nearly the same, i.e., \(~1\)s, which means concurrent model downloads are often triggered within a short time. In contrast, the duration for update aggregations increases with the growth of average input size, because the divergence of participant completions also increases. Secondly, as shown in Fig. 2 further, the lag between two consecutive startups is short, i.e., over 92% lags between two consecutive startups are often less than only 0.1s. As a result, such concurrent transmissions triggered within 1s, whose lags between consecutive startups are often less than 0.1s, easily overload shared mobile edge networks.

B. RISK OF CONCURRENT TRANSMISSION FAILURES

We then study the relationship between the risk of transmission failures and various numbers of concurrent transmissions. The trace analyzed is a nine-week raw data [12] dumped from a typical U.S. LAN, including 743MB sized files, 70 different services and 41 features. For most of these services, the increase of the connection numbers often leads to a high risk of transmission failures, as shown in Fig. 3. We further use sigmoid function and tanh function to fit such relationships. More specifically, we use sigmoid function \(\frac{1}{1+e^{-x}}\) and tanh function \((1-2\cdot52^{-x})\), respectively, which facilitate our numerical experiments as shown later.

These transmission failures are often derived from DDoS attacks [37] and the collisions of transmissions over shared links or wireless channels [38]. Although previous related works have already designed some re-transmission mechanisms based on various communication protocols, the performance of concurrent transmissions over shared mobile edge networks still affects the model updates in federated learning:

- Since large-size data are often split into multiple frames and then transferred over shared wireless networks,
model updates with large-size of parameters (high dimensions, typically hundreds of MBs [39]) easily lead to frequently frame collisions between the participants.

- Although the collisions of frames only incur nearly tens of milliseconds for re-transmission, plenty of participants with large-size models would incur costly collision overhead (exponential growth [40]).
- Moreover, with the growth of both the participant numbers and the sizes of transferred data, heavy-tailed performance in terms of the transmission latency or even zero throughput [41]–[43] is more likely to occur.

To avoid these severe situations, the number of concurrent transmissions should be fundamentally decreased as much as possible for a lower risk of transmission failures.

C. CONVERGENCE AND ACCURACY

1) CONVERGENCE

The convergence of federated learning indicates that as the iterations proceed, the output, i.e., updated model, gets closer to the target one. Such target model performs the best in terms of the accuracy on a given dataset. In order to show the relationship between the convergence and the number of participants, we illustrate the iterations needed for achieving an acceptable loss threshold [22] under various numbers of participants, as shown in Table 1. Note that the cases listed in Table 1 are trained using MNIST [44], and the whole dataset is distributed among selected participants. Other results with plenty of cases are further shown in previous works [22]. Although the number of participants changes from 10 to 100, as well as the total size of the dataset increases, the number of iterations used in MNIST is stable. More specifically, federated learning needs 190 iterations for 10 participants to achieve 0.535 loss whereas it needs 188 for 100 participants. That means the number of participants has little effect on the convergence of models.

TABLE 1. Model convergence in federated learning under various number of participants [22].

| Number of Participants | Number of Iterations Used\(^1\) under Balanced Data Sizes | Number of Iterations Used\(^2\) under Skewed Data Sizes |
|------------------------|----------------------------------------------------------|----------------------------------------------------------|
| 10                     | 190                                                      | 186                                                      |
| 30                     | 193                                                      | 187                                                      |
| 50                     | 189                                                      | 187                                                      |
| 70                     | 188                                                      | 187                                                      |
| 100                    | 188                                                      | 188                                                      |

\(^1\) The loss threshold is 0.536; Overall number of iterations is 200.

\(^2\) The loss threshold is 0.323; Overall number of iterations is 200.

2) ACCURACY

Furthermore, Table 1 illustrates the numbers of iterations needed under different data sizes generated among participants. When the data sizes are distributed unbalanced among participants [45], the number of iterations needed is almost the same with that under balanced data sizes among participants. More specifically, the desired loss thresholds, i.e., 0.536 and 0.323 for two different cases, are both achieved easily within 190 iterations. That means the skewed data sizes have little effect on the accuracy. We should mention here that the overall number of iterations is 200. Furthermore, the accuracy could be theoretically guaranteed, so long as each data item of each participant is sampled uniformly from the dataset (common assumption).

3) THEORETICAL RELATIONSHIP

Although the number of iterations used under both balanced and skewed data sizes as well as the iterations used under various numbers of participants are almost the same, i.e., both of them are not far away from the overall number of iterations used, i.e., 200 in this case, the influences caused by these two factors on the iterations are quite different for achieving desired training accuracy. In order to reveal such relationship, we analyze these factors based on previous theoretical analysis [22] as follows:

$$T_\epsilon \propto \left(1 + \frac{1}{K}\right)EG^2 + \frac{\sum_k \beta_k^2 \sigma_k^2 + \Lambda \Gamma + \kappa G^2}{E}, \tag{1}$$

where \(T_\epsilon\) denotes the number of global iterations required for federated learning (by using FedAvg) to achieve a given accuracy \(\epsilon\), \(K\) denotes the number of participants, and \(E\) denotes the number of local iterations performed in a device. Except for other problem-related constants, \(\sigma_k\) and \(\Gamma\) are used to measure the impact of unbalanced raw user data among devices. More specifically, \(\sigma_k\) is the maximum change on the local loss caused by each data sample in device-\(k\) and \(\Gamma\) is used for quantifying the degree of unbalanced data among devices. If the data are independently identically distribution (IID) sampled from the underlying data distribution, then \(\Gamma\) goes to zero as the number of samples grows. If the data are non-IID, then \(\Gamma\) is nonzero, and its magnitude reflects the heterogeneity of the data distribution. Then, intuitively, if we exclude those devices with small proportions of data, the accuracy could be also achieved by using the rest participants.

D. MOTIVATION FOR PARTICIPANT SELECTION

Since: 1) iterative model updates in federated learning easily incur iterative concurrent transmissions, 2) too many concurrent transmissions easily lead to a high risk of transmission failures, and 3) suitable excluding those devices with a small proportion of data has little effect on the convergence, we propose to select suitable devices as participants to decrease the number of concurrent transmissions so as to decrease the high risk of transmission failures in federated learning.

Fig. 4 shows a simple illustrating example, Device 1 with a small proportion of data, i.e., 0.1%, could be excluded in federated learning so as to decrease the number of concurrent transmissions. However, excluding too many participants may also influence the accuracy of federated learning. For example, excluding all of these devices is unrealistic. Therefore, we propose to ensure a given ratio of involved data to the whole dataset. For example, if we set the given ratio as 70%, Devices 2 - 4 could not be excluded. Furthermore,
the availability of device-$\pi$, we denote by $\pi$ connected to the same edge (logically). Due to the mobility of generality, we assume that all of these mobile devices are so as to communicate with the outside world. Without loss of normal use.

**IV. SYSTEM MODELS AND PROBLEM FORMULATION**

The notations used in this paper are listed in Table 2.

![A diagram illustrating the device with a small proportion of data could be excluded to decrease the number of transmissions, considering both convergence and accuracy in federated learning.](image)

**TABLE 2. Summary of major notations.**

| Symbol | Description |
|--------|-------------|
| $N$    | Number of all devices |
| $M$    | Number of concurrent transmissions during model updates |
| $\pi_i$ | Variable indicates the availability of device $i$ |
| $e_i$  | Error rate of device $i$ during daily normal use |
| $d_i$  | Data size generated within device $i$ |
| $f(M)$ | Error rate of transmissions when there are $M$ participants |
| $r_i$  | Error rate of device $i$ during model updates |
| $\rho$ | Given proportion of involved data |
| Decision | Description |
| $\phi_i$ | Decision variable indicates whether device $i$ is a participant |

**A. SYSTEM MODEL**

1) **MOBILE EDGE NETWORKS**

We consider a mobile edge scenario that consists of a group of distributed mobile devices, including smart phones, portable tablets and so on, denoted as $\mathcal{N} = \{1, 2, \ldots, N\}$. Each mobile device connects to a nearby edge, e.g., cellular base station so as to communicate with the outside world. Without loss of generality, we assume that all of these mobile devices are connected to the same edge (logically). Due to the mobility of mobile devices, we denote by $\pi_i \in \{0, 1\}, \forall i \in \mathcal{N}$ the availability of device-$i$, i.e., $\pi_i = 1$ when device-$i$ is available, and $\pi_i = 0$ otherwise. Since mobile devices are also vulnerable [11] due to limited battery resources and the threats like malware or physical disassembly, we denote by $e_i \in [0, 1], \forall i \in \mathcal{N}$ the error rate of device-$i$ during its daily normal use.

The mobile edge network often supports concurrent data transmissions, which means multiple mobile devices could exchange data with the same edge at the same time. However, if there are too many concurrent transmissions, the network may incur transmission failures with high probability. As shown in Fig. 1, with the growth of concurrent transmission numbers, the network is more likely to have a higher risk of transmission failures. To describe the relationship between the error rate and the number of concurrent transmissions, we denote by $f(M)$ the error rate under a fixed participant number $M$, where $M$ indicates the number of concurrent transmissions connected to the edge during model updates. Two typical examples of $f(M)$ have been illustrated according to the U.S. LAN trace in previous section.

2) **FEDERATED LEARNING**

Federated learning enables mobile devices to train their local models by using raw user data, and then iteratively upload model updates for model aggregation. We denote by $d_i \in \mathbb{N}, \forall i \in \mathcal{N}$ the size of raw user data generated in device-$i$. In order to indicate the participation of each device in federated learning, we use $\phi_i \in \{0, 1\}, \forall i \in \mathcal{N}$ as an indicator variable for device-$i$, i.e., the decision variable. More specifically, $\phi_i$ equals 1 indicates device-$i$ is enrolled in the training of federated learning, and $\phi_i$ equals 0 otherwise. Then, the overall number of concurrent transmissions, i.e., the number of model updates concurrently within each iteration in federated learning is

$$M = \sum_i \phi_i.$$  \hspace{1cm} (2)

Considering the error rate of devices as well as the error rate caused by concurrent transmissions, the overall error rate for any one of the participant, e.g., device-$i$, is defined as

$$r_i \triangleq \phi_i \cdot \left(e_i + \left(1 - e_i\right) \cdot f\left(\sum' \phi_i\right)\right),$$  \hspace{1cm} (3)

where $\left(1 - e_i\right) \cdot f\left(\sum' \phi_i\right)$ indicates the error rate of transmission for device-$i$ because: 1) the transmission only occurs when the device is normally in used, 2) the item $\left(1 - e_i\right)$ indicates that device-$i$ works well, and 3) concurrent number of transmissions should be considered.

3) **DIVERSE DEVICES**

We should mention here that different devices may have various error rates and data sizes. On the one hand, as shown in Table 3, various Android models own different failure rates according to the report [46] in 2017. On the other hand, the data sizes generated among devices are quite different, e.g., users own various sizes of subscribers in Reddit. Regarding the data generated from a user as a subreddit dataset, some statistics for subreddit datasets are shown in Table 4 [47]. Intuitively, more subscribers would incur more data, e.g., ClickLogs [3] and Blogs.

**TABLE 3. Top 10 failure rates of Android models [46].**

| Models (Top 10) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------|---|---|---|---|---|---|---|---|---|----|
| Failure Rate (%) | 9 | 6 | 5 | 4 | 3 | 3 | 2 | 2 | 2 | 2  |
The error rates regarding each type of devices could be derived from both related statistics and estimation. Since the error rates of various device types are quite different, it is suitable to classify the error rates according to their models as well as the regions during users’ usage. As a result, by assuming that in each round of training, the failure of each device in specific class-$k$ obeys Bernoulli distribution with unknown expected value $\mu_k$, we update the estimated error rate by

$$E_k = \frac{\sum_{r=1}^R \sum_{i \in S_k} a_{i,r}}{\sum_{r=1}^R \sum_{i \in S_k} b_{i,r}} \xrightarrow{a.s.} \mu_k,$$

where $a_{i,r}$ and $b_{i,r}$ are variables, indicating whether device-$i$ fails or device-$i$ participates in the $r$-th round, respectively. $S_k$ denotes the set of devices in class-$k$, and $R$ denotes the total number of rounds. Therefore, $\sum_{r=1}^R \sum_{i \in S_k} a_{i,r}$ denotes the failure samples in class-$k$, and $\sum_{r=1}^R \sum_{i \in S_k} b_{i,r}$ denotes the overall number of its samples. According to the Law of Large Numbers, the convergence of the estimated error rate could be ensured, i.e., it gets closer to the unknown expected value $\mu_k$. To evaluate such approach, we set the expected error rate of class-A, class-B, and class-C as 0.01, 0.05, and 0.1 relatively. Then we conduct thousands of updates based on the estimation and obtain the convergence of estimated error rates as shown in Fig 5, i.e., it gets closer to the expected value $\mu_k$ for each class-$k$.

**FIGURE 5. Convergence of estimated error rates.**

Since the overall error rate may vary due to various devices and the number of concurrent transmissions, for a set of devices, i.e., participants, we propose to minimize the maximal error rate among them so as to decrease the risk of transmission failures.

**B. PROBLEM FORMULATION**

With the system model, we formulate the following optimization problem, i.e., minimizing the maximal error rate of transmission failures so as to decrease the high risk of transmission failures in federated learning (RFL):

$$\min \quad \max_i \left\{ \phi_i \cdot (|e_i + (1 - e_i) \cdot f(\sum_{\phi})| \right\} \quad \text{[RFL]}$$

s.t. $$\sum_i (\phi_i \cdot d_i) \geq \rho \cdot \sum_i d_i,$$

$$\forall i : \phi_i \leq \pi_i,$$

$$\forall i : \phi_i \in \{0, 1\}.$$

Constraint (5) guarantees that the given ratio of involved training data to the whole dataset, i.e., $\rho$, the training quality of federated learning can be ensured since it is highly related to the accuracy of federated learning. Constraint (6) guarantees that participants are only chosen from those available devices. And at last, constraint (7) indicates the domain of an indicator variable, i.e., the domain of a decision variable is $\{0, 1\}$. Here, we use $OPT(RFL)$ to represent the value of the optimum for RFL.

When the number of concurrent transmissions $\sum \phi_i$ is fixed or even the error rate function $f$ is a constant value, the special case of the proposed problem RFL, i.e., the Integer Linear Programming (ILP) problem is NP-hard in canonical form [48]. Note that a NP-hard problem, whose decision counterpart is NP-Complete, cannot be solved efficiently if NP$\neq$P. Although there have been plenty of mature mathematical techniques for treating a special class [49] of problems efficiently by re-transforming them to related Linear Programming (LP) problems without violating their optimal solutions, some restrict assumptions are often emphasized. For example, the objective functions are often assumed to be convex and the problem should have a totally unimodal constraint matrix. In contrast, we prefer to design a general method for treating RFL when error rate function may be any differentiable function with respect to the sum of the decision variables. However, there is still a chance for us to know the lower bound of RFL, and use it as a guide to select suitable participants so as to minimize the concurrent number with guaranteed proportion of involved data.

**V. DESIGN OF RANDOMIZED ALGORITHM**

Thus, in this section, we propose to fix the value of $\sum \phi_i$ first, as well as to relax the domain of the decision variables in RFL to reals, obtaining the relaxed version of RFL, which can be solved efficiently by using mature linear programming (LP) techniques. Guided by the solution from relaxed version of RFL, we propose to select participants with a series of probabilities, which facilitates the theoretical analysis.

**A. RELAX VARIABLES IN RFL**

In order to simplify RFL to a practical problem, by fixing the value of $\sum \phi_i$ as well as relaxing the decision variables to reals, we transform RFL to a series of linear programs (a linear program for each fixed value of concurrent numbers), which can be solved efficiently by using mature
techniques [50]. Thus, we propose the relaxed version of RFL (rRFL) as follows:

\[
\min \max_i \{ p_i \cdot (1 - e_i) \cdot f(M) \} \quad [rRFL]
\]

s.t. \( \sum_i p_i = M, \)  
\( \sum_i (p_i \cdot d_i) \geq \rho \cdot \sum_i d_i, \)  
\( \forall i : p_i \leq \pi_i, \)  
\( \forall i : p_i \in [0, 1], \)

where we substitute \( [p_i] \in [0, 1] \) for reals \( \{\phi_i\} \in \{0, 1\} \) under fixed concurrent number of transmissions, i.e., \( M \). We denote by \( OPT_{r,M} \) the optimum of rRFL under fixed concurrent number \( M \), and denote by \( OPT_r \) the optimum among all rRFL instances with various \( M \in \mathcal{N} \), i.e.: 

\[
OPT_r = \min\{OPT_{r,1}, \ldots, OPT_{r,N}\}. \quad (12)
\]

After this transformation, we have \( |\mathcal{N}| \) sub-problems. Each of them is a linear programming (LP) problem, which can be solved efficiently by using mature techniques regarding the linear programming problems, such as simplex algorithm, ellipsoid algorithm, etc. The overall complexity is \( O(|\mathcal{N}| \cdot L) \), where \( L \) is the cost for a LP on \( |\mathcal{N}| \) devices.

**B. LP WITH PRUNING**

In order to obtain \( OPT_r \), intuitively, we have to enumerate all possible concurrent numbers from 1 to \( N \). However, the realistic scenario helps us to terminate the enumeration as early as possible if we have little space to improve the error rate caused by concurrent transmissions, as shown in lines 1 to 4 of Algorithm 1.

**Algorithm 1 Randomized Algorithm for RFL (ranRFL)**

Require: Error rates of devices \( \{e_i\} \); Raw data sizes \( \{d_i\} \); Availability of devices \( \{\pi_i\} \).

1. for Enumerate \( M \) from 1 to \( N \) with pruning do
2. \( \{p_i\}_M \leftarrow \) Solve rRFL(\( M \), \( \{e_i\} \), \( \{d_i\} \), \( \{\pi_i\} \));
3. end for
4. Choose \( \{p_i^*\} \) in \( \{p_i\}_M \), which minimizes rRFL;
5. Round \( \{p_i^*\} \) randomly twice to obtain \( \{\phi_1\}, \{\phi_2\} \);
6. Choose \( \{\phi_i\} \) from \( \{\phi_1\}, \{\phi_2\} \) with lower value;
7. Use \( \{\phi_i\} \) to select participants.

More specifically, due to the optimality of \( OPT_r \), if the error rate of transmissions under a larger concurrent number is higher than the result of any solved rRFL instance with small concurrent number, then there is no need for us to continue solving the rest rRFLs with higher concurrent numbers. Correspondingly, \( \forall M \in \mathcal{N} \), the condition of termination (i.e., the property of monotonically increasing) in the pruning of Algorithm 1 is listed as follows:

\[
\exists k \leq M : (1 - e_{\max}) \cdot f(M + 1) + e_{\min} > \mathcal{F}(k),
\]

s.t. \( \forall k \in \mathcal{N}, \mathcal{F}(k) := (1 - e_{\min}) \cdot f(k) + e_{\max}. \quad (13)\]

Based on such pruning, we can terminate the search and find the optimum \( OPT_r \) as early as possible, as well as find the optimal solution, i.e., \( \{p_i^*\} \) in line 4 of ranRFL. Note that the error rate increases with the growth of concurrent number, i.e., \( f(\cdot) \) is actually monotonically increasing in terms of the number of concurrent transmissions, as shown in our previous case study in Section III.

**C. RANDOMIZED Rounding**

To apply the optimal results of rRFL, i.e., \( \{p_i^*\} \), into a feasible solution of RFL, i.e., converting reals to integers, as shown in lines 5 to 7 of ranRFL, we use a randomized rounding strategy with the power of two choices. More specifically, \( \forall i \in \mathcal{N} \), we pick a positive decimal \( r_i \in [0, 1] \) randomly. Then, \( \phi_i \) equals to 1 if and only if \( r_i \leq p_i^* \), and \( \phi_i \) equals to 0 otherwise. Such rounding strategy takes the value of \( p_i^* \) into consideration, and is willing to select device- \( i \) into the training of federated learning with the probability of \( p_i^* \), where \( p_i^* \) is considered as the preference on choosing device- \( i \) by ranRFL, i.e., \( E[\phi_i] = p_i^* \). We should mention here that the randomized rounding strategy would not break constraint (6) because a non-zero \( p_i \) solved from rRFL indicates \( \pi_i = 1 \). After randomized rounding, \( \phi_i \) is at most 1. Constraint (5) is ensured through our analysis.

Although each device is selected according to a given probability, the bad event may still occur with a certain probability, which means the overall error rate caused by concurrent transmissions may be large by using rounded \( \{\phi_i\} \) from \( \{p_i^*\} \). In order to avoid such event as much as possible, we use the technique called the power of two choices [51], as shown in line 6 of ranRFL, which selects the one with lower error rate from two deployments, in terms of the result defined in RFL. After that, we select devices according to the scheduling derived from the power of two choices. Finally, the complexity of ranRFL is \( O(ML) \), where \( L \) is the cost for LP on at most \( M \) devices. The overall cost for running ranRFL is measured in the experiments as shown later.

**VI. THEORETICAL ANALYSIS**

In this section, we show the theoretical analysis on ranRFL. More specifically, the result of ranRFL concentrates on its optimum with high probability (illustrated in Theorem 1), and the proportion of involved data could be also ensured (illustrated in Theorem 2).

**Lemma 1 (Relationship Between RFL and rRFL): For a given instance of RFL and its related instances of rRFL, the optimum of rRFL is less than the optimum of RFL:**

\[
OPT_r \leq OPT(RFL). \quad (14)
\]

**Proof:** For any solution of RFL, i.e., \( \{\phi_i\}, \forall i : \phi_i \in \{0, 1\} \), it is also a solution of a specific rRFL, where the fixed concurrent number \( M \) equals to \( \sum \phi_i \). Thus, the optimal solution of RFL \( \{\phi_i^*\} \) is also a solution of a specific rRFL, whose concurrent number equals to \( M^* = \sum \phi_i^* \). Therefore, the optimum of rRFL is less than the optimum of RFL under
fixed concurrent number $M^*$. After that, we have

$$OPT_r = \min\{OPT_{r,1}, \ldots, OPT_{r,N}\} \leq OPT_{r,M^*} \leq OPT(RFL).$$

Lemma 2 (Analysis of Randomized Rounding): The maximal error rate of transmissions among $N$ devices is concentrated on its optimum with high probability by using the randomized rounding strategy, i.e., following inequality holds with the probability of $1 - O(e^{-t^2})$:

$$E[RFL] \leq OPT(RFL) + g(t),$$

where $\forall i \in \mathcal{N}$ : $g(t) \geq f(i + t) - f(i)$ captures the maximal growth of $f(\cdot)$ when concurrent number increases $t$ units.

Proof: The details of the proof are listed in Appendix A.

Theorem 1 (Analysis of $ranRFL$): The maximal error rate of transmissions among $N$ devices is concentrated on its optimum with high probability by using $ranRFL$, i.e., following inequality holds with the probability of $1 - O(e^{-2t^2})$:

$$E[RFL] \leq OPT(RFL) + g(t).$$

Proof: By using the power of two choices illustrated in $ranRFL$ as well as applying Lemma 2, the probability of the event in which inequality (16) is broken under both of two rounded deployments is at most $O(e^{-2t^2})$. Therefore, inequality (16) holds with the probability of at least $1 - O(e^{-2t^2})$ by using $ranRFL$ including the power of two choices as shown in Algorithm 1.

Theorem 2 (Analysis of Involved Data Proportion): By using $ranRFL$, the ratio of involved data to the whole dataset is expected to be satisfied, i.e., constraint (5) is ensured.

Proof: By using the designed randomized rounding strategy described in $ranRFL$, the proportion of involved data, i.e., the ratio of involved data to the whole dataset, is guaranteed:

$$E\left[\frac{\sum_i [\phi_i \cdot d_i]}{\sum_i d_i}\right] = \frac{\sum_i E[\phi_i \cdot d_i]}{\sum_i d_i} = \frac{\sum_i [p_i \cdot d_i]}{\sum_i d_i}. \quad (17)$$

Equation (17) holds since expectation is linear. Due to the fact that $\{p_i\}$ is solved from $rRFL$, such series of $\{p_i\}$ satisfies constraint (9). Therefore, the involved data proportion is at least $\rho$, i.e., $E\left[\frac{\sum_i [\phi_i \cdot d_i]}{\sum_i d_i}\right] \geq \rho$.

VII. EXPERIMENTAL STUDY

In this section, extensive simulations are conducted to show the performance of $ranRFL$. Compared with the state-of-the-art schemas, $ranRFL$ decreases the error rate of transmissions by up to 38.3%, through averaging the maximal improvements among all scenarios, considering both participant selection and the number of concurrent transmissions.

A. DATA AND SETTINGS

1) MOBILE EDGE NETWORKS

The mobile edge network in our simulations supports shared usage of concurrent transmissions. We use a nine-week trace dumped from a typical U.S. Air Force LAN described in previous section to represent the error rate of transmissions under different concurrent numbers. More specifically, we use the fitted curves in Fig. 3 as the function $f$ defined in our proposed problem. The number of mobile devices in our simulations ranges from 1 to 40 in the network as available candidates, each of which has an error rate ranges from 0 to 0.25.

2) FEDERATED LEARNING

According to the literature [2], [10], [11], raw user data are often generated unevenly, i.e., the data sizes are skewed among mobile devices. As mentioned in previous work [52], we use Zipf distribution, whose default skewed parameter is 1.2, ranging from 1 to 4, to describe the sizes of raw user data among mobile devices. As shown in Fig. 6, when Zipf skewed parameter is as small as 1, the skewness of data among devices is slight. With the growth of the candidate number (maximum to 500), the CDF seems to be the same as that under 100 candidates. In contrast, when the Zipf skewed parameter is as high as 4, the skewness of data is severe. Although continuously increasing the Zipf skewed parameter would result in a more skewed data distribution, data distribution under Zipf skewed parameter 4 has already been unbalanced enough. We use $F$ distribution as a substitute to describe the data distribution, whose parameters range from (1, 1) to (500, 500) and the default is (8, 3).

3) ALGORITHMS AND METRICS

Except for our proposed algorithm $ranRFL$, we also compare our schema with other two algorithms: 1) GDS always greedily chooses the device with the largest raw Data Size before reaching to the given ratio, and 2) SLSQP [53] uses the Han-Powell quasi-Newton method [54] with a BFGS [55] update in the gradient based algorithm, which solves RFL based on the estimation.

We should mention here that our primary metric is the maximal error rate of model updates among participants, and all of our illustrated results are based on 10 attempts on the laptop with Intel Core i9-9900HK and Python 3.7. The large scale of simulations with various data distributions and error rates of devices for evaluating the scalability are conducted on PowerEdge R740 with 30 cores and 40GB memory.

FIGURE 6. Impact of Zipf skewed parameters.
B. EVALUATION RESULTS

1) EFFECTIVENESS OF runRFL

As shown in Fig. 7, we evaluate runRFL under various skewness parameters of data distributions as well as various error rates of devices to show the effectiveness of runRFL through extensive simulations, which covers a wide range of realistic settings.

Fig. 7(a) and Fig. 7(b) illustrate the results given high data proportion (0.8). The parameters of Zipf distribution (Dis.) range from 1.1 to 1.3, i.e., $Z_S$, $Z_M$ and $Z_L$ equal to 1.1, 1.2 and 1.3, respectively, while the parameters of F Dis. range from (300, 300) to (500, 500), (400, 400) and (300, 300). Note that $S$, $M$ and $L$ refer to the skewness. We also vary the error rates of devices for each target data distribution. For Zipf Dis., error rates of devices range from 0.03 to 0.07, i.e., $E_S$, $E_M$ and $E_L$ equal to 0.03, 0.05 and 0.07, respectively. runRFL decreases the error rate of model updates by up to 21.44%. For F Dis., error rates of devices range from 0.01 to 0.1, i.e., $E_S$, $E_M$ and $E_L$ equal to 0.01, 0.05 and 0.1, respectively. runRFL decreases the error rate of model updates by up to 31.35% compared with other two strategies, also illustrating the effectiveness of runRFL under wide range of realistic settings.

Fig. 7(c) and Fig. 7(d) illustrate the results given middle data proportion (0.6). The parameters of Zipf Dis. range from 2.5 to 3.5 while the parameters of F Dis. range from (1, 1) to (20, 20). For Zipf Dis., error rates of devices range from 0.18 to 0.38. runRFL decreases the error rate of model updates by up to 51.5% compared with other two strategies. For F Dis., error rates of devices range from 0.1 to 0.3. runRFL decreases the error rate of model updates by up to 34.1% compared with other two strategies.

2) CHARACTERISTICS OF runRFL

Fig. 8(a) to Fig. 8(c) show the characteristics of participant selection by all of these three strategies. GDS greedily selects those with large proportions of data, as shown in Fig. 8(a). Compared with SLSQP, runRFL selects more devices with larger proportions of data as well as with small error rates of devices, which helps decrease the overall error rate on model updates, as shown in Fig. 8(b) and Fig. 8(c). runRFL only spends at most 0.09s for solving a series of rRFL as well as conducting randomized rounding, which is more realistic than that of SLSQP, i.e., at most 0.34s. Although GDS, a greedy based algorithm, spends less than runRFL, runRFL performs much better than GDS.

We illustrate the values of $M$ under four various scenarios, as shown in Fig. 8(d). When $M$ is small, e.g., it gets closer to 1, the linear programming solver cannot find a feasible solution unless there is a device occupying most of the data samples used in federated learning. As $M$ grows larger, there must exist a minimal value $M_{\text{min}}$, in which the linear programming solver succeeds. When there is an increasing tendency on the error rates, it is suitable to conduct the pruning part, since those remaining values of $M$ would only increase the error rates continuously. As a result, the optimal solution
Y. Yan et al.: Risk Minimization Against Transmission Failures of Federated Learning in Mobile Edge Networks

![Figure 9](image1)

**FIGURE 9.** Evaluating ranRFL with various number of candidates under Zipf and F distributions. ranRFL decreases the error rate by up to 39.6% and 51.1%, respectively. Further evaluating ranRFL with various ratios of involved data to the whole dataset under Zipf and F distributions. ranRFL decreases the error rate by up to 81.0% and 72.9%, respectively, compared with state-of-the-art schemas.

![Figure 10](image2)

**FIGURE 10.** Results in terms of the improvement on error rate under various settings to show the scalability of ranRFL, covering a wide range of realistic scenarios, including the changes on data distributions and error rates of devices. ranRFL decreases the error rate by up to 28.15% and 13.54%, respectively, under Zipf and F distributions.

would not appear for large values of $M$. Furthermore, the cost for running ranRFL is at most 77ms. In contrast, SLSQP spends at most 346ms.

3) **VARIOUS NUMBERS OF CANDIDATES**

As shown in Fig. 9(a), ranRFL decreases the maximal error rate by up to 39.6% compared with other two strategies when the data sizes obey Zipf distribution. With the increase on the number of available candidate devices, ranRFL has more opportunities to select participants from these candidates with lower error rates of model updates. Therefore, the maximal error rate grows slower than that of other two strategies. GDS selects those devices with larger sizes of raw user data, but ignore to consider the error rate of devices during their normal uses. SLSQP uses a gradient based algorithm with approximate estimation on $f$, leading to an approximate solution of RFL. In contrast, ranRFL selects participants based on the lower bound of RFL’s optimum, i.e., the result from its relaxed version. Further shown in Fig. 9(b), similarly, ranRFL decreases the maximal error rate among participants by up to 51.1% compared with other two strategies under F distribution.

4) **VARIOUS PROPORTIONS OF INVOLVED DATA**

We also evaluate ranRFL under various ratios of involved training data to the whole dataset by using different distributions.

As shown in Fig. 9(c), under Zipf distribution with default parameter 2, ranRFL decreases the error rate by up to 81% compared with other strategies. At the same time, the average error rate decreases at least 37.8% under various scenarios with different proportions of involved data, ranging from 0.1 to 0.7. As further shown in Fig. 9(d), under F distribution with parameter $(1, 1)$, ranRFL decreases the error rate by up to 72.9%. At the same time, ranRFL decreases at least 23.9% on average error rate under various scenarios with different proportions of involved data. When the proportion of involved data is too large or small, ranRFL has less opportunities to select participants. Therefore, ranRFL improves less than those scenarios, where the proportion of involved training data ranges from 0.4 to 0.6.

5) **SCALABILITY OF ranRFL**

ranRFL and other two strategies are also evaluated under various scenarios. As shown in Fig. 10, the improvements of ranRFL compared with SLSQP and GDS under both Zipf and F distributions are illustrated so as to show the scalability of ranRFL under diverse scenarios.

As shown in Fig. 10(a), under Zipf distribution with the parameters ranging from 1 to 4, as well as the error rates of devices ranging from 0.01 to 0.15, compared with GDS, ranRFL decreases the error rate by at most 21.4%, 26.8%, 28.3%, and 57.4% respectively. Besides, the reductions are 28.1%, 31.4%, 35.7%, and 60.4% compared with SLSQP.
In general, ranRFL improves much compared with these two state-of-the-art strategies. With the growth of Zipf skewed parameter, data sizes are more skewed among few devices. Thus, by carefully selecting excluded devices, the resource usage could be reduced much while the data proportion could be ensured. As further shown in Fig. 10(c), under F distribution with the parameters ranging from (3, 1) to (11, 11), as well as the error rates of devices ranging from 0.01 to 0.15, the improvement of ranRFL reaches 13.10% compared with SLSQP. Similarly, as shown in Fig. 10(d), the improvement reaches 13.54% compared with GDS.

**APPENDIX A**

**PROOF OF THEOREM 1**

**Theorem 1:** The maximal error rate of transmissions among \( N \) devices is concentrated on its optimum with high probability by using randomized rounding strategy, i.e., following inequality holds with the probability of \( 1 - O(e^{-2t^2}) \):

\[
E[RFL] \leq OPT(RFL) + g(t). \tag{18}
\]

**Proof:** First of all, \( \forall j \in \mathcal{N} \), we define a series of variables as

\[
X_j = \sum_{i=1}^j \phi_i, \quad Y_j = X_j - E[X_j], \tag{19}
\]

where \( X_j \) is the cumulative count of participants from device-1 to device-\( j \), \( \forall j \in \mathcal{N} \), and \( Y_j \) describes the distance between actual number of concurrent transmissions and its expectation. Note that \( X_N \) is the actual number of concurrent transmissions. After analyzing consequence \( \{Y_j\} \), we have

\[
E[Y_j | \phi_1, \ldots, \phi_{j-1}] = E[Y_{j-1} | \phi_1, \ldots, \phi_{j-1}] + E[\phi_j - E[\phi_j] | \phi_1, \ldots, \phi_{j-1}] = Y_{j-1} + E[\phi_j | \phi_1, \ldots, \phi_{j-1}] - E[E[\phi_j] | \phi_1, \ldots, \phi_{j-1}] = Y_{j-1} + \phi_j - \phi_j = Y_{j-1}, \tag{20}
\]

where \( E[E[\phi_j] | \phi_1, \ldots, \phi_{j-1}] \) equals \( E[\phi_j] \) because \( \phi_j \) is selected independently by only using \( p_i \). Equation (20) implies that \( \forall j \in \mathcal{N}, E[Y_j | \phi_1, \ldots, \phi_{j-1}] \) equals \( Y_{j-1} \). That means sequence \( \{Y_j\} \) is a martingale sequence after defining \( Y_0 = 0 \) for completeness. Analyzing two consecutive items, we have

\[
|Y_j - Y_{j-1}| = |\phi_j - E[\phi_j]| \leq \max\{p_j, 1 - p_j\} = c_j.
\]

After applying Azuma’s Inequality [56] on martingale sequence \( \{Y_j\} \), we have

\[
Pr[|Y_N - Y_0| \geq t] \leq 2exp\left(-\frac{t^2}{2 \sum_{j=1}^{j=N} c_j^2}\right), \tag{21}
\]

which equals to the following inequality holds with the probability of at least 1 - \( O(e^{-t^2}) \):

\[
\sum_i \phi_i \leq E[\sum_i \phi_i] + t = \sum_i p_i^* + t. \tag{22}
\]

Since function \( f(\cdot) \) is given and non-decreasing, following inequality holds with the probability of at least 1 - \( O(e^{-t^2}) \):

\[
f(\sum_i \phi_i) \leq f(\sum_i p_i^* + t). \tag{23}
\]

After that, we analyze the expectation of the maximal error of concurrent transmissions. \( \mathcal{S}_t = [\sum_i p_i^*, \sum_i p_i^* + t] \cap \mathcal{N} \) is defined as the set of integers ranging from \( \sum_i p_i^* \) to \( \sum_i p_i^* + t \). Then, for any given integer \( I \in \mathcal{S}_t \) and \( \forall i \in \mathcal{N} \), we have

\[
E[\phi_i | \{e_i + (1 - e_i) \cdot f(I)\}] = p_i^* \cdot \{e_i + (1 - e_i) \cdot f(I)\}.
\]

Then, assuming \( u = \text{argmax} \{e_i + (1 - e_i) \cdot f(I)\} \), the maximal error rate of RFL can be treated as:

\[
E[RFL] = E[\phi_u | \{e_u + (1 - e_u) \cdot f(I)\}] = p_u^* \cdot \{e_u + (1 - e_u) \cdot f(I)\}. \tag{24}
\]

As a result, inequality (24) equals to that following inequality holds with the probability of at least 1 - \( O(e^{-t^2}) \):

\[
E[RFL] \leq p_u^* \cdot \{e_u + (1 - e_u) \cdot E[I|f(I)]\} \leq p_u^* \cdot \{e_u + (1 - e_u) \cdot \sum_{i \in \mathcal{S}_t} Pr[\sum_i \phi_i = I] \cdot f(I)\}. \tag{25}
\]

Since \( \sum_i \phi_i = I \), after applying inequality (23), we have

\[
\sum_{i \in \mathcal{S}_t} \{Pr[\sum_i \phi_i = I] \cdot f(I)\} \leq \sum_{i \in \mathcal{S}_t} \{Pr[\sum_i \phi_i = I] \cdot f(\sum_i p_i^* + t)\} = f(\sum_i p_i^* + t) \cdot \sum_{i \in \mathcal{S}_t} \{Pr[\sum_i \phi_i = I]\} = f(\sum_i p_i^* + t). \tag{26}
\]

where the last equation holds due to the law of total probability. Applying inequality (26) to (25), we have

\[
E[RFL] \leq p_u^* \cdot \{e_u + (1 - e_u) \cdot f(\sum_i p_i^* + t)\} \leq \max_i \{p_i^* \cdot \{e_i + (1 - e_i) \cdot f(\sum_i p_i^* + t)\}\}. \tag{27}
\]
where the first inequality holds due to (23). We assume that function $g(t)$ captures the growth of $f$. That means it is larger than the maximal increase on $f$, whose abscissa length is at most $t$, i.e.:

$$\forall i \in N : g(t) \geq f(i + t) - f(i).$$  \hspace{1cm} (28)

Then, (27) equals to that following inequality holds with the probability of at least $1 - O(e^{-t^2})$:

$$E[RFL] \leq \max \left\{ p_i^* \cdot (1 - e^{-t}) \cdot \left( f \left( \sum_{j=p_i^*} g(t) \right) \right) \right\}$$

$$\leq \max \left\{ p_i^* \cdot (1 - e^{-t}) \cdot f \left( \sum_{j=p_i^*} g(t) \right) \right\} + g(t)$$

$$= OPT_t + g(t).$$  \hspace{1cm} (29)

REFERENCES

[1] W. Yang Bryan Lim, N. Cong Luong, D. Thai Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, “Federated learning in mobile edge networks: A comprehensive survey,” 2019, arXiv:1909.11875. [Online]. Available: http://arxiv.org/abs/1909.11875

[2] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agpager y Arcas, “Communication-efficient learning of deep networks from decentralized data,” 2016, arXiv:1602.05629. [Online]. Available: http://arxiv.org/abs/1602.05629

[3] A. Vulimiri, C. Curino, P. B. Godfrey, T. Jungblut, J. Padhye, and G. Varghese, “Global analytics in the face of bandwidth and regulatory constraints,” in Proc. USENIX NSDI, 2015, pp. 323–336.

[4] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, “Multi-key privacy-preserving deep learning in cloud computing,” Future Gener. Comput. Syst., vol. 74, pp. 76–85, Sep. 2017.

[5] B. Wahl, A. Cossey-Gantner, S. Germann, and N. R. Schwabhe, “Artificial intelligence (AI) and global health: How can AI contribute to health in resource-poor settings?” BMJ Global Health, vol. 3, no. 4, Aug. 2018, Art. no. e00798.

[6] P. Qiu, G. Anantanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl, and I. Stoica, “Low latency geo-distributed data analytics,” ACM SIGCOMM Comput. Commun. Rev., vol. 45, no. 4, pp. 421–434, Sep. 2015.

[7] A. Jahangiri and H. A. Rakha, “Applying machine learning techniques to transportation mode recognition using mobile phone sensor data,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 9, pp. 5991–6004, Sep. 2018.

[8] K. Lueth, “State of the IoT 2018: Number of IoT devices now at 7.5 billion and accelerating,” IoT Anal., Aug. 2018. [Online]. Available: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7-5bn

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

[10] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. Brendan McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, “Towards federated learning at scale: System design,” 2019, arXiv:1902.01046. [Online]. Available: http://arxiv.org/abs/1902.01046

[11] (1999) A Nine-Week Rawdata Dumped From A Typical U.S. Air Force Base. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[12] (2016) L. Cheng, Y. Gu, J. Niu, T. Zhu, C. Liu, Q. Zhang, and T. Hel, “Taming collisions for delay reduction in low-duty-cycle wireless sensor networks,” in Proc. 55th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM), Apr. 2016, pp. 1–9.
YUTING YAN was born in Hailar, Inner Mongolia, in 1999. She is currently pursuing the B.S. degree in financial engineering with the Business School, Nanjing University, Nanjing, Jiangsu, China. Her research interests include distributed computing, wireless networks, and mobile edge computing.

SONGHUA LIU is currently pursuing the B.S. degree. He was with the Department of Computer Science and Technology, Nanjing University, Jiangsu, China, in 2017. His research interests include mobile edge networks, federated learning, and collaborative training.

YIBO JIN (Member, IEEE) received the B.S. degree from Nanjing University, in 2017, where he is currently pursuing the Ph.D. degree. He was a Visiting Student with Hong Kong Polytechnic University, in 2017. His research interests include big data analytics, edge computing, and federated learning.

ZHUZHONG QIAN (Member, IEEE) received the Ph.D. degree in computer science, in 2007. He is an Associate Professor with the Department of Computer Science and Technology, Nanjing University, China. He is the Chief Member of several national research projects on cloud computing and pervasive computing. He has published more than 30 research articles in related fields. His current research interests include cloud computing, distributed systems, and pervasive computing.

SHENG ZHANG (Member, IEEE) received the B.S. and Ph.D. degrees from Nanjing University, in 2008 and 2014, respectively. He is an Associate Professor with the Department of Computer Science and Technology, Nanjing University. He is also a member of the State Key Laboratory for Novel Software Technology. He has published more than 70 articles, including those appeared in IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON COMPUTERS, Mobile-Hoc, ICDCS, INFOCOM, I WoQoS, and ICPP. His research interests include cloud computing and edge computing. He is a Senior Member of the CCF. He received the Best Paper Runner-Up Award from the IEEE MASS 2012. He was a recipient of the 2015 ACM China Doctoral Dissertation Nominations Award.

SANGLU LIU (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Nanjing University, in 1992, 1995, and 1997, respectively. She is currently a Professor with the Department of Computer Science and Technology and the State Key Laboratory for Novel Software Technology. Her research interests include distributed computing, wireless networks, and pervasive computing. She has published over 80 articles in refereed journals and conferences in the above areas.