Eiffel: Efficient and Fair Scheduling in Adaptive Federated Learning

Abeda Sultana, Md. Mainul Haque, Li Chen, *Member, IEEE,* Fei Xu, *Member, IEEE,* and Xu Yuan, *Senior Member, IEEE*

Abstract—Emerging machine learning (ML) technologies, in combination with the increasing computational power of mobile devices, lead to the extensive adoption of ML-based applications. Different from conventional model training that needs to collect all the user data in centralized cloud servers, federated learning (FL) has recently drawn increasing research attention as it enables privacy-preserving model training. With FL, decentralized edge devices in participation, train their model copies locally over their siloed datasets, and periodically synchronize the model parameters. However, model training is computationally extensive which easily drains the battery of mobile devices. In addition, due to the uneven distribution of siloed datasets, the shared model may become biased. To address the *efficiency* and *fairness* concerns in a resource-constrained federated learning setting, in this paper, we propose *Eiffel* to judiciously select mobile devices to participate in the global model aggregation, and adaptively adjust the frequency of local and global model updates. *Eiffel* aims to make scheduling and coordination for the federated learning towards both resource efficiency and model fairness. We have conducted theoretical analysis of *Eiffel* from the perspectives of fairness and convergence. Extensive experiments with a wide variety of real-world datasets and models, both on a networked prototype system and in a larger-scale simulated environment, have demonstrated that while maintaining similar accuracy performance, *Eiffel* outperforms existing baselines with respect to reducing communication overhead by up to 6$\times$ for higher efficiency and improving the fairness metric by up to 57% compared to the state-of-the-art algorithms.

Index Terms—Federated Learning, Fairness, Efficiency, Scheduling, Resource Constraints

1 Introduction

With the ever-growing computation capability and the extensive adoption of mobile devices (*e.g.*, smartphones, wearable medical devices, sensory equipment) in today's era of Internet-of-Things, an astronomical amount of data are generated daily over the network. According to a recent survey of Cisco, IoT devices will account for 50% (14.7 billion) of all global networked devices by 2023 [1]. Each edge device is producing massive amount of data every year, which can be naturally leveraged by user-interactive applications driven by machine learning techniques. Typically, a machine learning model is trained in a centralized fashion where a datacenter gathers input data from all the participating edge devices. As one might anticipate, this is not a suitable method of model training for edge devices due to privacy sensitivity of user data and communication burden incurred by transferring massive raw data.

To overcome these limitations, *federated learning* [2] has emerged as an attractive paradigm for decentralized machine learning across edge devices. Instead of aggregating raw data from user devices to a centralized server, federated learning enables client devices to collaboratively participate in the computation process on their local data towards learning a shared model. In particular, in a typical iterative training process, each device calculates on its local data, sends local update of model parameters for global aggregation and pulls the updated parameters for the next iteration. In this way, user data will be kept at local device rather than being sent to a remote server, thus preventing the privacy leakage[1].

Despite the salient advantages from the privacy-preserving perspective, there are unique challenges and open problems that remain under-explored in federated learning. The uppermost issue to tackle is the constrained resources on edge devices, including the limited network bandwidth and various network latency that render the communication stage a bottleneck in the model training process. Moreover, as input data are distributed across millions of devices in a highly uneven fashion (not independent-and-identically-distributed, *i.e.*, non-i.i.d.) [3] and devices are not always available to participate in the training due to dynamics on power condition or network connection [4], the model training performance, with respect to accuracy and convergence, is negatively impacted.

To enable efficient federated learning in such a resource-constrained environment, existing works have investigated

- *A. Sultana, M.M. Haque, L. Chen and X. Yuan are with School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA 70504, USA.*
  E-mail: abeda.sultana1@louisiana.edu, md-mainul.haque1@louisiana.edu, li.chen@louisiana.edu, xu.yuan@louisiana.edu
- *F. Xu is with Department of Computer Science and Technology, East China Normal University, Shanghai, China. E-mail: fxu@cs.ecnu.edu.cn*

This research was supported in part by NSF under grants 2019511, 1763620, and 1948374, in part by the Louisiana Board of Regents under Contract Number LEQSF(2019-22)-RD-A-21, in part by the NSFC under grant No.61972158 and the Science and Technology Commission of Shanghai Municipality under grant No.20511102802 and No.18DZ2270800.

1. Note that there may be some indirect privacy leakage from the model updates when a potential adversary can infer some sensitive attributes, which is out of the scope of our consideration.

a number of approaches ([3]–[14], *etc.*) to reducing the communication overhead, including gradient compression [8], sparsification [15], less frequent synchronization [4], *etc.*

On the other hand, considering the uncertain availability of mobile devices, the dynamic selection of participating devices brings another degree of freedom towards training efficiency [11], [12], [16]. Apart from the *efficiency* issue, an equally important concern is *fairness*, which has not yet been amply investigated in the context of resource-constrained federated learning.

In this paper, we aim to design an *efficient* federated learning scheme with the guaranteed degree of device-level *fairness*, extending the state-of-the-art fairness proposal [17] to a resource-constrained setting. In particular, our notion of fairness is defined with respect to the model loss (or accuracy) distribution among user devices, such that the enforcement of fairness can attenuate the possibility of learning a biased global model.

To achieve these objectives, we present our design of *Eiffel*, an **E**ff**I**cient and **F**air scheduling algorithm for **FE**derated **L**earning, in a resource-constrained environment. Having observed that involving the complete set of a large number of devices for model update is impractical and suboptimal in terms of prolonging communication time, *Eiffel* selects participating edge devices and controls the frequency of global model update aggregation in an adaptive manner. Particularly, a variety of factors, including the local loss, data size, computation power, resource demand and age of update of each user device, are comprehensively considered. Participants are dynamically selected and coordinated through the learning process, to achieve the best resource efficiency given a fixed budget while ensuring fairness.

We have theoretically analyzed the model fairness achieved by our algorithm with two metrics: the variance of performance distribution and the cosine similarity between the performance vector and all-ones vector. Both metrics are used to evaluate the uniformity of loss (or accuracy) distribution among the devices resulted from the proposed algorithm.

We have further conducted convergence analysis of our algorithm for convex models and derived the upper bound of the difference between the resulted loss function with the optimal loss. Based on the convergence bound and under a resource budget, the frequency of the global model update is calculated following the adaptive framework [4] to minimize the loss function.

Through extensive experiments using real-world datasets on both a hardware setting and in a larger-scale simulated environment for convex and non-convex models and for different data distributions, we demonstrate the fairness and efficiency of our proposed approach. *Eiffel* performs fairer than the state-of-the-art q-FFL [17] by resulting in at least 49% less variance in terms of loss distribution. It also achieves at least 60% smaller variance compared to RS (random selection) and 100% compared to LLS (a better performance-based selection algorithm), under both i.i.d. and non-i.i.d. settings for various models. With respect to model accuracy, the results demonstrate that *Eiffel's* performance remains similar to the adaptive federated learning baseline [4] in both non-i.i.d. and i.i.d. settings for the convex model. For computation-intensive model training, *i.e.*, a complex model CNN trained on CIFAR-10 dataset, *Eiffel* saves the communication overhead by up to 6.45×, thanks to the efficient device selection algorithm and the strategically calculated global aggregation frequency, while sacrificing the accuracy performance by less than 5% compared to [4] for the non-i.i.d setting. To summarize, our extensive experimental results confirm the effectiveness of our proposed approach in achieving model fairness for different machine learning models and data distribution settings. Compared to the state-of-the-art adaptive federated learning baseline, *Eiffel* is able to achieve similar model performance while significantly improving the communication efficiency.

The remainder of this paper is organized as follows. We discuss the state-of-the-arts and differentiate our work from the existing literature in Section 2. The system model and problem setting are presented in Section 3. We then present our design of *Eiffel* in Section 4, and conduct theoretical analysis of its fairness and convergence in Section 5. Our evaluation setting and experimental results are presented in Section 6 to demonstrate the advantages of our solution. Finally, we discuss the practical aspects and future directions in Section 7, and conclude the paper in Section 8.

2 State-of-the-Art and Motivation

Due to the privacy concern of raw data generated and stored at edge devices, federated learning [2], without exposing raw data, has been increasingly employed by large companies and organizations for machine learning tasks across thousands to millions of user devices [18]. Unique challenges and open problems come long with its promising advantages to increasingly draw research attention, including uneven data distribution (non-i.i.d.) across devices, constrained resources (power condition), network dynamics (bandwidth, latency) which impact the communication stage, *etc.*

Under the resource-constrained learning environment, existing works have proposed a variety of approaches towards *efficiency* improvement, such as reducing the communication traffic volume with gradient compression ([8], [19]) or sparsification [15], reducing the communication frequency by adaptive model synchronization([4], [7]), reducing the number of communicating entities through dynamic participant selection ([11], [12], [16], [20]), *etc.* In particular, participant selection has become a prevailing problem to be addressed in federated learning, where edge devices are not always available to participate. A selection algorithm was proposed in [11] to randomly select user devices as many as possible without violating resource constraints. Amiri *et al.* [12] scheduled devices based on the channel condition and the significance of local model updates. Yang *et al.* [21] proposed an analytic model on the performance of federated learning given a set of scheduling schemes and inter-cell interference. The factor related to the staleness of model updates for user devices is introduced in [13] for the scheduling decision. Recently proposed participant selection algorithm, Power-Of-Choice [22], establishes that biasing the devices with higher local losses increases the rate of convergence compared to unbiased participant selection. SCAFFOLD [23] states that FEDAVG [2] suffers from the

"client-drift" issue due to the non-i.i.d. data which results in slow convergence. SCAFFOLD handles this issue by modifying the local loss calculation. Oort [20] schedules user devices for participation based on their statistical utilities defined by the loss values of local models and their global system utilities determined by device speeds. Convergence analysis has also been conducted by recent studies [4], [24], [25] for federated learning with different client selection settings, *i.e.*, all participation and selective participation of client devices. Different from synchronous federated learning as aforementioned, some recent studies focus on addressing the challenges of asynchronous federated learning. SAFA [26] enables a deadline for receiving parameter update from the participating user devices. Thus it can distinguish the straggler participants and also take necessary steps to update the model with stale parameters. FLEET [27] also enables stale updates but is adjoined with a dumpening factor to give smaller weights as staleness increases.

Apart from the efficiency goal, another important concern is the *fairness* with respect to how the collaboratively learned model performs (measured by loss value or accuracy level) across user devices.

A common definition of fairness in machine learning is with respect to the *accuracy parity* across protected groups [28]. Such a fairness cannot be trivially extended to federated learning, since it makes no sense to ensure identical accuracy on each device given the significant variation among the data. *Good-intent fairness* [16] was introduced to address this issue to some extent, by maximizing the performance of the worst performing device.

Li *et al.* [17] and Huang *et al.* [29] proposed algorithms to achieve the fairness which is defined as the distribution of model accuracy across devices. Collaborative Fairness [30] regulates that the participants will get different model parameter updates based on their contributions. Such a fairness definition is important for business models in biomedical or financial institutions to make predictions in practice. A similar idea to incentivize contributors owing high quality data is proposed by Yu *et al.* [31]. The algorithm provides a dynamic payoff-sharing scheme that distributes budgets among data owners to maximize the utility and minimize the inequality. HFFFL [32] presents a similar reward mechanism for data contribution among the clients. It ensures proportional fairness by categorizing clients into different levels for collaboration. FairFed [33] presents a mechanism to detect adversarial devices and reject their model updates. FLASH [34] is a heterogeneity-aware fair algorithm which considers heterogeneity in device type (in terms of hardware variety) and user behavior (in terms of device status, such as idle, charging, connected to a slow network).

In sharp contrast to these approaches, our proposed solution takes into account both the resource efficiency and the model fairness. To the best of our knowledge, we are the first to incorporate both the adaptive update frequency and the selection of user devices per round in the synchronous federated learning setting, achieving the best utilization of limited resources while ensuring fairness of the learned model. With respect to the fairness notion, *Eiffel* ensures fair model distribution among heterogeneous devices in the strategical selective setting, considering different data distribution and computation power across devices.

Notation	Meaning
$F_i(\theta)$	Local loss function at node i
$F(\theta)$	Global loss function
D_i	Dataset at node i
$\theta_i(t)$	Local model parameter at node i in iteration t
$\theta(t)$	Global model parameter in iteration t
η	Gradient descent step size
τ	Number of local update steps between two global aggregations
T	Total number of local update steps at each node
K	Total number of global aggregation steps, equal to T/τ
δ	Gradient divergence defined in Definition 1
$h(\tau)$	Defined in Eq. (11), gap between the model parameters obtained from distributed and centralized gradient descents
ρ	Lipschitz parameter of $F_i(\theta)$ for all i and $F(\theta)$
β	Smoothness parameter of $F_i(\theta)$ for all i and $F(\theta)$
c_m	Resource (type m) consumption in one local update step
b_m	Resource (type m) consumption in one global aggregation step
R_m	Resource (type m) budget

TABLE 1: Main notations in system model.

3 System Model

In this section, we present our system model, with main notations summarized in Table 1. The preliminaries and basics of adaptive federated learning are presented in our problem setting, followed by the elaboration on the efficiency and fairness requirements.

3.1 Federated Learning

In federated learning, each participating device maintains a model copy, trains the copy with its local data and communicates the model parameter update through an aggregator (*e.g.*, an edge server).

Consider a mobile edge computing system consisting of N user equipment (or UEs, interchangeably used for user devices) and an aggregator, as illustrated by Fig. 1. To collaboratively learn a shared model with federated learning, all the UEs perform local update computing and their parameter updates are averaged in the aggregator. In particular, for the n-th UE, we denote its local dataset as $D_n = \{x_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}_{i=1}^{|D_n|}$ with size $|D_n|$, where $|.|$ denotes the cardinality, x_i is the input of the machine learning model with cardinality d and y_i is the desired output. For dataset D_n at UE n, the loss function associated with this UE can be represented as

$$F_n(\theta) \triangleq \frac{1}{|D_n|} \sum_{i \in D_n} f_i(\theta),$$

where $f_i(\theta)$ is the loss function defined on its parameter vector θ for each data point i. Let us denote the entire set of data by $D \triangleq \sum_{n=1}^{N} D_n$. Then the global loss function on all the distributed datasets can be expressed as

$$F(\theta) \triangleq \frac{\sum_{n=1}^{N} D_n F_n(\theta)}{D}.$$

The objective of federated learning is to learn the model with the non-i.i.d. data residing in the UEs, by minimizing a particular loss function $F(\theta)$ to find $\theta^* = arg\ min F(\theta)$.

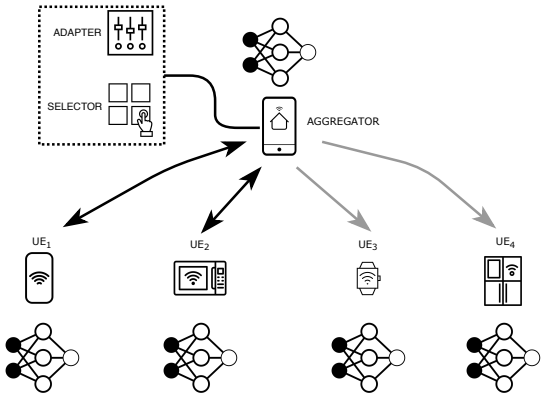


Fig. 1: The federated learning setup for our proposed approach. Each *User Equipment (UE)* or device is preloaded with the model. An *Aggregator* selects (using Selector) a subset of UEs to receive model parameters (2-way darker arrows are used to denote those selected UEs) after each adaptive number of local iterations.

3.2 Resource Efficiency

The resource efficiency in federated learning can be intuitively interpreted as how fast a model can be learned by coordinating the resources on multiple user equipments across time. The resources are generally related to computation and communication, which are constrained in the federated learning setting with low-end mobile devices, as opposed to the conventional machine learning training in centralized datacenters. More formally, resource efficiency is evaluated by the performance of the global model, after a given number of iterations for learning. For the purpose of maximizing resource efficiency in a resource-constrained environment, an optimally designed *adapter* is desired to determine the frequency of local updates and global aggregations [4]. Let τ denotes the number of local model iteration between each consecutive global aggregation, and T is the total number of local iterations required to complete the learning at each node. We further use K to represent the total number of global aggregations through the learning process. Assuming that T is an integer multiple of τ, we have $K = \frac{T}{\tau}$. Upon the completion of training, the learned model parameters, denoted as θ^f, is defined as

$$\begin{aligned} &\theta^f \triangleq argmin\ F(\theta), \\ &\theta \in \{\theta(k\tau) : k = 0, 1, 2...K\}. \end{aligned} \tag{1}$$

For each user device, different types of resources are required to participate in federated learning. Given a total of M types of resources, each participating UE consumes c_m units of resource with type $m \in \{1, 2, ..., M\}$ in every local update step, and consumes b_m units at each global aggregation step, where $c_m \geq 0$, $b_m \geq 0$ and both are finite real values.

Throughout the training process with T iterations, the total amount of type-m resource to be consumed is $(T+1)c_m + (K+1)b_m$, where the additional "+1" is for computing $F(\theta(K\tau))$. This amount should not exceed a given budget R_m. Essentially, the adaptive federated learning needs to determine the optimal τ and K (and thus T), to achieve the best possible training performance at the end of training, *i.e.*, the minimum loss function $F(\theta^f)$ computed with the final model parameter θ^f, given resource constraints. This optimization problem is formally presented as follows:

$$\begin{aligned} \min_{\forall \tau, K \in \{1,2,3....\}} \quad & F(\theta^f) \\ \text{s.t.} \quad & (T+1)c_m + (K+1)b_m \leq R_m, \forall m \\ & T = K\tau. \end{aligned} \tag{2}$$

3.3 Model Fairness

Apart from adaptively determining the frequency of local updates and global aggregations, we further consider the flexibility of device participation for global aggregation (determined by a *selector*), based on the following observations.

First, due to limited network resource and unstable connection, not all UEs can always participate in the global aggregation step. *Second*, as data are non-i.i.d. across UEs, aggregating updates from all the UEs is not the best option for model convergence. Therefore, instead of blindly involving all the UEs, we hope to judiciously coordinate the UEs for their participation.

Intelligent selection of UEs can save essential resource such as bandwidth compared to the default setting of all participation. Moreover, since different UEs contribute differently towards training performance, based on their datasets, energy consumption, *etc.*, appropriate selection of UEs, with the intuitive idea of involving important UEs more frequently, has strong promise to achieve better training performance. On the other hand, with flexible participation introduced, there exists risk that a device may never be selected and the learned model is severely biased.

To avoid such situations, we consider the fairness of the learned model, which is defined uniquely for the federated learning setting: *given trained models θ and θ', we can say that θ is more fair than θ', if the loss or accuracy of θ on the N devices $\{f_1, f_2, ..., f_N\}$, is more uniform than that of θ'* [17]. With the consideration of both resource efficiency and model fairness, we will present our solution of dynamic UE selection in the next section.

4 Design of *Eiffel* for Efficient and Fair Scheduling

Now we present our design of *Eiffel* to *efficiently* and *fairly* coordinate mobile devices for their participation in the complete training process of federated learning.

The main idea of *Eiffel* is to select a set of mobile devices, *i.e.*, UEs, in each round, to participate in federated learning. To make judicious decisions on the selection of UEs, we consider the comprehensive factors of the local loss, the data size, the computation power, resource demand and last update time associated with each UE. Accounting for these factors, an overall index will be calculated by *Eiffel*, to indicate the priority of each UE to be selected for participation by the scheduling algorithm.

Particularly, before each global aggregation, the following factors will be considered for the i^{th} UE: the loss value

$f_i(\theta)$ achieved with local model, the size of local data d_i, the computation power c_i, the resource demand r_i and the age of update (AoU) t_i which refers to the last communication round when the UE participated in global aggregation. With the these metrics captured before the global update, the priority index of the i^{th} UE will be calculated as

$$\omega/f_i(\theta) + \varrho d_i + \gamma\frac{c_i}{r_i} + \psi t_i. \quad (3)$$

Here, $\omega, \varrho, \gamma, \psi$ are used to set weight for each of the factors aforementioned to be considered in priority. A higher value of this index indicates a higher priority. Intuitively, a UE with a lower loss $f_i(\theta)$ indicates a more accurate copy of model, and thus should have higher priority to participate in and contribute to the global model aggregation. Similarly, if a UE has more local data (a larger d_i), it should be preferred in our selection to make contribution. The third term, $\frac{c_i}{r_i}$, represents the resource efficiency of the UE, and a higher value makes the UE more competitive to be selected. Finally, the AoU metric t_i helps to prevent a UE from being left isolated for a long time. The four weight parameters will remain constant for all the UEs within one global update but can be flexibly tuned based on the performance of the global model.

The procedures at the aggregator and each UE are presented in Algorithm 1, which is coordinated by *Eiffel* with an essential design of a dynamic selection of UEs for the participation in global updates. The aggregator initiates the learning process by sending the model θ, initialized as a constant or random vector, and the local training step τ, initialized as 1, to all the UEs (`line 6-7`). Accordingly, each UE, upon receiving data from the aggregator, will perform local training iterations for τ steps. Then it reports the local updates and per-step resource usage to the aggregator (`line 30-34`), to be globally aggregated as elaborated next. , When the aggregator receives weights and other parameters from the selected UEs, it will update the global model using weighted average (`line 10-16`, specifically `line 13`). The relative weight, associated with each UE, is proportional to the amount of its local data d_i, the ratio of its computation power to resource demand $\frac{c_i}{r_i}$, and its AoU metric maintained using $L_t(i)$ in the algorithm (with more elaboration to come in the next section). Meanwhile, the local loss at each selected UE will be recorded in $L_l(i)$, the accumulated resource consumption for each type will be updated (`line 15`), and the AoUs of all the UEs will be updated (`line 14,18`).

With all the metrics readily available, the adaptive value of τ for the next round of local updates is calculated following [4]. In addition, a dynamic participant selection comes into action to decide the next set of UEs for reporting their model updates in the next round (`line 21-22`).

Given the per-round resource budget R, a subset of UEs will be selected based on their priority index values in Eq. (3). More specifically, a number of UEs are chosen from $select$, which is the list of UEs selected in the last round, to exhaust a portion (κ percent) of the budget, while the rest of the budget is used to involve UEs from $L - select$, which did not participate in the previous round. The rationale for introducing the proportion parameter κ is to promote contribution from more participants: for example, if the priority ranking of each UE remains the same across two consecutive rounds, this proportional selection gives opportunities to a few promising UEs that were just below the threshold to be selected in the previous round.

After participant selection, the aggregator checks the availability of each type of resource for the next round, based on an estimation. Given the selected UEs in $select$, the updated τ, and the current total accumulated type-m resource usage s_m, it calculates the expected total usage after the next round, based on the historical $c_{m,i}$ and available b_m (`line 23`). If the resource budget R_m is violated for any

Algorithm 1: Procedures at the Aggregator and each user equipment (UE) coordinated with *Eiffel*

Input:
List of all UEs L,
List of computation power of UEs $L_c < c_1, ..., c_N >$,
List of resource demand of UEs $L_r < r_1, ..., r_N >$,
List of data size of UEs $L_d < d_1, ..., d_N >$,
Resource budget $R_m, \forall m$,
Per-round resource budget R.
Initialize:
1: Initialize θ as a constant or random vector
2: List of local loss of UEs $L_l \leftarrow < 0, ..., 0 >$
3: List of AoU of UEs $L_t \leftarrow < 1, ..., 1 >$
4: $select \leftarrow L$, $\tau \leftarrow 1$, $s_m \leftarrow 0, \forall m$
At the Aggregator:
5: **while** True **do**
6: **if** $1st\ round$ **then**
7: Send θ and τ to each UE in L
8: **else**
9: $\theta \leftarrow 0$, $\nabla F(\theta) \leftarrow 0$
10: **for** UE i in $select$ **do**
11: Receive θ_i, $F_i(\theta), \nabla F_i(\theta)$ and $c_{m,i}, \forall m$
12: $\nabla F(\theta)$ += $d_i\alpha_i\nabla F_i(\theta)/D$
13: θ += $d_i\alpha_i\theta_i/D$ `// where` $\alpha_i = \frac{c_i L_t(i)}{r_i}$
14: $L_t(i) \leftarrow 1$, $L_l(i) \leftarrow \nabla F_i(\theta)$
15: s_m += $c_{m,i}\tau + 2b_m$, $\forall m$
16: **end for**
17: **for** UE i in $L - select$ **do**
18: $L_t(i) \leftarrow L_t(i) + 1$
19: **end for**
20: Calculate τ based on [4]
21: $select \leftarrow$ UEs from sorted $select$ to consume κR
22: $select \leftarrow$ UEs from sorted $L - select$ to consume $(1-\kappa)R$ `// sorting based on Eq.(3)`
23: **if** $\exists\ m \mid s_m + \sum_{i \in select}(c_{m,i}\tau + 2b_m) > R_m$ **then**
24: Send `STOP` flag to each UE in L
25: Return θ
26: **end if**
27: Send θ and τ to each UE in $select$
28: **end if**
29: **end while**
At each UE:
30: **while** `STOP` flag not received **do**
31: Receive θ and τ from the Aggregator
32: Perform local iterations for τ steps to update θ_i
33: Send $\theta_i, F_i(\theta), \nabla F_i(\theta), c_{m,i}, \forall m$ to the Aggregator
34: **end while**

type m, the aggregator will stop training at each UE and return the final model parameters (`line 23-26`). Otherwise, it sends model and step updates to each selected participant to start the next round of training.

In summary, the dynamic selection of UEs, in combination with the adjustment of local computation steps, manages to improve the resource efficiency and benefit the model convergence. Moreover, the criteria for selecting UEs accounts for a comprehensive set of factors, contributing to the guarantee of fairness. These features will be formally analyzed in our next section.

5 Analysis of Fairness and Convergence

In this section, we will analyze the behavior of *Eiffel* from both of the convergence and fairness perspectives.

5.1 Convergence Analysis

Our convergence analysis is based on the adaptive federated learning setting which incorporates our scheduling algorithm[2]. The goal is to find the upper bound of:

$$F(\theta^f) - F(\theta^*), \tag{4}$$

where θ^* is the optimal model parameter. As aforementioned, T iterations throughout the training can be divided into K different intervals, with only the first and last iterations in each interval involving global aggregation. We use the shorthand notation $[k]$ to denote the iteration interval $[(k-1)\tau, k\tau]$, for $k = 1, 2, ..., K$. The global loss function on all distributed datasets can be expressed as:

$$F(\theta) \triangleq \frac{\sum_{i=1}^{P} D_i \alpha_i F_i(\theta)}{D}, \tag{5}$$

where $\alpha_i = \frac{c_i t_i}{r_i}$, impacting how likely the local model of the i^{th} UE will be selected for the contribution to the global model update, and P is the number of devices selected from a total of N devices on each global iteration. The device selection probability is proportional to $\frac{D_i c_i t_i}{r_i}$, thus a local device which possesses more training data, higher computational power, has longer waiting time to send parameters and has less resource demand will get a higher probability of selection.

In contrast, prior works define $F(\theta) = \frac{\sum_{i=1}^{P} D_i F_i(\theta)}{D}$ to select a subset of UEs with probabilities $\frac{D_i}{D}$ at each round. Considering only data size leads to unfairness among the participating devices. *Eiffel*'s selection algorithm provides a good balance of efficiency and fairness.

Our analysis is conducted based on the assumptions that for all i, 1) $F_i(\theta)$ is convex, 2) $F_i(\theta)$ is $\rho - Lipschitz$, that is for any θ and θ', $\|F_i(\theta) - F_i(\theta')\| \leq \rho\|\theta - \theta'\|$, and 3) $F_i(\theta)$ is $\beta - smooth$, *i.e.*, for any θ and θ', $\|\nabla F_i(\theta) - \nabla F_i(\theta')\| \leq \beta\|\theta - \theta'\|$. The learning problem here is to minimize $F(\theta)$, *i.e.*, to find the optimal model parameter θ^* such that:

$$\theta^* \triangleq arg\ min\ F(\theta). \tag{6}$$

After the global iteration, for node i, the update is

$$\theta_i(t) = \tilde{\theta}_i(t-1) - \eta \nabla F_i \alpha_i (\tilde{\theta}_i(t-1)), \tag{7}$$

2. The convergence analysis is only for convex models, similar to the literature. Non-convex models are considered in our experimental evaluation.

where $\tilde{\theta}_i(t)$ denotes the parameter after previous global aggregation. η is the step size. For any iteration t which may or may not be a global aggregation step, we have

$$\theta(t) = \frac{\sum_{i=1}^{P} D_i \alpha_i \theta_i(t)}{D}. \tag{8}$$

For each local iteration interval $[k]$, we use $v_{[k]}(t)$ to denote an auxiliary parameter vector that follows a centralized gradient descent according to

$$v_{[k]}(t) = v_{[k]}(t-1) - \eta \alpha \nabla F(v_{[k]}(t-1)), \tag{9}$$

where $v_{[k]}(t)$ is defined for interval $t \in [(k-1)\tau, k\tau]$ for a given k. At the beginning of each interval $[k]$, $v_{[k]}((k-1)\tau) \triangleq \theta((k-1)\tau)$, where $\theta(t)$ is the average of local parameters defined in Eq.(8). For the analysis, gradient divergence is further defined below which is the difference between the gradient of local loss function and the gradient of global loss function.

Definition 1. (Gradient Divergence) For any i and θ, we define δ_i as an upper bound of $\|\nabla F_i(\theta) - \nabla F(\theta)\|$, *i.e.*,

$$\begin{aligned} &\|\nabla F_i(\theta) - \nabla F(\theta)\| < \alpha_i \delta_i \\ &also \quad \delta \triangleq \frac{\sum_{i=1} D_i \alpha_i \delta_i}{D} \end{aligned} \tag{10}$$

This definition is related to the data distribution and accounts for the metrics we use for scheduling defined in the previous section. Upper bound of Eq. (4) can be derived by adopting the adaptive setting of [4] in two steps:

- The first step is to find the difference between the distributed ($\theta(t)$) and centralized gradient descents ($v(t)$) after each τ steps of local update without global aggregation.
  For any interval $[k]$ and $t \in [k]$, the upper bound of difference between $\theta(t)$ and $v_{[k]}(t)$ is derived as

  $$\|\theta(t) - v_{[k]}(t)\| \leq h(t - (k-1)\tau) \tag{11}$$

  where $h(x) \triangleq \frac{\delta\alpha}{\beta}((\eta\beta + 1)^x - 1) - \eta\delta x,\ \forall x \in \{0, 1, 2, ...\}$.
- The second step is to combine the aforementioned gap with the convergence bound of $v(t)$ within each interval $[k]$ to obtain the convergence bound of $\theta(t)$ which is essentially deriving the upper bound of $F(\theta(T)) - F(\theta^*)$.

  $$F(\theta(T)) - F(\theta^*) \leq \frac{1}{T(\phi\alpha\eta(1 - \frac{\eta\beta\alpha}{2}) - \frac{\rho h(\tau)}{\tau\epsilon^2})} \tag{12}$$

  where $\phi = \frac{1}{\|v_{[k]}(t) - \theta^*\|^2}$ and ϵ represents the lower bound of $F(\theta(T)) - F(\theta^*)$, β, ρ are Smoothness and Lipschitz parameters. From Eq. (12) we can say that the impact of α is that the increment of computation power and age of update and decrement of resource demand will lead to a faster convergence as the right hand side of Eq. (12) will be smaller.

5.2 Quantification of Fairness

Our analysis of fairness is based on two metrics: the variance of performance across devices and the Cosine similarity between the performance distribution and **1** [17]. For the ease of mathematical exposition, we restate the objective function of federated learning as follows:

$$\min_{\theta} F_q(\theta) = \frac{\sum_{i=1}^{N} D_i \alpha_i^q f_{i,q}(\theta)}{D}; \quad q = 0 \; or \; 1$$

Here α denotes the contribution of each selected device on the global model update according to our selection algorithm. $q = 0$ corresponds to the general objective function for federated learning without the selection algorithm.

The variance of performance distribution: The performance distribution on m devices $X = \{f_{1,q}(\theta), ..., f_{m,q}(\theta)\}$ is defined to be *more uniform* under solution θ than θ', if the variance of X under solution θ is *less than* that under θ', that is:

$$Var(f_{1,q}(\theta), ..., f_{m,q}(\theta)) < Var(f_{1,q}(\theta'), ..., f_{m,q}(\theta'))$$

Our proposed algorithm selects a set of devices in each round for global aggregation. Since the selected devices include κ percent of the previously unselected devices, as in Algorithm 1, it ensures that the variance among performance of all the N devices will be smaller. In a more formal manner, suppose θ is the optimal solution of our problem $min_\theta \; F_1(\theta)$, and θ' is the solution to the conventional problem $min_\theta \; F_0(\theta)$ which does not enable selection, we can easily verify that our solution θ leads to a more uniform performance distribution than θ':

$$\begin{aligned} & \frac{\sum_{i=1}^{N} f_{i,1}^2(\theta)}{N} - \left(\frac{1}{N}\sum_{i=1}^{N} f_{i,1}(\theta)\right)^2 \\ \leq \; & \frac{\sum_{i=1}^{N} f_{i,1}^2(\theta')}{N} - \left(\frac{1}{N}\sum_{i=1}^{N} f_{i,1}(\theta)\right)^2 \\ \leq \; & \frac{\sum_{i=1}^{N} f_{i,1}^2(\theta')}{N} - \left(\frac{1}{N}\sum_{i=1}^{N} f_{i,0}(\theta')\right)^2 \end{aligned}$$

Cosine similarity between the performance distribution and 1:
The performance distribution on m devices $X = \{f_{1,q}(\theta), ..., f_{m,q}(\theta)\}$ is defined to be *more uniform* under solution θ than θ', when the cosine similarity of X and **1** under solution θ is *greater than* that under θ', which is:

$$\frac{\frac{1}{N}\sum_{i=1}^{N} f_{i,q}(\theta)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} f_{i,q}^2(\theta)}} \geq \frac{\frac{1}{N}\sum_{i=1}^{N} f_{i,q}(\theta')}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} f_{i,q}^2(\theta')}}$$

Following a similar setting and analysis as the previous metric, we have the following inequalities for our solution θ and other θ': $\frac{1}{N}\sum_{i=1}^{N} f_{i,0}(\theta) \geq \frac{1}{N}\sum_{i=1}^{N} f_{i,0}(\theta')$ and $\frac{1}{N}\sum_{i=1}^{N} f_{i,0}^2(\theta) \geq \frac{1}{N}\sum_{i=1}^{N} f_{i,0}^2(\theta')$. Therefore, we can derive the following expression, omitting the steps similar as in [17] due to space limit:

$$\frac{\frac{1}{N}\sum_{i=1}^{N} f_{i,0}(\theta)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} f_{i,0}^2(\theta)}} \geq \frac{\frac{1}{N}\sum_{i=1}^{N} f_{i,0}(\theta')}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} f_{i,0}^2(\theta')}}$$

Again, we have demonstrated that our solution leads to a more uniform performance distribution, when measured using the cosine similarity metric.

6 Experimental Analysis

In this section, we evaluate *Eiffel* with extensive experiments, from the perspectives of machine learning models' performance (accuracy and loss), efficiency (communication improvement) and fairness (variance of loss distribution across devices). All these metrics collectively demonstrate the superiority of *Eiffel* in achieving fairness as well as efficiency. We show that *Eiffel*'s device selection barely hampers the model performance while reducing communication overhead.

6.1 Experimental Setup

Our experiments were conducted in both a prototype system and an emulated large-scale environment. To conduct the experiments in a resource-constrained heterogeneous environment, we use devices of different memory and computation power, as shown in Table 2. Two different laptops and a mobile phone were employed as the user devices (UEs), and a cloud server, *i.e.*, a small AWS instance, was used as the aggregator in the prototype federated learning system. All the UEs have local datasets on which model training was conducted. We also conducted larger-scale simulation experiments on an AWS `c5a.4xlarge` instance with 16 vCPUs, where we emulated a federated learning environment with hundreds of UEs participating. We took careful measures when instantiating each of the participating UE processes to mitigate the gap between our emulation and reality. Models were trained with emulated UEs where the resource budget (in terms of computation time) was generated according to a Gaussian distribution with mean and standard deviation values derived from our prototype measurements. The computational power of each UE follows commonly used devices such as laptops, mobile phones, Raspberry Pi, and *etc.*, consistent with the real-world heterogeneous environment. Each UE and the aggregator communicates with each other using Socket.

Models and Datasets. We use both convex and non-convex models to evaluate our proposed algorithm. One of them is the popular binary classifier, *squared*-SVM (to be mentioned as SVM for simplicity), and the other one is a convolutional neural network (CNN). For the SVM model, we feed the publicly available large-scale MNIST [35] handwritten digit dataset for model training. It contains gray-scale images of 70,000 handwritten digits (60,000 for training and 10,000 for testing). As a binary classifier, the SVM model will classify a digit as either *odd* or *even* for the MNIST dataset. The CNN model used in our experiments follows a standard structure with 9 layers in total, including two 5x5x32 convolutional layers, two 2x2 max-pool layers, two local response normalization layers, two fully-connected layers and one softmax classification layer. In addition to the MNIST dataset aforementioned, two large-scale image datasets, Fashion-MNIST and CIFAR-10 [36], are also used for CNN model training. More specifically, Fashion-MNIST has the same format as MNIST but includes 28x28 grayscale

Devices and Servers	Configuration	
UE1 (Laptop)	CPU	Quad-Core Intel (Core i5@1.4 GHz)
	Memory	8 GB
	System	macOS Catalina
UE2 (Laptop)	CPU	Intel Core i5@2.3 GHz
	Memory	8 GB
	System	macOS Sierra
UE3 (Mobile)	CPU	Exynos 7904,Octa-Core (2@1.6 GHz, 6@1.35 GHz)
	Memory	4GB RAM
	System	Android Pie
Aggregator (Cloud Server)	CPU	Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
	Memory	1GB
	System	Ubuntu 18.04.5 LTS
Emulation (Cloud Server)	CPU	c5a.4xlarge, 16-vCPU AMD EPYC 7R32 @3.3GHz
	Memory	32 GB
	System	Ubuntu 18.04.5 LTS

TABLE 2: Experimental setup: prototype setting and emulation environment.

images of fashion items instead of digits. It consists of 70000 images, categorized into 10 classes, with 60,000 used as a training set and the rest for a test set. CIFAR-10 [36] consists of 60,000 32x32 color images. We have also conducted our experiments with two types of data distribution (i.e., i.i.d. and non-i.i.d.) among the UEs. More specifically, for the i.i.d. setting, each data sample is assigned randomly to a UE. For the non-i.i.d. setting, each UE consists of data with the same label. If there are more labels than UEs, each UE will have data with more than one label, but the number of labels at each UE is no more than the total number of labels divided by the total number of UEs rounded to the next integer. In the training phase, the learning step hyperparameter is set to 0.01 and the batch size is 100. We launch our model training in the popular machine learning framework, Tensorflow [37], with stochastic gradient descent as the optimizer. Other hyperparameters such as control parameters for different models and the maximum value of τ are kept the same with [4] for the adopted control algorithm to select the optimal value of τ accordingly.

Evaluation Metrics. Under both i.i.d. and non-i.i.d data distribution settings, we evaluate our solution with respect to the accuracy, efficiency and fairness. The accuracy metrics include the loss value and accuracy level of the trained model with respect to different τ values (*i.e.*, the local iteration number). The efficiency is measured by the communication frequency for global aggregation multiplied by the number of UEs selected and the time for exchanging parameters between the aggregator and UEs. Our fairness evaluation is based on two commonly used metrics in the federated learning setting: *variance* and *skewness* [16], [17]. The variance of loss distribution among the UEs indicates how the final global model is biased among a group of UEs, while the skewness metric implies how much the loss distribution is deviated from the normal distribution and how symmetric it is across the UEs.

Baselines. We compare *Eiffel* with the following baselines to evaluate their model accuracy performance and communication efficiency:

- Canonical federated learning, where every participating UE contributes to the model training using a fixed (non-adaptive) τ value.
- Adaptive federated learning strategy in [4] which adopts the optimal τ value given the resource constraints.
- Centralized gradient descent version of model training where the whole training is carried out on a single UE.

With respect to fairness, we choose three additional baselines:

- q-FFL [17], the state-of-the-art fairness-oriented federated learning, with the q value set as 1.5.
- Random selection (RS), which selects UEs to contribute to global model aggregation in a random manner.
- Least training loss based selection (LLS), which selects the devices that generate the smallest local training loss in each round.

All the baselines select the same number of UEs as *Eiffel* in each experiment.

Model *Dataset*	Alg.	Avg. loss (%)	Worst 10%	Best 10%	Var.	Skew.
SVM *MNIST*	*Eiffel*	22.3	43	5.5	0.0101	0.067
	LLS	21.4	44	5.0	0.0216	1.806
	RS	23.2	45	5.4	0.1240	2.105
	q-FFL	23.5	42.8	5.1	0.0159	1.055
CNN *CIFAR-10*	*Eiffel*	29.2	35.4	7.0	0.0120	0.036
	LLS	30.4	45.7	9.0	0.1640	0.734
	RS	28.1	43.8	7.0	0.0192	0.167
	q-FFL	25.2	34.4	7.4	0.0179	0.0673

TABLE 3: Statistics of model loss distribution for SVM trained on *MNIST* and CNN on *CIFAR-10*, achieved by *Eiffel*, LLS, RS, and q-FFL, respectively.

6.2 Results and Analysis

6.2.1 Fairness

We have conducted two sets of experiments to evaluate the fairness achieved by *Eiffel*, in comparison with the three baselines aforementioned: LLS (Least training loss based selection), RS (Random Selection), and q-FFL [17]. Table 3 presents the results for SVM and CNN models, trained on MNIST and CIFAR-10 datasets, respectively, involving 100 emulated UEs with different computation power, time budget and non-i.i.d data distribution. In particular, we show the average loss of all the UEs (column 3), the average loss of the worst 10% UEs (column 4), the average loss of the best 10% UEs (column 5), the variance of the loss across all the UEs (column 6), and the skewness (column 7).

As observed in Table 3, *Eiffel* achieves the minimum variance among all the comparing baselines, for both SVM and CNN models. Compared with q-FFL, the state-of-the-art fairness baseline, *Eiffel* reduces the variance by 57.8% and 59% for the two models, respectively. Moreover, with respect to the skewness metric, the advantages of *Eiffel* over q-FFL and the other two baselines are more obvious. For CNN model, *Eiffel* achieves a 86.94% smaller skewness of loss distribution than q-FFL, and for SVM the skewness is reduced by more than 100%. Both of the variance and skewness metrics demonstrate the advantages of *Eiffel* over all

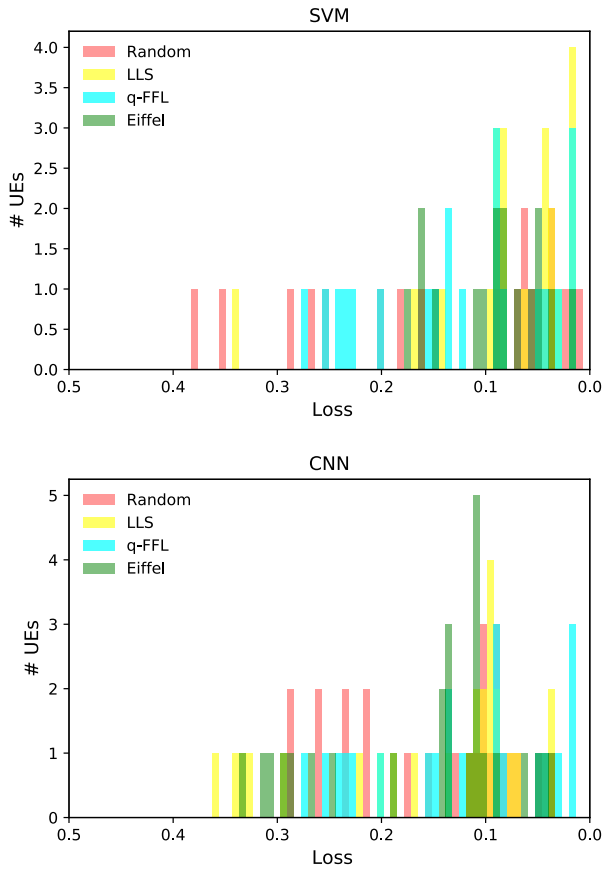


Fig. 2: Distribution of loss values across user devices for CNN and SVM models, respectively, achieved by *Eiffel* and the three comparing baselines.

the baselines in achieving fairness of model loss distribution across UEs.

In a more intuitive manner, Fig. 2 illustrates the distribution of loss values across 20 user devices in our prototype experiment for the two models and four algorithms. More specifically, the x-axis represents the loss values of a learned global model at local devices, and the y-axis implies how many devices have the similar loss values, falling to the same loss range. For example, for SVM, considering the range of 0.3 to 0.4, there are two devices in the Random baseline whose loss values fall into this range, and one device in the LLS setting in this range. As observed, with *Eiffel*, 60% of the UEs have their loss values of the SVM model in the range of 0.17 to 0.085, while for the CNN model, 65% UEs have loss values centered around the range of 0.08 to 0.1. In comparison, with the other three algorithms, we hardly identify the similar range of loss value where there are more than 40% devices, except for LLS on SVM model. Although 50% UEs' losses for SVM with LLS are centered around 0.01 to 0.1, the variance of the overall loss distribution is higher than all the other algorithms.

6.2.2 Efficiency

Next we show the communication efficiency of *Eiffel* compared to the adaptive federated learning baseline [4]. Table 4 presents the average number (τ) of local iterations across rounds achieved by the baseline (Column 3) and *Eiffel* (Column 4), under two settings of UEs (corresponding to two values in each cell of the table), for SVM and CNN models with i.i.d and non-i.i.d data distributions, respectively.

Data Distribution	Model	Avg. τ of [4]	Avg. τ of *Eiffel*	Comm. Improvement
i.i.d.	SVM	29.65	35.23	2.06
		87.31	94.45	3.05
	CNN	80.28	100.50	5.77
		70.28	90.50	6.45
non-i.i.d.	SVM	4.31	7.89	2.44
		6.33	8.74	1.57
	CNN	101.52	179.75	6.19
		89.65	120.79	4.98

TABLE 4: The frequency of global aggregation, indicated by the average number (τ) of local iterations across rounds, achieved by *Eiffel* and the adaptive federated learning baseline [4], and the communication improvement of *Eiffel* over [4], under different settings.

As observed, the average τ value with *Eiffel* is consistently larger than the baseline across different settings. A larger τ indicates more local iterations on each UE between two consecutive global aggregations, which leads to a smaller communication frequency. This brings significant advantages in the resource-constrained environment where the overall communication network bandwidth is limited. In order to quantify the communication overhead, we use an intuitive measure: the multiplication of the total number of participating UEs, the total number of global aggregations, and the communication time for parameter exchange between the aggregator and participating UEs. We calculate the ratio of the baseline's communication overhead to *Eiffel*, shown as the communication improvement of *Eiffel* over [4] in Table 4 (Column 5). Results have demonstrated the communication efficiency improvement of *Eiffel* in comparison with the baseline. In addition to a relatively small increase of the τ value, the smaller number of participating UEs and the reduced communication time lead to the large improvement of communication efficiency.

6.2.3 Accuracy

As mentioned, we have also compared our strategy with the baselines in terms of the model loss and accuracy. Fig. 3 presents the loss and accuracy values of the SVM model, learned by 60 UEs in our prototype, with different strategies, *i.e.*, *Eiffel* (represented by the single blue dot), the adaptive federated learning baseline [4] (represented by the single red dot) and the centralized learning baseline (represented by the green line), respectively. In particular, *Eiffel* and [4] have adaptive τ values across rounds. Their average τ values and their model loss or accuracy values are used to position their corresponding dots in Fig. 3. We also investigate the variants of *Eiffel* and [4] with fixed global update frequency (fixed τ) across rounds, and illustrate how they perform given different values of τ (11 values ranging from 1 to 100), as represented by the blue dotted line and the red dotted line. The centralized approach does not depend on τ and thus be represented as a flat line. In a similar vein, Fig. 4 and Fig. 5 present the performance of the CNN model learned with 20 UEs on CIFAR-10 and Fashion-MNIST, respectively. The experiments are conducted with the identical resource budget setting for all the baselines.

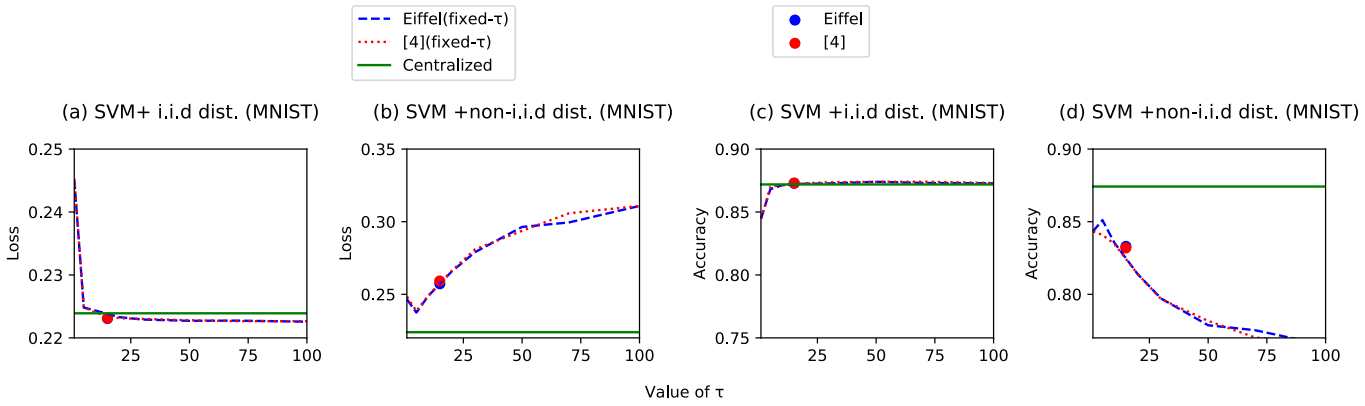


Fig. 3: Loss function values and classification accuracy with different τ achieved by Eiffel, adaptive federated learning baseline [4] and centralized baseline for SVM model trained on *MNIST* under different settings. The average τ value of *Eiffel*, baseline [4] and their corresponding loss, accuracy values are represented by single markers.

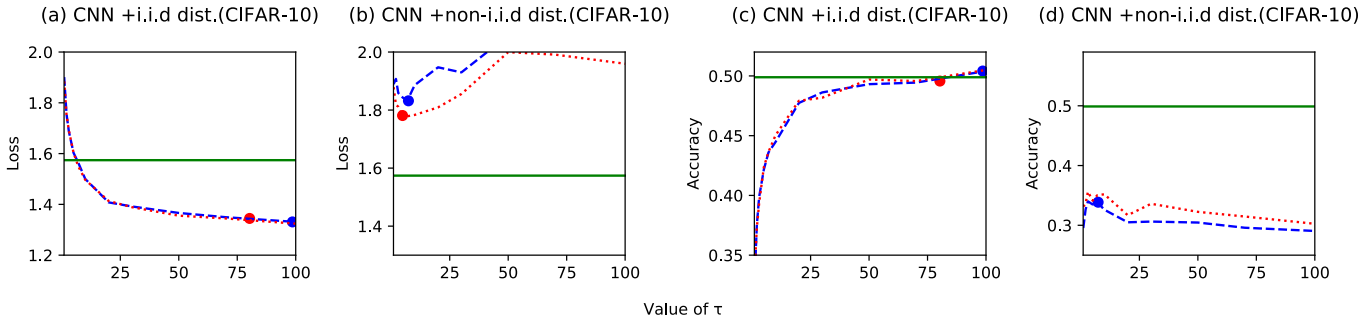


Fig. 4: Loss function values and classification accuracy with different τ achieved by Eiffel, adaptive federated learning baseline [4] and centralized baseline for CNN model trained on *CIFAR-10* under different settings. The average τ value of *Eiffel*, baseline [4] and their corresponding loss, accuracy values are represented by single markers.

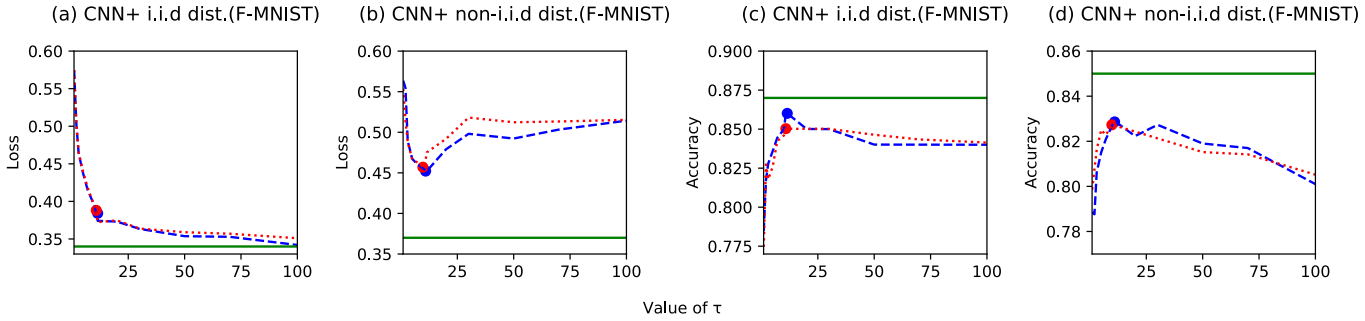


Fig. 5: Loss function values and classification accuracy with different τ achieved by *Eiffel*, adaptive federated learning baseline [4] and centralized baseline for CNN model trained on *F-MNIST* under different settings. The average τ value of *Eiffel*, baseline [4] and their corresponding loss, accuracy values are represented by single markers.

The experimental results shown in Fig. 3-Fig. 5 consist of both settings of i.i.d. and non-i.i.d. data distribution across the UEs. For the i.i.d. distribution, the performance of the variants of *Eiffel* and the baseline [4] (*i.e.*, the fixed-τ setting), represented by the blue dotted lines and red dotted lines in Fig. 3a,c, Fig. 4a,c and Fig. 5a,c, exhibits much similarity with the increasing τ value. With the adaptive setting, the average τ values of *Eiffel* for these cases are also aligned with or larger than the average τ of the adaptive federated learning baseline [4], as illustrated by the x-coordinates of the singe blue and red dots in the corresponding figures. A larger τ value resulted by *Eiffel* (Fig. 4a,c) indicates a lower global aggregation frequency which leads to the reduction of the communication overhead. With the non-i.i.d. distribution setting, the performance results exhibit a bit more diversity for *Eiffel* and the baseline [4]. For CNN on CIFAR-10, a more complex model on a larger dataset compared to SVM on MNIST and CNN on Fashion-MNIST, *Eiffel* sacrifices the model performance by less than 5% when compared to the baseline [4], as illustrated in Fig. 4b,d. Such a performance tradeoff is anticipated, since *Eiffel* has a smaller number of UEs selected in participation and ensures fairness among all the participants as aforementioned. In general, even though with fewer participating UEs, *Eiffel* manages to achieve

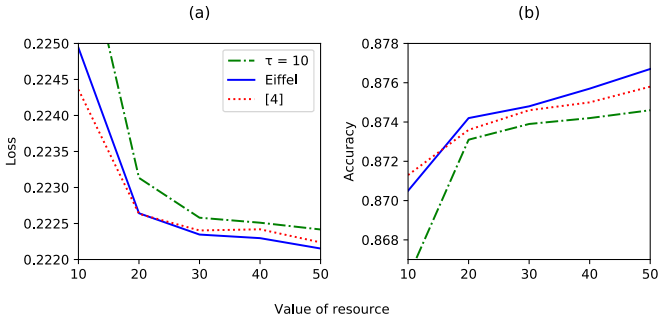


Fig. 6: Effect of different resource budget on model performance.

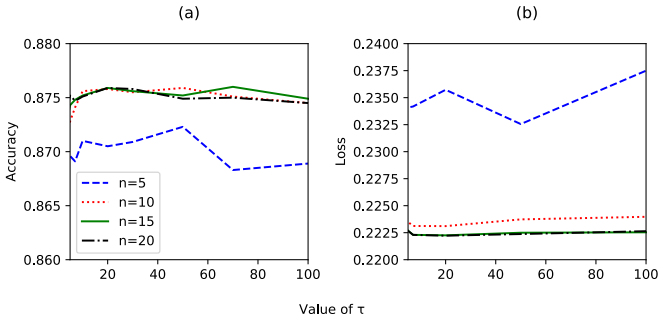


Fig. 7: Effect of different number (n) of selected participants among 20 UEs on model performance.

Model *Dataset*	Data Distribution	Adaptive Setting (τ value)	Loss Diff. of *Eiffel vs. Centralized* (%)	Loss Diff. of [4] *vs. Centralized* (%)
SVM *MNIST*	i.i.d	Avg-τ	(-) 0.0829	(-) 0.077
		1	(+) 2.109	(+) 2.137
		30	(-) 0.099	(-) 0.089
		70	(-) 0.116	(-) 0.124
	non-i.i.d.	Avg-τ	(+) 3.43	(+) 3.63
		1	(+) 2.366	(+) 2.54
		30	(+) 5.61	(+) 5.81
		70	(+) 7.66	(+) 82.9
CNN *CIFAR-10*	i.i.d.	Avg-τ	(-) 6.2	(-) 6.2
		1	(-) 55.62	(-) 53.95
		30	(-) 4.6	(-) 4.3
		70	(+) 0.045	(-) 0.03
	non-i.i.d.	Avg-τ	(+) 25.7	(+) 20.7
		1	(+) 31.52	(+) 30.47
		30	(+) 35.57	(+) 28.13
		70	(+) 34.53	(+) 33.70
CNN *F-MNIST*	i.i.d.	Avg-τ	(+) 4.41	(+) 4.81
		1	(+) 23.46	(+) 23.46
		30	(+) 2.343	(+) 2.383
		70	(+) 1.386	(+) 1.701
	non-i.i.d.	Avg-τ	(+) 8.21	(+) 8.694
		1	(+) 19.3	(+) 18.35
		30	(+) 12.809	(+) 14.809
		70	(+) 13.328	(+) 14.328

Avg-τ : average value of τ in the adaptive federated learning

TABLE 5: Comparison on loss value differences of *Eiffel vs.* centralized baseline and adaptive learning baseline [4] *vs.* centralized baseline, for SVM trained on *MNIST*, CNN trained on *CIFAR-10*, and CNN on *F-MNIST*, respectively, under different settings.

comparable model performance with the adaptive federated learning baseline [4], as demonstrated by the y-coordinates of blue and red single dots in Fig. 3-Fig. 5.

For the Centralized baseline, represented by the green line, the training datasets associated with each UE are residing in the centralized server, where the model is trained on the complete set of training data using centralized gradient descent. Intuitively, if given a fixed number of iterations or rounds, the Centralized baseline will result in better model performance, with lower loss and higher accuracy, than the decentralized approach. However, our experimental setting has a fixed resource budget (fixed computation time), thus the centralized approach does not necessarily beat the decentralized one. For example, as shown in Fig. 4a, federated learning can benefit from utilizing the computation capabilities of UEs in parallel, while the centralized approach relies on the computation resource of a single server which prolongs the iteration time, especially when the model is complex and the dataset is large.

Table 5 further provides a comprehensive presentation on the model performance of *Eiffel* and [4] under different settings as aforementioned, in comparison with the Centralized baseline. In particular, we show the difference of the loss values between *Eiffel* and the Centralized baseline (Column 4), and that between [4] and the Centralized baseline (Column 5), under the adaptive federated learning setting and the three fixed-τ settings (with values of 1, 30 and 70, respectively), for both i.i.d. and non-i.i.d. data distributions. The positive value indicates how far the loss increases from the Centralized baseline, while the negative value implies that the decentralized federated learning approach gets better model performance, in terms of lower loss, than the centralized one.

6.2.4 *Varying Resource Budget and Participation Ratio*

We have conducted another set of experiments in the prototype system, to show the impact of different parameters on model performance, including the resource budget and the number of selected participants. Fig. 6 shows the effect of different resource budget on the loss and accuracy performance achieved by the SVM model, when trained with the strategies of *Eiffel*, adaptive baseline [4], and the fixed τ setting ($\tau = 10$, the green dashed line), respectively. Intuitively, the model performance improves with the increasing resource budget for each comparing strategy. Adaptive τ setting is always more efficient than fixed τ setting with increasing resource budget. For *Eiffel* and [4], with the increment of resource budget, their τ values become close to 1. Therefore, with the increased frequency of global aggregation, the performance gets better. We can also observe that, initially, due to very small resource budget, *Eiffel* starts with selecting a small number of UEs which results in a lower performance compared to [4]. With increasing resource budget, it attains better performance compared to [4], because of its effective scheduling strategy that improves the learning efficiency. With respect to the impact of device selection ratio, we have conducted the experiments of learning the SVM model with *Eiffel* under the settings of selecting 5, 10, 15 and 20 out of 20 devices, given different τ values, respectively. As shown in Fig. 7, there exists an obvious performance gap between selecting 5 devices and the other three settings. The small number of participants leads to the

performance degradation, intuitively. On the other hand, when we select 10, 15 and all of the 20 devices, consecutively, the performance remains almost identical with the varying τ value. Thus we can afford to select only 10 or 15 out of 20 UEs to get the similar model performance while saving resource to a large extent.

7 Discussion and Future Direction

We have demonstrated the communication and computation efficiency as well as model fairness achieved by *Eiffel* in our prototype system and emulated large-scale environment. The crucial issues studied and challenges addressed by *Eiffel* generally exist in any real-world federated learning application. Thus, the solution proposed in *Eiffel* can be taken advantage of by a broad range of real-life federated learning applications deployed in today's Internet-of-Things infrastructure, with some additional privacy enhancement (*e.g.*, preventing the indirect privacy leakage) according to particular requirement.

Eiffel coordinates the learning based on a number of factors associated with participating UEs and the network, *i.e*, the memory, computation power, bandwidth and communication time. Further mathematical and algorithmic analysis could be explored to understand the impact of each term on achieving efficiency and fairness. We would like to conduct further investigation including more heterogeneity of resources among the UEs, *i.e.* mobility patterns of mobile devices, and better management of stragglers among UEs, including case study based on different real-life scenarios such as unexpected system failures, extremely constrained resource availability, *etc.* We also consider to incorporate the flexibility of adding different fairness preference in our algorithm, and conduct more large-scale real-world experiments to further evaluate *Eiffel*.

8 Conclusion

In this paper we propose *Eiffel*, which is an efficient and fair scheduling algorithm for large-scale federated learning in resource-constrained environments. Our algorithm saves the communication throughout the process of federated learning by selecting a subset of devices to achieve the best resource efficiency. The algorithm is also designed to ensure model fairness, which is defined with respect to the model performance distribution across the devices. We have analyzed the performance of *Eiffel*, including the fairness analysis and the convergence bound derivation. Furthermore, we have conducted experiments in a variety of settings, learning simple and complex models on publicly available large-scale image datasets for both i.i.d and non i.i.d data distributions in the real-world and simulated environment. Results demonstrate that *Eiffel* outperforms the state-of-the-art in terms of model fairness and communication efficiency, while achieving similar model performance.

References

[1] C. Systems, "Cisco Annual Internet Report 2018-2023 White Paper," 2020.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[3] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. (2018) Federated Learning with Non-IID Data. [Online]. Available: arXiv preprint arXiv:1806.00582

[4] S. Wang, T. Tuor, T. Salonidis, 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.

[5] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proc. the ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[6] 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.

[7] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning," in *Proc. IEEE International Conference on Computer Communications*, 2018.

[8] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," in *Proc. NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[9] H. B. McMahan, E. Moore, D. Ramage, and B. Agüera y Arcas. (2016) Federated Learning of Deep Networks using Model Averaging. [Online]. Available: arXiv preprint arXiv:1602.05629

[10] J. Konecný, H. B. McMahan, and D. Ramage. (2015) Federated Optimization:Distributed Optimization Beyond the Datacenter. [Online]. Available: arXiv preprint arXiv:1511.03575

[11] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," in *Proc. IEEE International Conference on Communications*, 2019, pp. 1–7.

[12] M. M. Amiri, G. Deniz, S. R. Kulkarni, and H. V. Poor, "Update Aware Device Scheduling for Federated Learning at the Wireless Edge," in *Proc. IEEE International Symposium on Information*, 2020, pp. 2598–2603.

[13] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. V. Poor, "AGE-BASED SCHEDULING POLICY FOR FEDERATED LEARNING IN MOBILE EDGE NETWORKS," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 8743–8747.

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

[15] A. F. Aji and K. Heafield, "Sparse Communication for Distributed Gradient Descent," in *Proc. Empirical Methods in Natural Language Processing*, 2017, pp. 440–445.

[16] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic Federated Learning," in *Proc. International Conference on Machine Learning*, 2019, pp. 4615–4625.

[17] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair Resource Allocation in Federated Learning," in *Proc. International Conference on Learning Representations*, 2020.

[18] H. Brendan McMahan and D. Ramage, "Federated Learning: Collaborative Machine Learning without Centralized Training Data," *Google Research Blog*, vol. 3, 2017.

[19] C. Hardy, E. Le Merrer, and B. Sericola, "Distributed Deep Learning on Edge-devices: Feasibility via Adaptive Compression," in *Proc. IEEE International Symposium on Network Computing and Applications*, 2017, pp. 1–8.

[20] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient Federated Learning via Guided Participant Selection," in *Proc. USENIX Symposium on Operating Systems Design and Implementation*, 2021, pp. 19–35.

[21] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling Policies for Federated Learning in Wireless Networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, 2020.

[22] Y. J. Cho, J. Wang, and G. Joshi. (2020) Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies. [Online]. Available: arXiv preprint arXiv:2010.01243

[23] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for

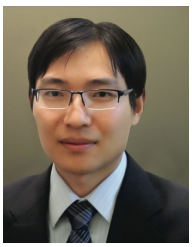
Federated Learning," in *Proc. International Conference on Machine Learning*, 2020, pp. 5132–5143.
[24] F. Haddadpour and M. Mahdavi. (2019) On the Convergence of Local Descent Methods in Federated Learning. [Online]. Available: arXiv preprint: arXiv 1910.14425
[25] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. (2020) On the Convergence of FedAvg on Non-IID Data. [Online]. Available: arXiv preprint: arXiv 1907.02189
[26] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. Jarvis, "SAFA: a Semi-Asynchronous Protocol for Fast Federated Learning with Low Overhead," *IEEE Transactions on Computers*, vol. 70, no. 5, pp. 655–668, 2020.
[27] G. Damaskinos, R. Guerraoui, A.-M. Kermarrec, V. Nitu, R. Patra, and F. Taiani, "Fleet: Online Federated Learning via Staleness Awareness and Performance Prediction," in *Proc. International Middleware Conference*, 2020, pp. 163–177.
[28] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment," in *Proc. International Conference on World Wide Web*, 2017, pp. 1171–1180.
[29] W. Huang, T. Li, D. Wang, S. Du, and J. Zhang. (2020) Fairness and Accuracy in Federated Learning. [Online]. Available: arXiv preprint:arXiv 2012.10069
[30] L. Lyu, X. Xu, Q. Wang, and H. Yu, "Collaborative Fairness in Federated Learning," in *Federated Learning*. Springer, 2020, pp. 189–204.
[31] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A Fairness-aware Incentive Scheme for Federated Learning," in *Proc. the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 393–399.
[32] J. Zhang, C. Li, A. Robles-Kelly, and M. Kankanhalli. (2020) Hierarchically Fair Federated Learning. [Online]. Available: arXiv preprint arXiv:2004.10386
[33] M. H. Rehman, A. Dirir, K. Salah, and D. Svetinovic, "FairFed: Cross-Device Fair Federated Learning," in *Proc. IEEE Applied Imagery Pattern Recognition Workshop*, 2020, pp. 1–7.
[34] C. Yang, Q. Wang, M. Xu, Z. Chen, K. Bian, Y. Liu, and X. Liu, "Characterizing Impacts of Heterogeneity in Federated Learning upon Large-Scale Smartphone Data."
[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[36] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," *Technical report, University of Toronto*, 2009.
[37] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "TensorFlow: A System for Large-Scale Machine Learning," in *Proc. USENIX symposium on operating systems design and implementation*, 2016, pp. 265–283.

Abeda Sultana received her BSc degree from Department of Computer Science & Engineering, University of Dhaka, Bangladesh in 2018. She is a third year PhD student of Department of Computer Science, School of Computing and Informatics, University of Louisiana at Lafayette. Her research interest includes distributed machine learning, scheduling of distributed system, resource allocation, and federated learning.

Md Mainul Haque received his BSc degree from Department of Computer Science & Engineering, University of Dhaka, Bangladesh, in 2015. He is a third year PhD student of Department of Computer Science, School of Computing and Informatics, University of Louisiana at Lafayette. His research interests include distributed machine learning, deep learning, scheduling.

Li Chen received the BEng degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2012, and the MASc and PhD degrees from the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada, in January 2015 and July 2018, respectively. She is currently an assistant professor with the Department of Computer Science, School of Computing and Informatics, University of Louisiana at Lafayette. Her research interests include big data analytics, machine learning systems, cloud computing, datacenter networking, resource allocation, and scheduling in networked systems.

Fei Xu received the PhD degree in computer science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2014. He received Outstanding Doctoral Dissertation Award in Hubei province, China, and ACM Wuhan & Hubei Computer Society Doctoral Dissertation Award in 2015. He is currently an associate professor with the School of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include cloud computing and datacenter, virtualization technology, and distributed systems.



Xu Yuan (Senior Member, IEEE) received the B.S. degree from the College of Information Technology, Nankai University, Tianjin, China, in 2009, and the Ph.D. degree from the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, in 2016. From 2016 to 2017, he was a Post-Doctoral Fellow of Electrical and Computer Engineering with the University of Toronto, Toronto, ON, Canada. He is currently a Hardy Edmiston Endowed Assistant Professor in the School of Computing and Informatics at the University of Louisiana at Lafayette, Lafayette, LA, USA. He was the receipt of NSF CRII Award and NSF CAREER Award. His research interest focuses on artificial intelligence, cybersecurity, networking and cyber-physical system.