Importance-Aware Data Selection and Resource Allocation in Federated Edge Learning System

Yinghui He, Jinke Ren, Guanding Yu, and Jiantao Yuan

***Abstract*—The implementation of artificial intelligence (AI) in wireless networks is becoming more and more popular because of the growing number of mobile devices and the availability of huge amount of data. However, directly transmitting data for centralized learning will cause long communication latency owing to the limited communication resource and may incur severe privacy issue as well. To address these issues, we consider the federated edge learning (FEEL) system in this paper and develop an importance-aware joint data selection and resource allocation algorithm to maximize the learning efficiency. Aiming at selecting important data for local training, we first analyze the relation between loss decay and gradient norm, which indicates that larger gradient norm leads to faster learning process. Based on this, a learning efficiency maximization problem is formulated by jointly considering the communication resource allocation and data selection. The closed-form results for optimal communication resource allocation and data selection are both developed, where some insights are also highlighted. Also, an optimal algorithm with low computational complexity is developed to obtain the optimal end-to-end latency in one training period. Furthermore, we show that the sample size should be set to its upper limit in order to maximize the learning performance. Finally, we conduct extensive experiments on three popular convolutional neural network (CNN) models. The results show that the proposed algorithm can effectively reduce the training latency and improve the learning accuracy as compared with some benchmark algorithms.**

***Index Terms*—Federated edge learning, learning efficiency, learning accuracy, data selection, data importance, resource allocation.**

I. Introduction

Over the past few years, artificial intelligence (AI) has achieved remarkable success in various areas, such as face recognition, image classification, and natural language processing [1]. AI has also been widely adopted in communication networks for improving the communication performance [2], [3]. The success of AI mainly comes from the large amount of data that are collected for training. However, data in wireless networks are generally distributed over a large amount of mobile devices, which can contribute to the network intelligentization for the future 6G [4], [5]. To fully utilize these data, conventional methods request devices to upload the raw data to a remote cloud server for centralized learning. However, direct data transmission would suffer from two major disadvantages, i.e., the privacy disclosure and the long communication latency, and will eventually degrade the learning performance, such as convergence time and learning accuracy. To address both issues, federated edge learning (FEEL) [6]–[8] has been recently proposed by combining federated learning (FL) [9], a specific distributed training framework, with the mobile edge computing (MEC) [10].

By periodically collecting the local learning updates (either gradient vectors or model parameters) at the network edge, FEEL not only preserves user privacy but also reduces the communication delay. However, calculating and transmitting learning updates may still cause large computation and communication overheads, i.e., high energy consumption and long latency, due to the limited communication resource and computation capacity of the mobile device. To deal with this challenge, several recent works [11]–[16] have investigated the communication-efficient FEEL. A joint bandwidth allocation and scheduling algorithm was developed in [11] to attain certain model accuracy by minimizing the total latency. A broadband analog aggregation technique was proposed in [12] to achieve a low-latency FEEL system based on the over-the-air computation technique and two communication-and-learning tradeoffs were revealed therein. Aiming at reducing energy consumption, the authors in [13] proposed an energy-efficient radio resource management strategy by optimizing bandwidth allocation and user scheduling in each training period. Besides, the authors in [14] investigated the energy-efficient radio resource management for analog aggregation in the FEEL system, where an online energy-aware dynamic worker scheduling policy was proposed under a long-term energy constraint. The tradeoff between energy consumption and end-to-end latency in the FEEL system was studied in both [15] and [16] and some closed-form results were also derived.

The above works mainly aim at reducing the energy consumption and end-to-end latency. However, the learning performance in the FEEL system cannot be guaranteed due to the dynamic channel fading. Towards this end, several works [17], [18] have studied the FEEL system from the perspective of learning performance improvement in wireless fading scenarios. The authors in [17] analyzed the impact of packet errors on the learning performance and minimized the loss function by jointly considering the communication resource allocation and user selection. A novel learning criterion, namely learning efficiency, was proposed in [18], where the batchsize was optimized to dynamically adapt to the wireless channel condition and device computation capacity to improve the learning performance.

The aforementioned works mainly focus on reducing the communication consumption or improving the learning performance by joint resource allocation and user selection, where

Y. He, J. Ren, G. Yu, and J. Yuan are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. e-mail: {2014hyh, renjinke, yuguanding, yuanjiantao}@zju.edu.cn.

the specific data structure is not exploited. However, different data are not equally important to the learning process. To further improve the learning performance, a straightforward way is to select only a part of important data based on their importance level, such as the loss value [19], the change in parameters [20], and the gradient [21], [22]. Some prior works have used the data importance for packet retransmission and user scheduling [23], [24]. In [23], the authors proposed a data-importance aware automatic-repeat-request for both support vector machine (SVM) and convolutional neural networks (CNNs), where the data importance is measured by the uncertainty. Later, based on the elegant communication-learning relation between the signal-to-noise ratio (SNR) and the data importance, an importance-aware user scheduling was developed for the edge learning system in [24] and some principles were proposed to achieve fast convergence.

Although the aforementioned works have developed several data importance indicators, none of them has considered to improve the learning efficiency by exploiting data importance in the FEEL system. Inspired by this, we propose a joint data selection and communication resource allocation algorithm based on the data importance to reduce end-to-end latency and improve learning efficiency in the FEEL system. Our study shows that the communication resource should be allocated dynamically based on wireless channel condition and data importance. The main contributions of this work are summarized as follows.

- We theoretically analyze the impact of the gradient norm on the loss decay, which drives us to use the square of estimated gradient norm after the forward propagation step as the data importance indicator. In this way, the computation latency for local training can be greatly reduced by properly selecting important data.
- To improve the learning performance, we formulate a learning efficiency maximization problem by jointly considering the communication resource allocation and data selection, which is difficult to solve. To tackle this challenging problem, we first develop the optimal data selection strategy and communication resource allocation for given end-to-end latency and sample size. Then, the optimal end-to-end latency can be found by the Golden-section search algorithm with low computational complexity.
- Through theoretical analysis, we find that the expected learning efficiency increases with the sample size, which implies that the sample size should be set to its upper limit to maximize the learning performance. Finally, test results on three popular CNN models show that the proposed scheme can reduce the training latency and improve the learning accuracy at the same time. Moreover, its generalization ability is also demonstrated.

The rest of this paper is organized as follows. In Section II, we introduce the FEEL system with data selection and analyze the delay in each training period. In Section III, we propose a data importance criterion based on the loss decay and formulate an optimization problem to maximize the learning efficiency. The optimal resource allocation and data selection policy is developed in Section IV. Section V presents the test results and the whole paper is concluded in Section VI.

II. System Model and Delay Analysis

In this section, we will introduce an FEEL system with data selection. After that, the detailed procedures and the corresponding latency in each training period are analyzed.

A. FEEL System

In an FEEL system, N devices, denoted by the set $\mathcal{N} = \{1, 2, \cdots, N\}$, collaborate with an edge cloud located at the base station (BS) for training an identical CNN model, as shown in Fig. 1. To achieve it, each device collects data and the dataset of device n is denoted by $\mathcal{D}_n = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_{M_n}, y_{M_n})\}$, where M_n is the size of the n-th device's dataset. In the training process, each device first calculates the local gradient based on its sampled data and then uploads the local gradient to the edge cloud for gradient aggregation. After that, the edge cloud broadcasts the global gradient to all devices and each device updates its CNN model based on the global gradient. However, due to the limited device computation capacity, calculating local gradient is usually time-consuming. Therefore, we are motivated to propose a scheme to reduce the computation consumption by selecting the important data for local training.

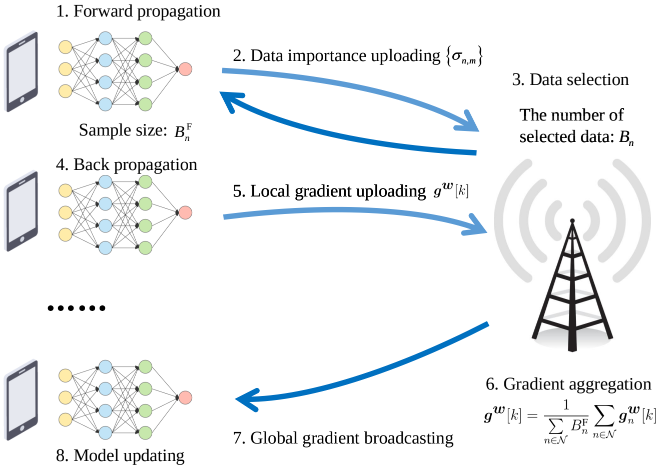
1. Forward propagation
Sample size: B_n^{F}
2. Data importance uploading $\{\sigma_{n,m}\}$
3. Data selection
The number of selected data: B_n
4. Back propagation
5. Local gradient uploading $g^{\boldsymbol{w}}[k]$
......
6. Gradient aggregation
$\boldsymbol{g}^{\boldsymbol{w}}[k] = \frac{1}{\sum_{n\in\mathcal{N}} B_n^{\mathrm{F}}} \sum_{n\in\mathcal{N}} \boldsymbol{g}_n^{\boldsymbol{w}}[k]$
7. Global gradient broadcasting
8. Model updating

Fig. 1. System model.

B. CNN Model

In this paper, we use $\Psi(\boldsymbol{x}, \boldsymbol{w})$ to represent the CNN model with parameter vector $\boldsymbol{w}$ and measure the training error of data sample $(\boldsymbol{x}_i, y_i)$ by the loss function $\ell\left(\Psi(\boldsymbol{x}_i, \boldsymbol{w}), y_i\right)$. Then, the local loss function of each device can be given by

$$L_n\left(\boldsymbol{w}, \mathcal{D}_n\right) = \sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}_n} \ell\left(\Psi(\boldsymbol{x}_i, \boldsymbol{w}), y_i\right), \ \forall n \in \mathcal{N}. \tag{1}$$

Consequently, the global loss at the edge cloud can be expressed as the average of all local loss functions, as

$$L(\boldsymbol{w}) = \frac{1}{|\cup_{n\in\mathcal{N}}\mathcal{D}_n|} \sum_{n\in\mathcal{N}} L_n\left(\boldsymbol{w}, \mathcal{D}_n\right). \tag{2}$$

To minimize the global loss, stochastic gradient descent (SGD) algorithm is widely used. Specifically, a subdataset $\widetilde{\mathcal{D}}$ is sampled from the dataset to calculate the gradient in each iteration. Then, the gradient vector of each device can be expressed as

$$\boldsymbol{g}_n^{\boldsymbol{w}} = \nabla L_n\left(\boldsymbol{w}, \widetilde{\mathcal{D}}_n\right) = \sum_{(\boldsymbol{x}_i, y_i)\in\widetilde{\mathcal{D}}_n} \frac{\partial \ell\left(\Psi(\boldsymbol{x}_i, \boldsymbol{w}), y_i\right)}{\partial \boldsymbol{w}}, \ \forall n \in \mathcal{N}. \tag{3}$$

According to [25], the gradient is calculated by two steps: forward propagation and back propagation. The forward propagation calculates the loss of each data and the backward propagation calculates the gradient based on the loss value. Moreover, in the k-th training period, the parameter updating at the edge cloud is given by

$$\begin{aligned}\boldsymbol{w}[k+1] &= \boldsymbol{w}[k] - \eta[k]\boldsymbol{g}^{\boldsymbol{w}}[k] \\ &= \boldsymbol{w}[k] - \eta[k]\frac{1}{\left|\cup_n \widetilde{\mathcal{D}}_n\right|}\sum_{n\in\mathcal{N}}\sum_{(\boldsymbol{x}_i, y_i)\in\widetilde{\mathcal{D}}_n}\frac{\partial \ell\left(\Psi(\boldsymbol{x}_i, \boldsymbol{w}[k]), y_i\right)}{\partial \boldsymbol{w}[k]},\end{aligned} \tag{4}$$

where $\boldsymbol{g}^{\boldsymbol{w}}[k]$ is the global gradient and $\eta[k]$ is the learning rate.

According to [7], model training can be accelerated by using important data for training. Let $\sigma_{n,m}\left(m \in \{1, 2, \cdots, M_n\}, n \in \mathcal{N}\right)$ measure the importance of the m-th data of device n. Moreover, we assume that $\{\sigma_{n,m}\}$ is sorted in a non-increasing order, i.e., $\sigma_{n,m} \geq \sigma_{n,m+1}, \forall n, m$. To measure the data importance, we need to calculate the loss of the data [23], [26], which can be derived after forward propagation. By selecting the important data, the gradients of these unimportant data are not needed to be calculated in the backward propagation step. In this way, the training latency can be reduced and the learning efficiency can be improved.

C. Wireless Channel Model

In the FEEL system, there exist two transmission stages in each training period, i.e, local gradient uploading and global gradient broadcasting.

In the local gradient uploading stage, we adopt the time division multiple access (TDMA) method for data transmission, where each time frame is divided into N time-slots. Denote W and N_0 as the system bandwidth and the noise power, respectively. Let h_n^{U} denote the channel power gain of the n-th device and p_n^{U} denote the corresponding transmit power. Since the data size of the gradient vector is usually large, the latency for gradient uploading (more than one second) is much longer than each time frame (10 ms in LTE standard [27])[1]. Therefore, we use the average achievable data rate to evaluate the device n's latency of gradient uploading [29], as

$$R_n^{\mathrm{U}} = W\mathbb{E}_h\left\{\log_2\left(1 + \frac{p_n^{\mathrm{U}}\left|h_n^{\mathrm{U}}\right|^2}{N_0}\right)\right\}, \tag{5}$$

where $\mathbb{E}_h\{\cdot\}$ is the expectation over the channel power gain. We should note that TDMA method is adopted in this paper since it has been widely used in current communication systems. The synchronization issue during the training period can be well guaranteed by the TDMA method. Nevertheless, our results can be extended to other access methods, such as non-orthogonal multiple access (NOMA) and orthogonal frequency division multiple access (OFDMA), with some modifications on the data rate model.

In the global gradient broadcasting stage, we adopt broadcasting for data transmission since the global gradient is the same for all devices. Then, the achievable data rate for all devices can be expressed as

$$R^{\mathrm{D}} = \min_{n\in\mathcal{N}}\left\{W\mathbb{E}_h\left\{\log_2\left(1 + \frac{p^{\mathrm{B}}\left|h_n^{\mathrm{B}}\right|^2}{N_0}\right)\right\}\right\}, \tag{6}$$

where h_n^{B} denotes the downlink channel power gain of device n and p^{B} denotes the corresponding transmit power of the BS.

D. Latency Analysis

As mentioned before, we aim at reducing the local computation latency and improving the learning efficiency via data selection based on data importance. Therefore, the end-to-end latency is essential and should be quantitatively analyzed. As shown in Fig. 1, the detailed procedures and the corresponding latency in each training period can be analyzed as follows.

1) **Forward propagation**. A subdataset is first sampled from each local dataset with the sample size B_n^{F}. After that, each device calculates the loss of its sampled data and the importance $\sigma_{n,m}$. Let t_n^{F} denote the forward propagation speed of device n, i.e., computation latency of forward propagation per data. Then, the latency for the forward propagation can be expressed as

$$T_n^{\mathrm{F}} = B_n^{\mathrm{F}} t_n^{\mathrm{F}}, \ \forall n \in \mathcal{N}. \tag{7}$$

2) **Data importance uploading**. After forward propagation, each device uploads the data importance $\sigma_{n,m}$ to the edge cloud via TDMA method.[2] Since the data importance is only a scalar, its size is small enough so that its transmission delay can be ignored.
3) **Data selection**. After receiving the data importance from all devices, the edge cloud then selects data based on their importance values and channel data rates. Specifically, the edge cloud feeds back the number of selected data, denoted by B_n, to each device.
4) **Backward propagation**. After receiving the number of selected data, device n will pick B_n data with the largest data importance to perform backward propagation and calculate the local gradient vector, $\boldsymbol{g}_n^{\boldsymbol{w}}[k]$. Let t_n^{B} denote the backward propagation speed of device n, i.e., computation latency of backward propagation per data. Then,

[1] Take ResNet18 as an example. The data size of the gradient vector is about 342 MBits. The uplink data rate in LTE standard is 50 Mbps and the transmit delay is more than one second. Even if we use the gradient compression method in [28], the transmission delay is still about 0.5 second.

[2] If one device fails to upload the data importance, it cannot join this training period. However, this device can still receive the global gradient vector from the edge cloud, which ensures that it can join the next training period.

the total latency for each device to perform backward propagation is

$$T_n^{\mathrm{B}} = t_n^{\mathrm{B}} B_n,\ \forall n \in \mathcal{N}. \tag{8}$$

5) **Local gradient uploading**. Each device sends its local gradient to the edge cloud. As we mentioned before, we adopt TDMA method for local gradient uploading. Let τ_n denote the proportion of device n's slot in one time frame. Then, according to [18], [29], the transmission delay of device n can be expressed as

$$T_n^{\mathrm{U}} = \frac{V}{\tau_n R_n^{\mathrm{U}}}, \tag{9}$$

where V is the data size of local gradient and is a constant for all devices.

6) **Gradient aggregation**. After receiving the local gradient vectors from all devices, the edge cloud aggregates them to calculate the global gradient, as

$$\boldsymbol{g}^{\boldsymbol{w}}[k] = \frac{1}{\sum\limits_{n\in\mathcal{N}} B_n^{\mathrm{F}}} \sum_{n\in\mathcal{N}} \boldsymbol{g}_n^{\boldsymbol{w}}[k]. \tag{10}$$

Since gradient aggregation is only an average operation, the latency of this stage can be neglected.

7) **Global gradient broadcasting**. After gradient aggregation, the edge cloud broadcasts the global gradient vector to all devices. Thus, the latency for global gradient broadcasting is given as follows for all devices

$$T^{\mathrm{D}} = \frac{V}{R^{\mathrm{D}}}. \tag{11}$$

8) **Model updating**. Each device then updates its model according to (4). Since the initial parameters $\boldsymbol{w}[0]$ is the same, each device shares the same parameter after updating. The updating latency of device n is denoted by T_n^{M}, which cannot be neglected due to its limited computation capacity.

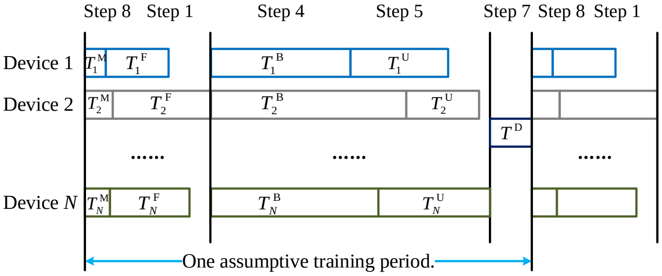

Fig. 2. Timing chart of one training period. Steps 2, 3, and 6 are not shown in the timing chart since the latency is too short that can be neglected.

Based on the above analysis, we can draw the timing chart as shown in Fig. 2. Note that the data selection step cannot be performed until receiving the data importance from all devices and the gradient aggregation step cannot be performed until receiving the local gradient vectors from all devices. Besides, although one training period starts from forward propagation and ends with model updating, it can be better understood if we assume that one training period starts from model updating and ends with global gradient broadcasting. Let $T^{\mathrm{M,F}}$ denote the total latency for model updating and forward propagation, i.e., $T^{\mathrm{M,F}} = \max\limits_{n\in\mathcal{N}} \left\{T_n^{\mathrm{M}} + T_n^{\mathrm{F}}\right\}$, representing the latency of step 1 and step 8. Then, the total latency of one training period can be expressed as

$$T = T^{\mathrm{M,F}} + \max_{n\in\mathcal{N}} \left\{T_n^{\mathrm{B}} + T_n^{\mathrm{U}}\right\} + T^{\mathrm{D}}. \tag{12}$$

To better utilize the time for forward propagation, each device should calculate the loss of its sampled data as many as possible in order to offer more data choices. Therefore, the sample size should be

$$B_n^{\mathrm{F}} = \frac{T^{\mathrm{M,F}} - T_n^{\mathrm{M}}}{t_n^{\mathrm{F}}},\ \forall n \in \mathcal{N}. \tag{13}$$

III. Learning Efficiency Analysis

In this section, we will first propose the data importance metric. Then, an optimization problem is formulated to maximize the learning efficiency.

A. Data Importance and Learning Efficiency

In this paper, we aim at reducing the local computation latency and improving the learning efficiency by data selection. To this end, we should first define the data importance based on model updating, which eventually influences the learning performance. From (4), we can find that the gradient vector influences the model updating, indicating that the data importance can be measured by its gradient. With a greater gradient, the data can contribute more to the parameter updating and is more important to the model convergence. However, the real gradient vector is obtained after two steps, forward propagation and back propagation. Therefore, the computation consumption would be very large if we select data based on the real gradient vector. Fortunately, we can estimate the norm of gradient vector based on the loss of each data after forward propagation according to [26], as

$$\begin{aligned} \left|\left|\boldsymbol{g}^{\boldsymbol{w}}[k](\boldsymbol{x}_i, y_i)\right|\right|_2 &= \left|\left|\frac{\partial \ell\left(\Psi(\boldsymbol{x}_i, \boldsymbol{w}), y_i\right)}{\partial \boldsymbol{w}}\right|\right|_2 \\ &\approx \rho \left|\left|\frac{\partial \ell\left(\Psi(\boldsymbol{x}_i, \boldsymbol{w}), y_i\right)}{\partial \boldsymbol{x}_i^L}\right|\right|_2, \end{aligned} \tag{14}$$

where $\boldsymbol{x}_i^L$ is the input of the active function in the output layer of CNN, ρ is a coefficient determined by the CNN model, and $||\cdot||_2$ is the L2 norm. According to [26], the backward propagation requires about twice the amount of time as the forward propagation since it needs to compute full gradients. Since the proposed method only calculates the gradient of the parameter in the output layer, it can significantly reduce the computation cost. Therefore, the latency for estimating the gradient norm can be greatly reduced.

Till now, we have found that the gradient vector influences the learning performance. However, the mathematical relation between them is not clear. To tackle it, we first measure the learning performance improvement by the global loss decay in one training period, as

$$\Delta L[k] = L(\boldsymbol{w}[k-1]) - L(\boldsymbol{w}[k]). \tag{15}$$

According to [30], the relation between the global loss decay and the gradient norm is

$$\Delta L[k] = \gamma \left|\left|\boldsymbol{g}^{\boldsymbol{w}}[k]\right|\right|_2^2, \tag{16}$$

where γ is a coefficient determined by the specific CNN model. Then, by combining (14) and (16), we can conclude that the square of estimated gradient norm can measure the loss decay. Therefore, we can define the data importance in the following.

Definition 1: The importance of data $(\boldsymbol{x}_i, y_i)$ in device n can be evaluated by the square of its estimated gradient norm, as

$$\sigma_{n,i} = \rho\gamma \left|\left|\frac{\partial \ell\left(\Psi(\boldsymbol{x}_i, \boldsymbol{w}), y_i\right)}{\partial \boldsymbol{x}_i^L}\right|\right|_2^2. \tag{17}$$

We should note that although our analysis is based on CNN, the proposed data importance can be utilized for other neural networks that adopt SGD algorithm to minimize the loss function. From the above definition, the importance of data represents the loss decay it can bring to the global loss function. Then, the loss decay function of device n can be defined as

$$f_n(B_n) = \sum_{m=1}^{B_n} \sigma_{n,m}, \;\; B_n \leq B_n^{\mathrm{F}}, \tag{18}$$

which represents the total loss decay brought by device n when B_n data with the largest importance are selected. It should be noted that it is challenging to analyze the function $f_n(B_n)$ since B_n is a discrete variable. However, we can relax B_n into a continuous variable since B_n^{F} is typically large, such as 128. Correspondingly, $f_n(B_n)$ can be fitted as a piecewise linear function, as shown in Fig. 3. Then, we have the following lemma.

Lemma 1: $f_n(B_n)$ is a concave function when B_n is relaxed into a continuous variable.

Proof: Please see Appendix A. ■

Based on the lemma, the global loss decay in one training period is given by $\Delta L = \sum\limits_{n\in\mathcal{N}} f_n(B_n)$, which is the summation of the loss decay brought by all devices. Therefore, according to [18], the learning efficiency can be defined as

$$E = \frac{\Delta L}{T} = \frac{\sum\limits_{n\in\mathcal{N}} f_n(B_n)}{T^{\mathrm{M,F}} + \max\limits_{n\in\mathcal{N}}\{T_n^{\mathrm{B}} + T_n^{\mathrm{U}}\} + T^{\mathrm{D}}}, \tag{19}$$

which represents the global loss decay rate in each training period. We should note that the learning efficiency jointly considers the learning performance (ΔL) and the communication performance (T). The improvement of learning efficiency represents that both the learning performance and communication performance are improved. By maximizing the learning efficiency, the training process can be accelerated.

B. Problem Formulation

In this paper, we aim at maximizing the learning efficiency. The optimization problem can be mathematically formulated as

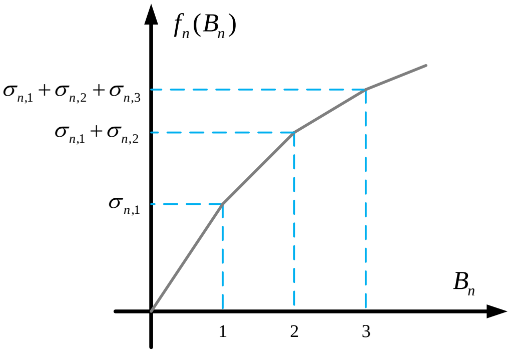

Fig. 3. The loss decay function of device n.

$$\mathcal{P}1: \max_{\{B_n,\tau_n,T,T^{\mathrm{M,F}}\}} \frac{\sum_{n=1}^{N} f_n(B_n)}{T^{\mathrm{M,F}} + \max\limits_{n\in\mathcal{N}}\{T_n^{\mathrm{B}} + T_n^{\mathrm{U}}\} + T^{\mathrm{D}}}, \tag{20a}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} \tau_n \leq 1, \tag{20b}$$

$$B_n^{\min} \leq B_n \leq B_n^{\mathrm{F}}, \;\forall n \in \mathcal{N}, \tag{20c}$$

$$T^{\mathrm{M,F}} \leq T^{\mathrm{M,F,max}}, \;\; \forall n \in \mathcal{N}, \tag{20d}$$

$$B_n^{\mathrm{F}}, \tau_n, T, T^{\mathrm{M,F}} \geq 0, \;\forall n \in \mathcal{N}. \tag{20e}$$

In the above, (20b) represents the uplink communication resource limitation, (20c) guarantees that the number of selected data should not exceed the sample size and should not be less than the minimum number requirement $B_n^{\min}$ to ensure the participation of each device, and (20d) bounds the maximum latency of model updating and forward propagation, denoted by $T^{\mathrm{M,F,max}}$, due to the hardware limitation, such as memory size.

The optimization variables in problem $\mathcal{P}1$ contain the data selection (B_n), the communication resource allocation (τ_n), the total latency of model updating and the forward propagation ($T^{\mathrm{M,F}}$), and the total latency of one training period (T). Note that the reason for optimizing $T^{\mathrm{M,F}}$ here is that $T^{\mathrm{M,F}}$ determines the sample size, which will also influence the data selection.

We should note that problem $\mathcal{P}1$ is not easy to solve for the following reasons. First, $T^{\mathrm{M,F}}$ can influence the sample size $\{B_n^{\mathrm{F}}\}$ and the loss decay function. With different $T^{\mathrm{M,F}}$, the learning efficiency would be different but it is hard to give a detailed expression between the learning efficiency and $T^{\mathrm{M,F}}$. Secondly, $T^{\mathrm{M,F}}$ should be decided before the forward propagation. Last but not the least, even if $T^{\mathrm{M,F}}$ is given, problem $\mathcal{P}1$ is still non-convex and cannot be solved directly.

Based on the considerations, we will first solve the problem when $T^{\mathrm{M,F}}$ is given and then analyze the impact of $T^{\mathrm{M,F}}$ on the learning efficiency in the following section.

IV. Optimal Solution

In this section, we first analyze the problem for given the latency of model updating and forward propagation, i.e., $T^{\mathrm{M,F}}$. Then, the optimal data selection strategy and resource allocation policy are proposed. After that, the optimal solution to problem $\mathcal{P}1$ is obtained by leveraging the probability theory.

A. Optimal Data Selection and Resource Allocation

As mentioned before, $T^{\text{M,F}}$ is hard to be optimized directly. Therefore, we first consider the following problem with given $T^{\text{M,F}}$, as

$$\mathcal{P}2: \quad \max_{\{B_n,\tau_n,T\}} \quad \frac{\sum_{n=1}^{N} f_n(B_n)}{T}, \tag{21a}$$

$$\text{s.t.} \quad T^{\text{M,F}} + T_n^{\text{B}} + T_n^{\text{U}} + T^{\text{D}} \le T,\ \forall n \in \mathcal{N}, \tag{21b}$$

$$\text{(20b), (20c), and (20e).}$$

Note that problem $\mathcal{P}2$ is still non-convex. To solve it, we further assume that the total latency of one training period, T, is given. Then problem $\mathcal{P}2$ becomes convex since the objective function is concave and all constraints are convex. Therefore, we can utilize the Lagrangian method to find the optimal solution. The partial Lagrange function can be defined as

$$L = -\frac{\sum_{n=1}^{N} f_n(B_n)}{T} + \lambda \left(\sum_{n=1}^{N} \tau_n - 1 \right) + \sum_{n=1}^{N} \mu_n \left(T^{\text{M,F}} + B_n t_n^{\text{B}} + \frac{V}{\tau_n R_n^{\text{U}}} + T^{\text{D}} - T \right), \tag{22}$$

where λ and μ_n are the Lagrange multipliers associated with the constraints (20b) and (21b), respectively. Before presenting the optimal solution, we should note that $f_n(B_n)$ is not differentiable at some points so that its subgradient can be expressed as

$$\frac{\partial f_n(B_n)}{\partial B_n} \begin{cases} = \sigma_{n,\lceil B_n \rceil}, & \text{if } B_n \notin \mathcal{Z}, \\ \in \left[\sigma_{n,B_n+1}, \sigma_{n,B_n}\right], & \text{otherwise,} \end{cases} \tag{23}$$

where $\mathcal{Z}$ is the set of non-negative integers and $\lceil \cdot \rceil$ represents the rounding up operation. Moreover, we define the maximum number of selected data as

$$B_n^{\max} = \min\left(B_n^{\text{F}}, \frac{T - T^{\text{M,F}} - T^{\text{D}}}{t_n^{\text{B}}} \right), \tag{24}$$

which is decided by the sample size and the latency for backward propagation. Then, the data selection function can also be defined as

$$g_n(B_n) = \frac{\partial f_n(B_n)}{\partial B_n} \left(T - T^{\text{M,F}} - T^{\text{D}} - B_n t_n^{\text{B}} \right)^2 \frac{R_n^{\text{U}}}{VT}, \tag{25}$$

which can be derived by solving problem $\mathcal{P}2$ under given T.

Denote $\{B_n^*, \tau_n^*\}$ as the optimal solution to problem $\mathcal{P}2$ under given T. Then, by applying the Karush-Kuhn-Tucker (KKT) conditions and simple mathematical calculation, we can obtain the optimal data selection and communication resource allocation policy, as shown in Theorem 1.

Theorem 1: The optimal data selection strategy and communication resource allocation policy can be expressed as

$$\begin{cases} B_n^* = \left[\phi_n(\lambda^*)\right]_{B_n^{\min}}^{B_n^{\max}}, & \forall n \in \mathcal{N}, \quad \text{(26a)} \\ \tau_n^* = \left(\dfrac{V}{\left(T - T^{\text{M,F}} - T^{\text{D}} - B_n^* t_n^{\text{B}}\right) R_n^{\text{U}}} \right)^+, & \forall n \in \mathcal{N}, \quad \text{(26b)} \end{cases}$$

where $\phi_n(x)$ is the inverse function of $g_n(B_n)$ and λ^* is the optimal value of the Lagrange multiplier satisfying the communication resource limitation: $\sum_{n=1}^{N} \tau_n = 1$. Note that $[X]_b^a = \max\{\min\{X,a\}, b\}$ and $(X)^+ = \max\{X, 0\}$.

Proof: Please see Appendix B. ■

Remark 1: The optimal data selection is achieved when $g_n(B_n) = \lambda^*$ is satisfied. Without loss of generality, we can consider the case that $B_n^* \notin \mathcal{Z}$, where the gradient of $f_n(B_n)$ is equal to the importance of $\lceil B_n \rceil$-th data, i.e., $\sigma_{n,\lceil B_n \rceil}$. Then, the date selection strategy is mainly determined by the backward propagation speed, the data importance, and the data rate. First of all, $g_n(B_n)$ decreases with B_n since $\sigma_{n,\lceil B_n \rceil}$ decreases with B_n. More specifically, $g_n(B_n)$ decreases with B_n in the power of 2 even if $\sigma_{n,\lceil B_n \rceil}$ is fixed. Based on this, we can conclude that one device with more important data would have more data to be selected to perform backward propagation, which consists with our intuition. On the other hand, $g_n(B_n)$ decreases with the backward propagation speed t_n^{B} in the power of 2. The higher the backward propagation speed is, the more data the device can calculate. Besides, a device with higher data rate is likely to have more selected data.

Moreover, from (26b), the communication resource allocation policy is related with the backward propagation speed, the data rate, and the optimal number of selected data. A device with higher backward propagation speed and data rate needs less communication resource to satisfy the latency requirement. Furthermore, the optimal communication resource allocation policy in (26b) indicates that the end of each device's transmission is synchronized, which can fully utilizes the available time to improve the learning efficiency.

The Lagrange multiplier λ^* in (26a) can be determined by classical bisection search algorithm. To reduce the computational complexity, we first analyze the upper and lower bounds for λ^*. Since λ^* is the Lagrange multiplier associated with the inequality constraint (20b), the lower bound is zero. The upper bound of λ^* is derived when the number of selected data of each device is the minimum one, $B_n^{\min}$. Then, by simple mathematical calculation, we have the range of λ^* as shown in the following lemma.

Lemma 2: The range of λ^* is given by

$$\begin{cases} \lambda^* \ge \lambda_{\min} = 0, & \text{(27a)} \\ \lambda^* \le \lambda_{\max} \\ \quad = \max\limits_{n \in \mathcal{N}} \left\{ \dfrac{\sigma_{n,B_n^{\min}} \left(T - T^{\text{M,F}} - T^{\text{D}} - B_n^{\min} t_n^{\text{B}}\right)^2 R_n^{\text{U}}}{VT} \right\}. & \text{(27b)} \end{cases}$$

The detailed algorithm for searching λ^* is given in Algorithm 1. The main idea is to update the value of λ until the communication resource allocation constraint (20b) is satisfied. The computational complexity is $\mathcal{O}\left(N \log(1/\epsilon)\right)$, where ϵ is the maximal error tolerance.

B. Optimal Selection of T

Thus far, we have obtained the optimal $E(T)$ under the given total latency requirement T. Then, we optimize T to develop an optimal solution to problem $\mathcal{P}2$. Note that the optimal data selection strategy and resource allocation policy are influenced by the total latency requirement. Therefore, we introduce the following theorem to ensure that our algorithm can find the optimal solution to problem $\mathcal{P}2$.

Algorithm 1: Bisection search algorithm for λ^*.

```
Set the maximal error tolerance $\epsilon$;
Set $\lambda_\ell = \lambda_{\min}$, $\lambda_u = \lambda_{\max}$;
repeat
    Let $\lambda = (\lambda_\ell + \lambda_u)/2$;
    Obtain $\{B_n, \tau_n\}$ according to (26a) and (26b);
    if $\sum_{n=1}^{N} \tau_n > 1$ then
        $\lambda_u = \lambda$;
    else
        $\lambda_\ell = \lambda$;
    end
until $\left|\sum_{n=1}^{N} \tau_n - 1\right| < \epsilon$;
Output $\lambda^* = \lambda$ and the corresponding $\{B_n^*, \tau_n^*\}$.
```

Theorem 2: $E(T)$ is a strictly unimodal function with $T \geq 0$.

Proof: Please see Appendix C. ■

Remark 2: A unimodal function is a function that has only one peak (maximum) or valley (minimum) in a given interval. Specifically, $E(T)$ has only one peak as it first increases and then decreases with T, as shown in Fig. 4. From Theorem 2, the local maximum is the global one in the given interval. Since the gradient of $E(T)$ cannot be directly expressed, we can utilize the Golden-section search algorithm [31] to find the optimal T^*. By narrowing the range of values, Golden-section search algorithm can efficiently find an extremum of a function inside a specified interval. As shown in Fig. 4, the extremum would not be in $[T_2, T_{\max}]$ since $E(T_1) > E(T_2)$. Therefore, the new range becomes $[T_{\min}, T_2]$. Moreover, T_1 and T_2 are decided based on the golden ratio.

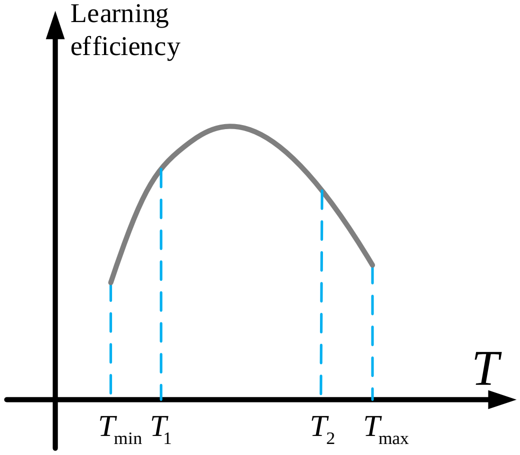

Fig. 4. The relation between learning efficiency $E(T)$ and total latency of one training period T.

To better perform the Golden-section search algorithm, we have the following lemma about the range of T^*.

Lemma 3: The range of T^* is given by

$$\begin{cases} T^* \geq T_{\min} = T^{\text{M,F}} + \sum_{n=1}^{N} \frac{V}{R_n} + T^{\text{D}}, & \text{(28a)} \\ T^* \leq T_{\max} = T^{\text{M,F}} + \max_{n \in \mathcal{N}} \left\{ B_n^{\text{F}} t_n^{\text{B}} + \frac{VN}{R_n^{\text{U}}} \right\} + T^{\text{D}}. & \text{(28b)} \end{cases}$$

Proof: Please see Appendix D. ■

The lower bound in (28a) corresponds to the case where no data is selected for backward propagation and the communication resource allocation is obtained to minimize the total latency. The upper bound in (28b) corresponds to the case where all data are selected for backward propagation and the communication resource is equally allocated to each device.

Based on above analysis, we can obtain the optimal algorithm to problem $\mathcal{P}2$, as described in Algorithm 2.

Algorithm 2: Optimal algorithm for problem $\mathcal{P}2$.

```
Set the maximal error tolerance $\epsilon$;
Set $T_\ell = T_{\min}$, $T_u = T_{\max}$;
while $|T_\ell - T_u| > \epsilon$ do
    Let $T_1 = T_u - \frac{2}{\sqrt{5}+1}(T_u - T_\ell), T_2 =$
      $T_\ell + \frac{2}{\sqrt{5}+1}(T_u - T_\ell)$;
    According to Theorem 1, obtain the optimal
      values, $E(T_1)$ and $E(T_2)$, when $T$ is set to $T_1$
      and $T_2$, respectively;
    if $E(T_1) > E(T_2)$ then
        $T_u = T_2$;
    else
        $T_\ell = T_1$;
    end
end
Output the optimal solution.
```

C. Optimal Selection of $T^{M,F}$

Thus far, we have obtained the optimal solution to problem $\mathcal{P}1$ for the given value of $T^{\text{M,F}}$. In the following, we will analyze how to choose $T^{\text{M,F}}$ for further performance improvement.

As mentioned in Section III, the specific impact of $T^{\text{M,F}}$ on the loss decay function is hard to mathematically expressed. Thus, we consider using the probability theory for analysis. Recall that $f_n(B_n)$ is the summation of the B_n most important data for device n. Then, according to [32], when the proportion of the selected data to the sample size, denoted by $q_n = B_n/B_n^{\text{F}}$, is fixed, we have

$$\lim_{B_n \to \infty} \frac{\mathbb{E}\left\{f_n\left(B_n\right)\right\}}{B_n} = C_n, \ \forall n \in \mathcal{N}, \tag{29}$$

where C_n is the mean value of loss decay of the selected data. From (29), the expectation of $f_n(B_n)$ is proportional to B_n when B_n is large. Then, the total loss decay is also proportional to the sample size. Meanwhile, the total latency of one training period is composed of two parts: the latency for local computation and the latency for communication. The former is proportional to the sample size while the latter is a constant. Therefore, the learning efficiency will increase with $T^{\mathrm{M,F}}$ until it reaches its limit, as described in the following theorem.

Theorem 3: Let E^* denote the optimal learning efficiency to problem $\mathcal{P}1$. Then, the expectation of E^* increases with $T^{\mathrm{M,F}}$ and its limit is

$$\lim_{T^{\mathrm{M,F}}\to\infty} \mathbb{E}\left\{E^*\right\} = \sum_{n=1}^{N} \frac{q_n C_n}{q_n t_n^{\mathrm{B}} + t_n^{\mathrm{F}}}. \tag{30}$$

Proof: Please see Appendix E. ■

Remark 3: From (30), the limit of the learning efficiency is mainly determined by the computation speed and the proportion of the selected data. With a faster computation speed, less time is required to perform a training period. Thus, the learning efficiency can be improved. Moreover, the learning efficiency increases with the proportion of important data, q_n. The reason is explained as follows. The proportion of the selected data in the training process is related to the model accuracy of CNN. Specifically, fewer data become important as the model accuracy increases, which eventually reduces the learning efficiency.

Based on Theorem 3, we should sample as many data as possible in the forward propagation. However, due to the hardware limitation, such as memory size, $T^{\mathrm{M,F}}$ has its upper limit, as

$$T^{\mathrm{M,F,max}} = \min_{n\in\mathcal{N}} \left(T^{\mathrm{M}} + t_n^{\mathrm{F}} B_n^{\mathrm{F,max}}\right), \tag{31}$$

where $B_n^{\mathrm{F,max}}$ is the maximal sample size for device n. It should be noted that the maximal sample size is determined before the forward propagation stage. Therefore, $T^{\mathrm{M,F}}$ can be set to its upper limit in advance to maximize the learning efficiency.

V. Test Results

In this section, we conduct experiments to evaluate the performance of the proposed data-importance-aware FEEL scheme.

A. Methodology

Wireless communication system: The BS covers a circle area with a radius of 300 m and 6 devices are randomly located in the coverage. The system bandwidth (W) is 10 MHz and the noise spectral density (N_0) is -174 dBm/Hz. The channel gains of cellular links (h_n^{U}, h_n^{B}) are all generated according to the path loss model, $128.1 + 37.6 \log\left(d \text{ [km]}\right)$, and the small-scale fading follows the Rayleigh distributed with uniform variance. The transmit power (p_n^{U}) is set as 24 dBm for all devices.

CNN model: We consider three common CNN models for image classification, ResNet18, DenseNet121, and MobileNetV2. The corresponding training dataset is CIFAR-10 [28], which consists of 60,000 32×32 colour images in 10 different classes. Specifically, it contains 50,000 training images and 10,000 test images. All training images are randomly partitioned into six equal parts, which are assigned to six devices, respectively. Due to the memory size limitation, the maximum sample size ($B_n^{\mathrm{F,max}}$) is set as 128 for each device. To ensure the participation of each device, we set the minimum sample size (B_n^{min}) as 40. Moreover, the learning rate (η) is set as 0.001 for all CNN models unless otherwise specified.

The computation frequency of each device is set as: 2 devices with 3.4 GHz, 2 devices with 3.8 GHz, and 2 devices with 4.3 GHz.

B. Generalization Tests

In this part, we first test the generalization ability of the proposed scheme. We implement our proposed scheme on the three CNN models mentioned above and compare the proposed algorithm with the conventional one where there is no data selection and all sampled data join the backward propagation. The learning accuracy and loss with different CNN models are shown in Fig. 5 and Fig. 6, respectively. We should note that curves in Fig. 5 and Fig. 6 vary with training period instead of training time. From Fig. 5, the proposed scheme achieves a significantly better performance than the conventional one for all three CNN models. Correspondingly, the loss of the proposed scheme decreases faster than that of conventional scheme, as depicted in Fig. 6. The results in both figures indicate two points. First, our proposed scheme is practical and has a good generalization ability. Secondly, the proposed scheme can accelerate the training process by selecting important data.

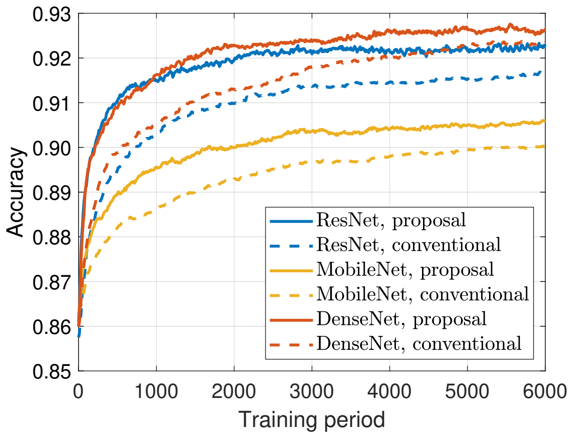

Fig. 5. The test accuracy with different CNN models.

To better verify the second point, we first introduce the gradient norm ratio that is the ratio of the selected data's gradient norm to all data's gradient norm. It can describe whether the selected data can represent all data. The selected data can exactly represent all data when the gradient norm achieves one. Table I illustrates the proportion of the selected data and the corresponding gradient norm ratio. Since they are almost the same during the training period in Fig. 5 and Fig. 6, we only plot the average values for different CNN models. It can be observed that although only a small proportion (around 36%) of data are selected, the norm of those selected data's gradient is almost the same as if all data are selected. The results confirm that our proposal can speed up the training

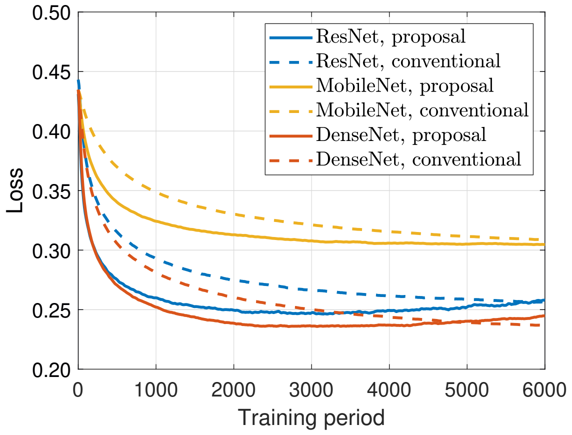

Fig. 6. The test loss with different CNN models.

TABLE I
THE PROPORTION OF SELECTED DATA AND THE CORRESPONDING GRADIENT NORM RATIO.

CNN Models	ResNet	MobileNet	DenseNet
Proportion of selected data	35.25%	36.59%	35.50%
Gradient norm ratio	99.94%	100.00%	99.95%

period by data selection since only important data are selected for backward propagation.

In Fig. 5, we can see that the proposed scheme achieves a higher final learning accuracy than the conventional scheme for all three CNN models. The reason can be explained as follows. During each training period, we select the important data with large gradient norm, which is not predicted well by CNN model. By only training those data, the model can improve its accuracy. However, we can find that the loss of our proposed scheme in Fig. 6 slightly increases at the end of training. It is because most data are unimportant at the end of training and the loss of unimportant data may increase though the loss of important data can be reduced.

C. Performance Comparison

In this part, to show the performance improvement of our proposed algorithm, we will compare it with several benchmarks as follows.

- *Equal resource scheme*: This scheme is similar to our proposal except that the communication resource is equally allocated to each device, i.e., $\tau_n = 1/N$.
- *All selected scheme*: This scheme is similar to our proposal except that all sampled data are selected for gradient calculation, i.e., $B_n = B_n^{\mathrm{F}}$.
- *Conventional scheme*: In this scheme, all sampled data join the backward propagation in each device, i.e., $B_n = B_n^{\mathrm{F}}$, and the communication resource is equally allocated to each device, i.e., $\tau_n = 1/N$.
- *Half sample size scheme*: This scheme is similar to our proposal except that the sample size is set as 64 that is half of the maximum sample size.

We consider 6 immobile devices whose distances to BS are ranked as: device 1 $<$ device 2 $<$ device 3 $<$ device 4 $<$ device 5 $<$ device 6, which influence the large-scale fading. The small-scale fading varies with time for each device. Without loss of generality, we adopt ResNet18 for the following tests since different CNN models have the same comparative results.

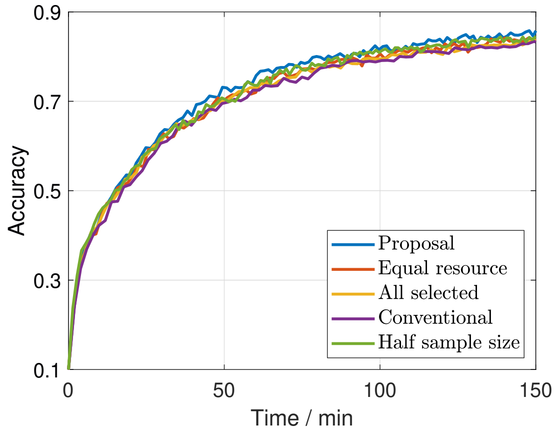

Fig. 7. The test accuracy of different schemes.

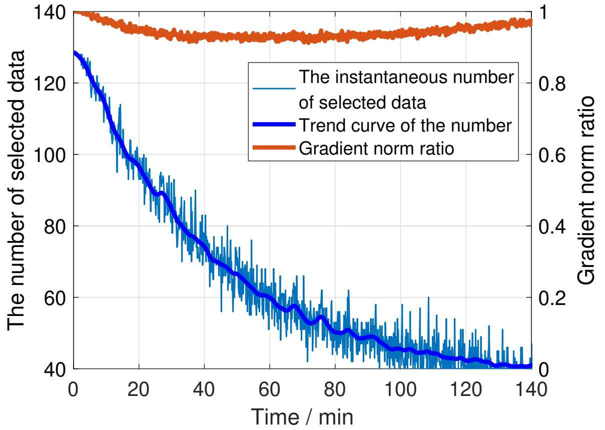

Fig. 8. The number of selected data and the gradient norm ratio for device 6.

At the beginning of the training process, we set the learning rate as 0.01. Fig. 7 shows the learning accuracy with different schemes. From the figure, all schemes almost have the same performance when the learning accuracy is low. The reasons are twofold. First, the model cannot fit dataset well and most of the data are important to be selected with low learning accuracy, which means that data selection has little performance gain in this case. Secondly, the communication latency is shorter than the computation latency when the number of selected data is large.

As training period continues, the learning accuracy increases and the number of selected data decreases. Fig. 8 plots the variation of the number of selected data with time and the gradient norm ratio for device 6, as an example. Since some data can be predicted well by the model with high learning accuracy, they become unimportant. Meanwhile, we should note that the gradient norm ratio is always above 90%, even almost 100%, although the number of selected data decreases, which means that the selected data can well represent all data.

At the end of the training process in Fig. 7, the increment of

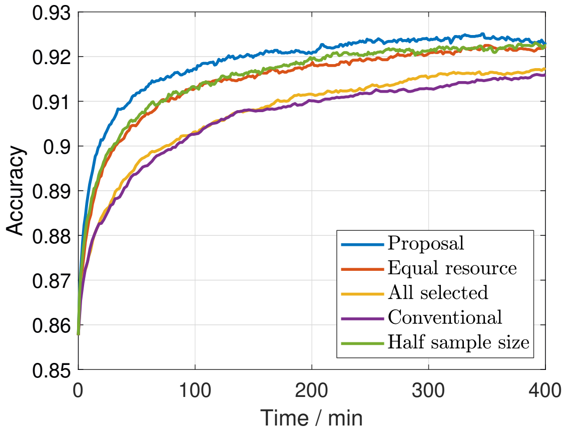

Fig. 9. The test accuracy of different schemes.

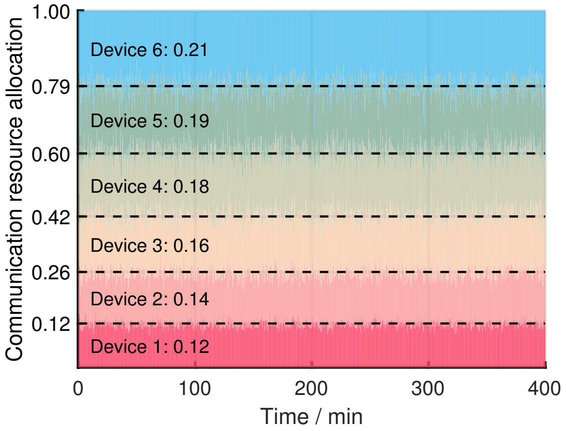

Fig. 10. An example of communication resource allocation with the proposed scheme. The boundary between adjacent areas represents instantaneous communication resource allocation result whereas the dashed line represents the average resource allocated for each device.

learning accuracy becomes small but the algorithm still needs long training time to converge. Therefore, we consider a pre-trained ResNet18 that has achieved an initial accuracy of 86%. Then, we plot the learning accuracy curve in Fig. 9. Here, the learning rate is set to be 0.001 for achieving higher accuracy. From Fig. 9, the proposed scheme can achieve its peak learning accuracy fast and is the best one among all schemes. By comparing the *all selected scheme* with our proposed scheme, one can clearly see the gain obtained by selecting important data. By comparing with the *equal resource scheme*, the gain obtained by optimal communication resource allocation can be clearly seen. Besides, the accuracy increment speed of the *half sample size scheme* is lower than that of our proposed scheme, which verifies Theorem 3. Furthermore, after 250 minutes of training, our proposed scheme almost achieves the peak learning accuracy, 0.923. However, in the same time, the *conventional scheme* only has an accuracy of 0.91 and still needs 150 minutes of training to converge to an accuracy of 0.916, which validates the performance improvement of the proposed scheme. Finally, the schemes with data selection (i.e., the proposed scheme, the *half sample size scheme*, and the *equal resource scheme*) achieve higher learning accuracy than those without data selection, demonstrating that importance-aware data selection can indeed improve the learning accuracy.

To analyze the influence of wireless channel fading, we finally illustrate the communication resource allocation for each device in Fig. 10. We should note that the communication resource allocation is mainly determined by the channel gain according to (26b). As we have mentioned before, the channel gain is mainly determined by the large-scale fading. Therefore, device 1 is allocated with the least average communication resource due to its shortest distance to the BS. Meanwhile, since the channel gain is influenced by small-scale fading, the resource allocation varies with time, as can be observed in the figure.

VI. Conclusion

In this paper, we propose a joint data selection and resource allocation scheme based on the data importance in the FEEL system to improve the learning efficiency. The relation between the gradient norm and loss decay is first analyzed, which suggests us to measure the data importance by its gradient norm. Afterwards, we formulate a learning efficiency maximization problem by jointly considering the wireless resource allocation and data selection. For given sample size and end-to-end latency, the optimal data selection and communication resource allocation policy is derived in closed-form. Based on this, the optimal end-to-end latency is obtained by the Golden-section search algorithm. Furthermore, by analyzing the relation between the learning efficiency and the sample size, we observe that each device should sample as many data as possible in the forward propagation step for learning performance improvement. Finally, test results verify the generalization ability of the proposed scheme and show that our proposal can accelerate the training process and improve the learning accuracy.

Appendix A
Proof of Lemma 1

First, we can rewrite $f_n(B_n)$ as the pointwise minimum of linear functions, as

$$f(B_n) = \min\left\{h_{n,1}(B_n), h_{n,2}(B_n), \cdots, h_{n,B_n^{\mathrm{F}}}(B_n)\right\}, \qquad 0 \leq B_n \leq B_n^{\mathrm{F}}, \quad (32)$$

where $h_{n,B_m}(B_n) = \sigma_{n,\lceil B_m \rceil}\left(B_n - \lfloor B_m \rfloor\right) + \sum_{m=1}^{\lfloor B_m \rfloor} \sigma_{n,m}$, $0 \leq B_n \leq B_n^{\mathrm{F}}$. Note that $\lceil \cdot \rceil$ represents the ceil operation and $\lfloor \cdot \rfloor$ represents the floor operation. According to [33], the pointwise minimum $f_n(B_n)$ is concave since $h_{n,1}(B_n), h_{n,2}(B_n), \cdots, h_{n,B_n^{\mathrm{F}}}(B_n)$ are all linear.

Appendix B
Proof of Theorem 1

Since problem $\mathcal{P}2$ is convex, we can utilize the Lagrangian method to solve it and the partial Lagrange function is given in (22). Then, based on the KKT conditions, we can obtain the following necessary and sufficient conditions, as

$$\frac{\partial L}{\partial B_n^*} = -\frac{\partial f_n\left(B_n^*\right)}{\partial B_n^*} \cdot \frac{1}{T} + \mu_n^*$$

$$\begin{cases} \geq 0, & B_n^* = B_n^{\min}, \\ = 0, & B_n^{\min} \leq B_n^* \leq B_n^{\mathrm{F}}, \quad \forall n \in \mathcal{N}, \\ \leq 0, & B_n^* = B_n^{\mathrm{F}}, \end{cases} \tag{33}$$

$$\frac{\partial L}{\partial \tau_n^*} = -\mu_n^* \frac{V}{R_n^{\mathrm{U}}(\tau_n^*)^2} + \lambda^* \begin{cases} \geq 0, & \tau_n^* = 0, \\ = 0, & 0 \leq \tau_n^*, \end{cases} \forall n \in \mathcal{N}, \tag{34}$$

$$\mu_n^* \left(T^{\mathrm{M,F}} + B_n^* t_n^{\mathrm{B}} + \frac{V}{\tau_n^* R_n^{\mathrm{U}}} + T^{\mathrm{D}} - T \right) = 0, \ \forall n \in \mathcal{N}, \tag{35}$$

$$\lambda^* \left(\sum_{n=1}^{N} \tau_n^* - 1 \right) = 0, \ \lambda^*, \mu_n^* \geq 0. \tag{36}$$

Note that we have fitted the loss decay function $f_n(B_n)$ as shown in Fig. 3 and the subgradient of $f_n(B_n)$ is

$$\frac{\partial f_n(B_n)}{\partial B_n} \begin{cases} = \sigma_{n,\lceil B_n \rceil}, & \text{if } B_n \notin \mathcal{Z}, \\ \in [\sigma_{n,B_n+1}, \sigma_{n,B_n}], & \text{otherwise}, \end{cases} \tag{37}$$

where $\mathcal{Z}$ is the set of non-negative integers. With simple mathematical calculation, we can derive three cases about the optimal data selection strategy as follows

1) If $g_n(B_n^{\min}) < \lambda^*$, $B_n^* = B_n^{\min}$;
2) If $g_n(B_n^{\max}) > \lambda^*$ and $B_n^{\max} = \min \left(B_n^{\mathrm{F}}, \frac{T - T^{\mathrm{M,F}} - T^{\mathrm{D}}}{t_n^{\mathrm{B}}} \right)$, $B_n^* = B_n^{\max}$;
3) If $g_n(B_n^{\min}) \geq \lambda^*$ and $g_n(B_n^{\max}) \leq \lambda^*$, $B_n^* = \phi_n(\lambda^*)$, where $\phi_n(x)$ is the inverse function of $g_n(B_n)$ in the interval $[B_n^{\min}, B_n^{\max}]$.

According to the above three cases, we can derive the optimal data selection strategy as shown in (26a). Furthermore, the resource allocation policy achieves the optimum when

$$T^{\mathrm{M,F}} + B_n^* t_n^{\mathrm{B}} + \frac{V}{\tau_n^* R_n^{\mathrm{U}}} + T^{\mathrm{D}} = T, \quad \forall n \in \mathcal{N}. \tag{38}$$

Therefore, τ_n^* is given by

$$\tau_n^* = \left(\frac{V}{\left(T - T^{\mathrm{M,F}} - T^{\mathrm{D}} - B_n^* t_n^{\mathrm{B}}\right) R_n^{\mathrm{U}}} \right)^+, \tag{39}$$

which ends the proof.

Appendix C
Proof of Theorem 2

It is easy to prove that problem $\mathcal{P}2$ is a concave-convex fractional programming problem. Then, we can transform $\mathcal{P}2$ into a convex problem via the Charnes-Cooper transformation [34] by introducing the following auxiliary variables

$$z = \frac{1}{T}, \quad q_n = \frac{B_n}{T}, \quad x_n = \frac{\tau_n}{T}. \tag{40}$$

Now, the problem $\mathcal{P}2$ can be rewritten as

$$\mathcal{P}3: \min_{\{q_n, x_n, z\}} \ -z \left(\sum_{n=1}^{N} f_n \left(\frac{q_n}{z} \right) \right), \tag{41a}$$

$$\text{s.t.} \quad zT^{\mathrm{M,F}} + q_n t_n^{\mathrm{B}} + \frac{Vz^2}{R_n^{\mathrm{U}} x_n} + zT^{\mathrm{D}} \leq 1, \forall n \in \mathcal{N}, \tag{41b}$$

$$\sum_{n=1}^{N} x_n \leq z, \tag{41c}$$

$$B_n^{\min} z \leq q_n \leq B_n^{\mathrm{F}} z, \ \forall n \in \mathcal{N}, \tag{41d}$$

$$q_n, x_n, z \geq 0. \tag{41e}$$

Given z (or T), the optimal value obtained by optimizing q_n and x_n in $\mathcal{P}3$ is exactly equal to $E(T)$. Since problem $\mathcal{P}3$ is convex, $E(T)$ is also convex with z, i.e., $\frac{1}{T}$. Therefore, we can conclude that $E(T)$ is a strictly unimodal function with T when T is bigger than zero since $E(T) > 0$. This ends the proof.

Appendix D
Proof of Lemma 3

To prove this lemma, we consider the following two cases.

Case A: In this case, there is no data selected for backward propagation and each device's latency is given by

$$T_n = T^{\mathrm{M,F}} + \frac{V}{\tau_n R_n^{\mathrm{U}}} + T^{\mathrm{D}}, \ \forall n \in \mathcal{N}, \tag{42}$$

where τ_n satisfies $\sum_{n=1}^{N} \tau_n \leq 1$. Then, by optimizing τ_n, we can obtain the minimal total latency as

$$T_{\min} = T^{\mathrm{M,F}} + \sum_{n=1}^{N} \frac{V}{R_n} + T^{\mathrm{D}}. \tag{43}$$

Since $T_{\min}$ is the minimal total latency when no data is selected, it can be regarded as a lower bound of T^*.

Case B: The maximal total latency would happen when all data are selected to join the backward propagation. In this case, the loss decay achieves its maximum. Then, to maximize the learning efficiency, the total latency should be minimized by optimizing τ_n. By allocating equal communication resource to all devices, we can obtain the total latency as

$$T_{\max} = T^{\mathrm{M,F}} + \max_{n \in \mathcal{N}} \left\{ B_n^{\mathrm{F}} t_n^{\mathrm{B}} + \frac{VN}{R_n^{\mathrm{U}}} \right\} + T^{\mathrm{D}}, \tag{44}$$

which can be regarded as an upper bound of T^*.

Appendix E
Proof of Theorem 3

The expectation of learning efficiency can be expressed as

$$\mathbb{E}\{E\} = \frac{\sum_{n \in \mathcal{N}} \mathbb{E}\{f_n(B_n)\}}{T}. \tag{45}$$

We mark two different values of $T^{\mathrm{M,F}}$ as $T^{\mathrm{M,F,(1)}}$ and $T^{\mathrm{M,F,(2)}}$ that satisfy $T^{\mathrm{M,F,(1)}} < T^{\mathrm{M,F,(2)}}$ and the corresponding sample sizes are $\{B_n^{\mathrm{F,(1)}}\}$ and $\{B_n^{\mathrm{F,(2)}}\}$, respectively. According to (13), we have

$$T^{\mathrm{M,F,(2)}} = \frac{B_n^{\mathrm{F,(2)}}}{B_n^{\mathrm{F,(1)}}} T^{\mathrm{M,F,(1)}} - \frac{B_n^{\mathrm{F,(2)}} - B_n^{\mathrm{F,(1)}}}{B_n^{\mathrm{F,(1)}}} T^{\mathrm{M}}, \forall n \in \mathcal{N}. \tag{46}$$

Let $\{B_n^{*,(1)}, \tau_n^{*,(1)}\}$ denote the optimal solution to problem $\mathcal{P}2$ when $T^{\mathrm{M,F}} = T^{\mathrm{M,F,(1)}}$. Then, when $T^{\mathrm{M,F}} = T^{\mathrm{M,F,(2)}}$, a feasible solution to problem $\mathcal{P}2$ is $\left\{ \frac{B_n^{*,(1)} B_n^{\mathrm{F,(2)}}}{B_n^{\mathrm{F,(1)}}}, \frac{\tau_n^{*,(1)} B_n^{\mathrm{F,(1)}}}{B_n^{\mathrm{F,(2)}}} \right\}$.

According to (29), the expectation of $f_n \left(\frac{B_n^{*,(1)} B_n^{\mathrm{F,(2)}}}{B_n^{\mathrm{F,(1)}}} \right)$

is equal to $\frac{B_n^{\mathrm{F},(2)}}{B_n^{\mathrm{F},(1)}}\mathbb{E}\left\{f_n\left(B_n^{*,(1)}\right)\right\}$ since the selected data's proportion is $\frac{B_n^{*,(1)}}{B_n^{\mathrm{F},(1)}}$ for $T^{\mathrm{M,F,(1)}}$ and $T^{\mathrm{M,F,(2)}}$. Since $T^{*,(1)} = T^{\mathrm{M,F,(1)}} + t_n^{\mathrm{B}} B_n^{*,(1)} + \frac{V}{\tau_n^{*,(1)} R_n^{\mathrm{U}}} + T^{\mathrm{D}},\ \forall n \in \mathcal{N}$ according to Theorem 1, we have the following equation

$$\begin{aligned} T^{(2)} &= T^{\mathrm{M,F,(2)}} + t_n^{\mathrm{B}} \frac{B_n^{*,(1)} B_n^{\mathrm{F},(2)}}{B_n^{\mathrm{F},(1)}} + \frac{V B_n^{\mathrm{F},(2)}}{\tau_n^{*,(1)} R_n^{\mathrm{U}} B_n^{\mathrm{F},(1)}} + T^{\mathrm{D}} \\ &= \frac{B_n^{\mathrm{F},(2)}}{B_n^{\mathrm{F},(1)}} T^{*,(1)} - \frac{B_n^{\mathrm{F},(2)} - B_n^{\mathrm{F},(1)}}{B_n^{\mathrm{F},(1)}} \left(T^{\mathrm{D}} + T^{\mathrm{M}}\right),\ \forall n \in \mathcal{N}. \end{aligned} \tag{47}$$

Then, the corresponding objective value is given by

$$\begin{aligned} E^{(2)} &= \frac{\mathbb{E}\left\{\sum_{n\in\mathcal{N}} f_n\left(\frac{B_n^{*,(1)} B_n^{\mathrm{F},(2)}}{B_n^{\mathrm{F},(1)}}\right)\right\}}{T^{(2)}} \\ &= \frac{\sum_{n\in\mathcal{N}} \frac{B_n^{\mathrm{F},(2)}}{B_n^{\mathrm{F},(1)}} \mathbb{E}\left\{f_n\left(B_n^{*,(1)}\right)\right\}}{T^{(2)}} \\ &= \sum_{n\in\mathcal{N}} \frac{\mathbb{E}\left\{f_n\left(B_n^{*,(1)}\right)\right\}}{T^{*,(1)} - \frac{B_n^{\mathrm{F},(2)} - B_n^{\mathrm{F},(1)}}{B_n^{\mathrm{F},(2)}} \left(T^{\mathrm{D}} + T^{\mathrm{M}}\right)} \\ &> \sum_{n\in\mathcal{N}} \frac{\mathbb{E}\left\{f_n\left(B_n^{*,(1)}\right)\right\}}{T^{*,(1)}} = E^{*,(1)}, \end{aligned} \tag{48}$$

where $E^{*,(1)}$ is the optimal value when $T^{\mathrm{M,F}} = T^{\mathrm{M,F,(1)}}$. Let $E^{*,(2)}$ denote the optimal value when $T^{\mathrm{M,F}} = T^{\mathrm{M,F,(2)}}$. Then we have $E^{(2)} \leq E^{*,(2)}$. Combining this with (48), we can conclude that $E^{*,(2)} > E^{*,(1)}$ when $T^{\mathrm{M,F,(2)}} > T^{\mathrm{M,F,(1)}}$. In conclusion, the expectation of learning efficiency increases with $T^{\mathrm{M,F}}$.

Furthermore, the limit of learning efficiency can be given by

$$\begin{aligned} \lim_{T^{\mathrm{M,F}}\to\infty} \mathbb{E}\left\{E^*\right\} &= \lim_{T^{\mathrm{M,F}}\to\infty} \sum_{n=1}^{N} \frac{\frac{T^{\mathrm{M,F}}}{t_n^{\mathrm{F}}} q_n C_n}{\frac{T^{\mathrm{M,F}}}{t_n^{\mathrm{F}}} q_n t_n^{\mathrm{B}} + T^{\mathrm{M,F}}} \\ &= \sum_{n=1}^{N} \frac{q_n C_n}{q_n t_n^{\mathrm{B}} + t_n^{\mathrm{F}}}. \end{aligned} \tag{49}$$

This ends the proof.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[2] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning," *IEEE Trans. Comput.*, vol. 66, no. 11, pp. 1946-1960, Nov. 2017.

[3] S. Zhang, J. Liu, H. Guo, M. Qi, and N. Kato, "Envisioning device-to-device communications in 6G," *IEEE Netw.*, vol. 34, no. 3, pp. 86–91, May. 2020.

[4] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security and intelligence," *IEEE Wireless Commun.*, early access, doi: 10.1109/MWC.001.1900516.

[5] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies towards 6G," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 96–103, Jun. 2020.

[6] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.

[7] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Towards an intelligent edge: Wireless communication meets machine learning," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.

[8] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, early access, doi: 10.1109/COMST.2020.2986024.

[9] J. Konečný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint* arXiv:1511.03575, 2015.

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

[11] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," *arXiv preprint* arXiv:1911.00856, 2019.

[12] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[13] Q. Zeng, Y. Du, K. K. Leung, and K. Huang, "Energy-efficient radio resource allocation for federated edge learning," *arXiv preprint* arXiv:1907.06040, 2019.

[14] Y. Sun, S. Zhou, and D. Gündüz, "Energy-aware analog aggregation for federated learning with redundant data," *arXiv preprint* arXiv:1911.00188, 2019.

[15] N. H. Tran, W. Bao, A. Y. Zomaya, N. N. H. Minh, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 1387–1395.

[16] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *arXiv preprint* arXiv:1911.02417, 2019.

[17] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *arXiv preprint* arXiv:1909.07972, 2019.

[18] J. Ren, G. Yu, and G. Ding, "Accelerating DNN training in wireless federated edge learning system," *arXiv preprint* arXiv:1905.09712, 2019.

[19] A. Katharopoulos and F. Fleuret, "Biased importance sampling for deep neural network training," *arXiv preprint* arXiv:1706.00043, 2017.

[20] T. Gao and V. Jojic, "Sample importance in training deep neural networks," In *Proc. Int. Conf. Learn. Representations (ICLR)*, Toulon, France, Apr. 2017.

[21] G. Alain, A. Lamb, C. Sankar, A. Courville, and Y. Bengio, "Variance reduction in SGD by distributed importance sampling," *arXiv preprint* arXiv:1511.06481, 2015.

[22] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, to appear.

[23] D. Liu, G. Zhu, J. Zhang, and K. Huang, "Wireless data acquisition for edge learning: Importance aware retransmission," In *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Cannes, France, Jul. 2019, pp. 1–5.

[24] D. Liu, G. Zhu, J. Zhang, and K. Huang, "Data-importance aware user scheduling for communication-efficient edge machine learning," *arXiv preprint* arXiv:1910.02214, 2019.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[26] A. Katharopoulos and F. Fleuret, "Not all samples are created equal: Deep learning with importance sampling," *arXiv preprint* arXiv:1803.00942, 2018.

[27] 3GPP, "LTE; Evolved universal terrestrial radio access (E-UTRA); Physical channels and modulation (3GPP TS 36.211 version 15.6.0 Release 15)," *TS 36 211 V15.6.0*, Jul. 2019.

[28] A. Polino, R. Pascanu, and D. Alistarh. "Model compression via distillation and quantization," In *Proc. Int. Conf. Learn. Representations (ICLR)*, Vancouver, Canada, Apr. 2018.

[29] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[30] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication–efficient distributed learning," in *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, Dec. 2018.

[31] J. Kiefer, "Sequential minimax search for a maximum," In *Proc. Amer. Math. Soc.*, 1953, pp. 502–506.

[32] H. N. Nagaraja, “An introduction to extreme order statistics and actuarial applications,” In *ERM Symposium*, Chicago, Apr. 2004.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[34] K. Shen and W. Yu, “Fractional programming for communication systems–Part I: power control and beamforming,” *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, May 2018.