Towards Energy-efficient Federated Edge Intelligence for IoT Networks

Qu Wang*, Yong Xiao*†, Huixiang Zhu*, Zijian Sun*, Yingyu Li*, and Xiaohu Ge*
*School of Elect. Inform. & Commun., Huazhong Univ. of Science & Technology, Wuhan, China
†Pazhou Lab, Guangzhou, China

***Abstract*—Federated edge intelligence (FEI) is an emerging framework that implements federated learning (FL)-based learning solutions in an edge networking and computing system. It has attracted significant interest due to its potential to enable machine learning (ML)-based smart services and applications in the next-generation wireless systems. Despite its potential, the environmental impact of implementing energy-consuming ML-based solutions in a large networking system has been considered as one of the major challenges for the sustainability of future digital networking systems. This paper proposes energy-efficient FEI (EE-FEI), an energy-efficient framework, that jointly optimizes multiple key parameters to minimize the overall energy consumption of an FEI-supported Internet of Things (IoT) network. We establish models to quantify the relationship between the total energy consumption of FEI and the key parameters including the number of edge servers, the number of local model training rounds, and the number of global coordination rounds. We formulate the energy consumption minimization problem and prove its approximation problem is biconvex. Alternate Convex Search (ACS) algorithm for solving the key parameters to minimize the energy consumption of an FEI system has been used. Finally, we evaluate our theoretical results using a hardware prototype. Numerical results have shown that EE-FEI can significantly reduce the energy consumption of FEI systems by 49.8%.**

***Index Terms*—Energy efficient, federated learning, edge Intelligence, IoT network.**

I. Introduction

The increasing popularity of IoT devices has resulted in exponential growth in the volume of data generated by wireless networking systems. According to a recent report [1], the global mobile data traffic is expected to increase over 100 times from 2020 to 2030, reaching around 163 Zettabytes in 2030. It is expected that the data transportation and computational resources required for emerging data-driven services and applications will soon exceed those of today's wireless networks. In particular, the traditional centralized data processing architecture in which data samples collected by all the IoT devices must be uploaded into a remote cloud data center for centralized processing is now viewed as the major obstacle for supporting high-performance ML-based applications requiring complex model training, fast service response, and privacy protection.

Recently, edge intelligence has been promoted as one of the key solutions to support ML-based smart services with stringent requirements [2]. By deploying a large number of decentralized edge servers to perform data processing and model training closer to IoT devices, edge intelligence has the potential to significantly reduce the communication overhead and improve service responsiveness. To address the need for decentralized data processing across multiple edge servers, federated edge intelligence (FEI) has been introduced to implement federated learning (FL), an emerging distributed learning framework, for training and constructing ML models over edge servers based on decentralized datasets [3]. FEI allows multiple edge servers to collaboratively train a model without exposing their local data and therefore can further improve communication efficiency and offer privacy protection for the local data samples.

Since FL is essentially a distributed coordination framework for implementing ML algorithms, it suffers from the high implementation cost and resource consumption as most existing ML solutions. It has been observed that the resource consumption for training the state-of-the-art deep learning models is doubled every few months, resulting in a total increase of 300,000 times from 2012 to 2017 [4]. There is still lacking a comprehensive framework to optimize the energy consumption of an FL-based networking system.

Recent reports observe that the resource consumption of FL-based algorithm is closely related to several key parameters including the number of edge servers participating in each round of global model coordination, the number of local computational steps performed by each edge server as well as the total number of global coordination rounds required to achieve a certain model accuracy. Although there are already existing works investigating the impact of these parameters, most of these works focus on optimizing a single parameter that could only affect the data processing and computation at edge servers [5]. In addition to the computational resources, the model training performance at the edge servers is also closely related to the data collecting and uploading capability of the IoT network. There is a pressing need to develop a comprehensive framework to model and optimize the energy consumption of an FEI network system involving both computation and communication related energy consumption.

Motivated by the above observations, in this paper, we study the multi-parameter optimization problem for minimizing the total energy consumption of an FEI-supported IoT network. Compared to the single parameter optimization problem that only focuses on minimizing the resource consumption of edge servers, the above problem is much more complex due to the following reasons: (1) IoT devices and edge servers are fundamentally different devices and their energy consumptions

are affected by different subsets of parameters, (2) it is known that different parameters are closely related to each other and different combinations of these parameters may result in different convergence performance as well as energy consumptions of the IoT networks and edge servers, yet there is still no analytical solution that is able to capture the relationship of different parameters as well as their impact on the overall system resource-consumption, and (3) although some recent works are focusing on the resource consumption of FL, an extensive measurement study based on a practical prototype is still lacking.

To address these challenges, we propose a novel framework called energy-efficient FEI (EE-FEI) that can jointly optimize multiple key parameters to minimize the overall energy consumption of training a model with desirable accuracy. In particular, we formulate mathematical models to characterize the energy consumption of every step of FEI and quantify the impact of three key parameters on the total energy consumption of it, including (1) the number of edge servers K to participate in each round of global model coordination, (2) the number of local model training rounds E in each round of global model coordination, and (3) the total number of global coordinations T required to train the model with a target accuracy level. We utilize ACS algorithm as the solution for K, T, and E to minimize the total energy consumption. Finally, we develop a hardware prototype with 20 Raspberry Pis and conduct extensive measurement studies on the energy consumption under different combinations of parameters. To the best of our knowledge, this is the first work that leverages the ACS algorithm as the solution for approximately optimizing multiple key parameters to minimize the energy consumption of FEI-based networking systems.

The main contributions of this paper are summarized as follows:

- **Energy Consumption Modeling:** We propose an energy-efficient framework, referred to as EE-FEI, for FEI-supported IoT networks. We formulate energy consumption models concerning K, T, and E for each step of FEI. Based on the formulated model, we analyze and quantify the impact of key parameters on the energy consumption of an FEI system.
- **Optimal Solution:** We formulate the energy consumption minimization problem and prove that the objective function of the problem is biconvex with respect to K and E, and take ACS algorithm as the solution for K, E, and T to minimize the energy consumption of FEI.
- **Prototype Development and Simulation:** We develop a hardware prototype and conduct extensive measurements for the energy consumption of FEI under different setups and combinations of parameters. Numerical results have shown that EE-FEI can reduce the energy consumption by 49.8% and achieve an optimal trade-off between computational load per edge server and communication overhead.

The rest of this paper is organized as follows. Existing works that are relevant to this paper are reviewed in Section II. We introduce our system model in Section III and formulate the energy consumption model of FEI in Section IV. The energy consumption optimization algorithm is proposed in Section V. Experiment results are presented in Section VI, and Section VII concludes this paper.

II. Related Work

Optimization of Federated Learning: FL has been introduced as a solution for distributed learning, which is more communication-efficient compared with mini-batch SGD. Most existing works in FL focus on investigating the trade-off between communication cost and convergence performance [6]–[8]. In particular, in [6], a K-step averaging SGD algorithm was developed to minimize the communication overhead and its convergence result was also established for non-convex objectives. The convergence rate of FL was derived in [7] on convex problems. In [8], an upper bound of convergence for FL was derived and utilized to develop a control algorithm to minimize the loss function under certain resource constraints.

Energy Model of Federated Learning: The environmental impact of implementing energy-consuming ML-based solutions in a large networking system has been considered as one of the major challenges for the sustainability of future networking systems. However, the energy consumption of federated learning has not been well explored. The authors in [9] developed a deep reinforcement learning-based solution for the joint optimization of training time and energy consumption. An iterative algorithm with low complexity was proposed in [10] to minimize total energy consumption while maintaining the requirements of latency. In [11], the authors studied the trade-off between convergence and energy consumption. An online energy-aware dynamic edge server scheduling policy was proposed in [12] to maximize the average number of edge servers participating in a single iteration with energy constraint.

III. System Model and Problem Formulation

In this section, we first introduce the main components of the federated edge intelligence (FEI)-based IoT system and then present the energy consumption model for each step of the model training process.

A. System model

An FEI system consists of the following components as shown in Fig.1:

(1) **IoT Network:** consists of a large number of IoT devices deployed across a wide geographical area. Each IoT device collects local data samples to be uploaded to its designated edge server.
(2) **Edge Servers:** correspond to mini-computational servers deployed close to the IoT devices to provide data storage and model training service.
(3) **Coordinator:** coordinates the model training processes among all the edge servers. It can be deployed at the cloud data center or one of the edge servers.

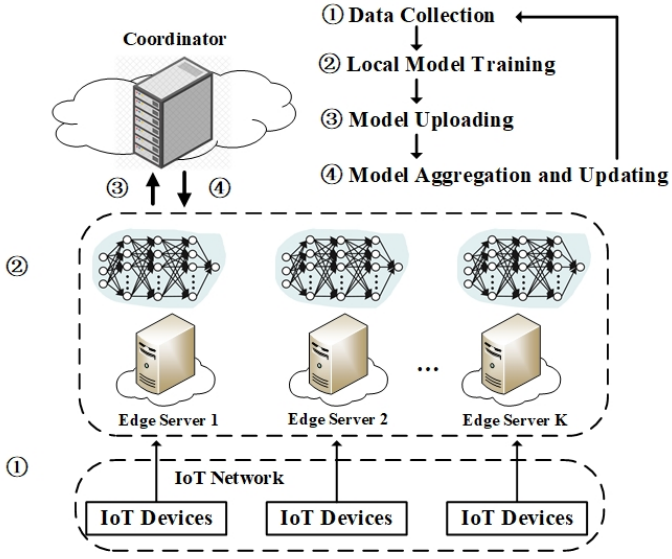


Fig. 1. System model and training procedure.

Due to the privacy concerns and the data transportation limit of the communication channel, edge servers cannot share their datasets with each other nor send any data samples to the coordinator. They can however collaborate with each other to train a shared machine learning model by following the FL procedures. In this paper, we assume all the edge servers adopt FedAvg, one of the most popular FL-based algorithms, to coordinate their model training process.

Suppose the FEI system consists of a set $\mathcal{K}$ of edge servers trying to train a shared machine learning model over T rounds of global coordination. Let $N := |\mathcal{K}|$, where $|\cdot|$ means the size of set. We use subscript t to denote the model training related parameters between the t-th and the $(t+1)$-th round of global model coordination, i.e., let $\omega_{k,t}$ be the locally trained model parameters of edge server k and ω_t be the globally updated model sent by the coordinator at the beginning of the t-th round of coordination. The algorithm procedure is illustrated in Fig.1. The main steps in each round t are described as follows:

(1) **Data Collection:** Each IoT device first uploads the requested number of data samples to its associated edge server k. Suppose edge server k requests n_k number of data samples from its local IoT devices. We use $x_{k,t}^{(i)}$ to denote the i-th data sample to be sent to edge server k in the t-th round of coordination. Let $\left\langle x_{k,t}^{(i)} \right\rangle_{i\in\{1,\cdots,n_k\}}$ be the sequence of data samples arrived at edge server k.

(2) **Local Model Training:** A subset $\mathcal{K}_t$ of edge servers will be randomly selected in each round t to participate in each round of model updating, for $\mathcal{K}_t \subseteq \mathcal{K}$. Once being selected, each edge server will receive a global parameter ω_t sent by the coordinator. Every selected edge server k will start performing E rounds (epochs) of local stochastic gradient descent (SGD) based on parameter ω_t and the data samples from the IoT network using the local loss function, denoted as

$$F_k(\omega_t) := \frac{1}{n_k}\sum_{j=1}^{n_k} l\left(\omega_t; x_{k,t}^{(j)}\right), \tag{1}$$

where $l\left(\omega_t; x_{k,t}^{(j)}\right)$ is the loss function over each data sample $x_{k,t}^{(j)}$.

(3) **Model Uploading:** After E rounds of local training, each edge server k obtains an updated local model parameter $\omega_{k,t}$ which will then be uploaded to the coordinator.

(4) **Model Aggregation and Updating:** After receiving updated model parameters from all the selected edge severs, the coordinator calculates the updated global model parameters by performing model aggregation as follows:

$$\omega_{t+1} \leftarrow \frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\omega_{k,t}, \tag{2}$$

where $\mathcal{K}_t$ is the subset of edge servers selected to participate in the FL training task in the t-th round of update, and $K := |\mathcal{K}_t|$ is the cardinality of the set.

B. Energy Consumption Analysis of FEI

The energy consumption of each step of the FEI during each round of global model coordination can be modeled as follows:

(1) **Energy Consumption of Data Collection**: is dominated by the energy consumed by IoT devices in transmitting the required number of data samples. We use $e_k^I(n_k)$ to denote the amount of energy consumed by a set of IoT devices for uploading n_k data samples to edge server k.

(2) **Energy Consumption of Local Model Training**: is mainly affected by the energy consumption of local computation and data processing performed by each edge server. Previous works as well as our measurements suggest that the energy consumption for the local model updating at each edge server k is closely related to several key parameters including the number of local epochs E and the size of the local dataset n_k. Let $e_k^P(E, n_k)$ be the energy consumption of edge server k to train E local epochs over a local dataset of size n_k.

(3) **Energy Consumption of Model Uploading**: is characterized by the energy used by the edge servers to transmit the local model to the coordinator. The total energy consumption of model uploading is closely related to the subset of edge servers being selected to participate in each round of model coordination, i.e., let e_k^U be the energy consumed by edge server k to upload its locally trained model. We can write the total energy consumed by FEI for model uploading as $e^U = \sum_{k\in\mathcal{K}_t} e_k^U$. Note that each edge server k only consumes energy for model uploading after E rounds of local training when it has been selected by the coordinator to participate in the global model coordination.

In this paper, we focus on energy minimization problem of FEI which can be written as follows:

$$\min_{E,K,T} \{\mathbb{E}\left[e(E,\boldsymbol{\mathcal{K}},T,\boldsymbol{n})\right]\} \tag{3a}$$

$$\text{s.t. } \mathbb{E}\left[F\left(\omega_T\right) - F(\omega_*)\right] \leq \epsilon, \tag{3b}$$

$$E,\mathcal{K}_t,T,n_k \in Z^+, \forall t \in \{1,...,T\}, k \in \mathcal{K}, \tag{3c}$$

where $\boldsymbol{\mathcal{K}} = \langle \mathcal{K}_t \rangle_{t\in\{1,...T\}}$, $\boldsymbol{n} = \langle\ n_k\rangle_{k\in\mathcal{K}}$, and $e(E,\boldsymbol{\mathcal{K}},T,\boldsymbol{n}) = \sum_{t=1}^{T}\sum_{k\in\mathcal{K}_t}(e_k^I\left(n_k\right) + e_k^P\left(E,n_k\right) + e_k^U)$. $F(\omega_*)$ is the minimum value of the loss function, and $\mathbb{E}[F(\omega_T) - F(\omega_*)]$ is the gap between the loss value after T rounds of global model coordinations and the global minimum loss value. ϵ is the target model accuracy.

It can be observed that solving problem (3a) requires analytical solutions that can characterize the relationship between $e(E,\boldsymbol{\mathcal{K}},T,\boldsymbol{n})$ and all the parameters $E,\boldsymbol{\mathcal{K}},T,\boldsymbol{n}$, as well as the target accuracy level ϵ, most of which are unavailable. In the rest of this paper, we first establish energy consumption models for every step of FEI and then adopt the convergence result from the existing literature to derive the analytical solution that can capture the relationship between all the above key parameters and the performance metrics.

IV. Energy Consumption Modeling of FEI

In this section, we discuss the possible energy model for each step of the FEI.

A. Energy Model of Data Collection

This part of energy is mainly consumed by the IoT network to collect and upload data samples to the edge servers. Since most IoT devices adopt passive sensors to record data samples, we can ignore the energy consumption of data collection. It is known that IoT devices are mostly low-cost devices without complex energy adaptation or channel equalization schemes, we can therefore assume each IoT device consumes the same amount of energy for uploading a fixed-size data sample. For example, NB-IoT consumes 7.74 mWs ($mW \cdot s$) per byte to send data packets. For some IoT technologies operating in the unlicensed band, the data uploading may suffer some data packet loss due to the transmission collision caused by simultaneous data transmission of multiple IoT devices. Recent results have shown that as long as the location of all the IoT devices can be assumed to be fixed, the probability of successful data uploading can also be regarded as a fixed value for each IoT device, i.e., the average energy consumed by an IoT device in the unlicensed band for uploading each data sample can also be assumed to be a constant. We assume the data samples uploaded from all the IoT devices to each edge server have equal size and follow the i.i.d. distribution. We can therefore write the energy consumption for all the associated IoT devices to upload n_k number of data samples to edge server k as

$$e_k^I\left(n_k\right) = \rho_k n_k, \tag{4}$$

where ρ_k is the normalized energy consumption for IoT devices to upload each data sample.

B. Energy Model of Local Model Training

Previous studies and our own measurement observe a linear relationship between $e_k^P\left(E,n_k\right)$ and the values of E and n_k [13]. In particular, we consider the FL with synchronized coordination among edge servers and assume each edge server performs the same number E of local computation steps. n_k is the mini-batch size of each edge server. It has been observed in [3] that the computational load scales almost linearly as the mini-batch size, i.e., we can write the energy consumption for each local round of model training at edge server k as $e_k^l = c_0 n_k + c_1$, where c_0 characterizes the energy consumed for computing each data sample and c_1 is the constant capturing the energy consumption that is unrelated to the computational load, i.e., stationary energy for most computing devices. We will give a more detailed discussion and provide empirical values of c_0 and c_1 in Section VI. Based on the above discussion, we can write the energy consumption of local model training at edge server k as

$$e_k^P\left(E,n_k\right) = c_0 E n_k + c_1 E, \tag{5}$$

Based on the above assumption, we can rewrite the energy consumption minimization problem in Eq.(3) as

$$\min_{E,K,T} \{\mathbb{E}\left[\hat{e}(E,K,T)\right]\} \tag{6a}$$

$$\text{s.t. } \mathbb{E}\left[F\left(\omega_T\right) - F(\omega_*)\right] \leq \epsilon, \tag{6b}$$

$$E,K,T,n_k \in Z^+, \quad \text{and} \quad 1 \leq K \leq N, \tag{6c}$$

where $\hat{e}(E,K,T)$ is given by $\hat{e}(E,K,T) = \sum_{t=1}^{T}\sum_{k\in\mathcal{K}_t}\left(\rho_k n_k + c_0 E n_k + c_1 E + e_k^U\right)$.

V. Energy Consumption Optimization Algorithm

In this section, we develop an optimization algorithm to solve problem (6a). We adopt an existing convergence rate upper bound to merge the convergence constraint in (6b) into objective function. We then propose a distributed algorithm to optimize K and E.

A. Convergence Constraint

Multiple theoretical convergence results have been reported in the existing literature. In this paper, we adopt the convergence solution from [14] due to the following reasons: (1) it has been claimed that the convergence rate proposed in [14] is the tightest compared to other solutions; (2) the convergence result covers several extreme scenarios as special cases including the mini-batch SGD ($E = 1$) and one-shot SGD ($T = 1$); and (3) it does not require the strong assumption such as a bounded dissimilarity of local gradients across different edge servers.

In [14, Theorem 4], it reports the following convergence results.

Proposition 1: Suppose the loss function of each edge server F_k is μ-convex and L-smooth, i.e., for all $x,y \in \mathbb{R}^d$, we have $\frac{\mu}{2}||x-y||^2 \leq F_k(x) - F_k(y) - \langle \bigtriangledown F_k(y), x-y\rangle \leq \frac{L}{2}||x-y||^2$. For any $E,K,T \in Z^+$, we have

$$\mathbb{E}[F(\bar{\omega}_T) - F(\omega_*)] \leq \frac{\alpha_0||\omega_0 - \omega_*||^2}{\gamma T E} + \frac{\alpha_1\gamma\sigma^2}{K} + \alpha_2\gamma^2 L\sigma^2(E-1), \tag{7}$$

where α_0, α_1, α_2 are constants, $||\omega_0 - \omega_*||^2$ is the distance between initial point and optimal point, $\bar{\omega}_T = \frac{1}{TE}\sum_{t=1}^{TE}\hat{\omega}_t$, where $\hat{\omega}_t$ is the local updated model, γ is the learning rate and σ is the metric measuring the variance of stochastic gradients at the optimum, defined as $\sigma^2 \triangleq \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{z_k\sim\mathcal{Z}_k}[||\bigtriangledown f(x_*, z_k)||^2]$, where $\mathcal{Z}_k$ is the distribution of samples at edge server k.

We have the following proposition:

Proposition 2: It has been proved that if all local models gradually reach consensus, then the FL can always converge to the loss minimization solution. We assume that for any $1 \le t_1 < t_2 \le T$, we have $\mathbb{E}[F(\omega_{t_1}) - F(\omega_*)] \ge \mathbb{E}[F(\omega_{t_2}) - F(\omega_*)]$. Then the average accuracy level over T rounds is always smaller than the expected precision at the T-th round of coordination, i.e., we have

$$\mathbb{E}[F(\bar{\omega}_T) - F(\omega_*)] \ge \mathbb{E}[F(\omega_T) - F(\omega_*)]. \tag{8}$$

Proof 1: From assumption in Proposition 2, we have

$$\begin{aligned}\mathbb{E}[F(\bar{\omega}_T) - F(\omega_*)] &= \mathbb{E}[\frac{1}{TE}\sum_{t=1}^{TE}F(\omega_t) - F(\omega_*)] \\ &\ge \mathbb{E}[\frac{1}{TE}TEF(\omega_T) - F(\omega_*)] = \mathbb{E}[F(\omega_T) - F(\omega_*)].\end{aligned} \tag{9}$$

This concludes the proof.

Substituting (7) into (9), we have

$$\frac{A_0}{TE} + \frac{A_1}{K} + A_2(E-1) \le \epsilon, \tag{10}$$

where A_0, A_1 and A_2 are constants given by $A_0 = \alpha_0||\omega_0 - \omega_*||^2/\gamma$, $A_1 = \alpha_1\gamma\sigma^2$ and $A_2 = \alpha_2\gamma^2L\sigma^2$. Then, in order to minimize the energy consumption, we need to reduce the global training rounds T as much as possible under the constraint of convergence. By rearranging inequality (10), we can get the constraint of T on ϵ and obtain the optimal T^* as follows:

$$T^* = \frac{A_0K}{[\epsilon K - A_1 - A_2K(E-1)]E}, \tag{11}$$

Substituting (11) into problem (6a), we can rewrite the objective function (6a) as follows:

$$\mathbb{E}[\hat{e}(K,E)] = \frac{A_0K}{[\epsilon K - A_1 - A_2K(E-1)]E}K\left(B_0E + B_1\right), \tag{12}$$

where $B_0 = \mathbb{E}(c_0)n_k + \mathbb{E}(c_1)$ and $B_1 = \mathbb{E}(\rho_k)n_k + \mathbb{E}[e_k^U]$.

In the rest of this paper, we follow the commonly adopted setting [14] and focus on optimizing the upper bound as an approximation of the energy consumption of FEI. We can therefore rewrite problem (6a) as follows:

$$\min_{E,K}\left\{\mathbb{E}\left[\hat{e}(E,K)\right]\right\} \tag{13a}$$

$$\text{s.t. } E, K, T, n_k \in Z^+, \quad \text{and} \quad 1 \le K \le N, \tag{13b}$$

$$\epsilon K - A_1 - A_2K(E-1) > 0. \tag{13c}$$

B. Solving the Energy Consumption Optimization Problem

As mentioned earlier, in FEI, all the edge servers will wait for the global model parameters as well as the model training setup information from the coordinator. In other words, the coordinator will first decide how many edge servers to select to participate in the next round of global model coordination. Actually, many existing results [5] have already shown that increasing K can always result in acceleration of convergence, e.g., reduced number of required global coordination to reach the target accuracy level. However, increasing K could also cause a higher energy consumption for FEI. In other words, there exists a fundamental trade-off between convergence and energy consumption. The existence of an optimal solution of K with a given E is proved as follows.

Lemma 1: For a fixed E, objective function (13a) is a strictly convex function of K.

Proof: For a fixed $E \ge 1$, we have

$$\frac{\partial^2\mathbb{E}[\hat{e}]}{\partial^2K} = \frac{2A_0A_1^2C_0}{(C_1K - A_1)^3} > 0, \tag{14}$$

where $C_0 = (B_0E + B_1)/E > 0$ and $C_1 = \epsilon - A_2(E-1) > 0$. Since the second-order partial derivative of the objective function with respect to K is positive and also the domain of K is convex, we can claim that the objective function (13a) is a convex function of K. This concludes the proof. ■

To find the optimal value of K, we use $\frac{\partial\mathbb{E}[\hat{e}]}{\partial K} = 0$ and derive the optimal value K^* as follows:

$$K^* = \begin{cases}\frac{2A_1}{\epsilon - A_2(E-1)}, & \frac{A_1}{\epsilon - A_2(E-1)} \in \{1, N\} \\ 1, & \frac{A_1}{\epsilon - A_2(E-1)} < 1 \\ N, & \frac{A_1}{\epsilon - A_2(E-1)} > N\end{cases} \tag{15}$$

where A_1 and A_2 are defined in (10).

Similarly, existing solutions have proved that increasing the value of E will reduce the required number of global coordination rounds to reach a fixed accuracy, resulting in a reduction of energy consumed for communication between edge servers and the coordinator. However, E directly affects the local computation load at each edge server. Therefore, E is closely related to the trade-off between the energy consumption of communication and local computation. We prove the following result which will lead to the optimal solution of E.

Lemma 2: For a fixed K, objective function (13a) is a strictly convex function of E.

Proof 2: For a fixed $N \ge K \ge 1$, we have

$$\begin{aligned}\frac{\partial^2\mathbb{E}[\hat{e}]}{\partial^2E} = \frac{2A_2^2C_2K^2}{(C_4 - A_2KE)^3} + \frac{2A_2C_3K}{(C_4 - A_2KE)^2E^2} + \\ \frac{2C_3(C_4 - 2A_2KE)^2}{(C_4 - A_2KE)^3E^3} > 0,\end{aligned} \tag{16}$$

where $C_2 = A_0B_0K^2 > 0$, $C_3 = A_0B_1K^2 > 0$ and $C_4 = \epsilon K - A_1 + A_2K > 0$.

Since the second-order partial derivative of the objective function with respect to E is always positive and the domain

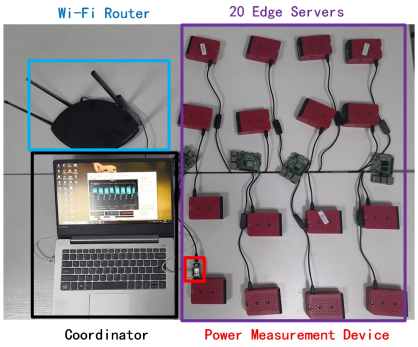


Fig. 2. Hardware prototype with the power measurement device (KM001C), laptop as coordinator, 20 Raspberry Pi as edge servers and the WiFi Router.

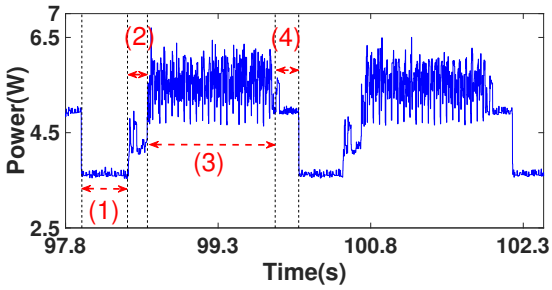


Fig. 3. Power consumption of an edge server during two rounds of global model coordination: we can observe different power consumptions at different steps of local model training and updating in each round including (1) Waiting (for data uploading), (2) (global) Model Downloading, (3) Local Model Training, and (4) Local Model Uploading.

of K is convex, we can claim that the objective function (13a) is a convex function for a fixed K. This concludes the proof.

By letting $\frac{\partial \mathbb{E}[\hat{e}]}{\partial E} = 0$ we can derive the optimal E^* as follows:

$$E^* = \begin{cases} \frac{(\epsilon K - A_1 + A_2 K)B_1 - A_2 B_0 K}{2A_2 B_1 K}, & \frac{(\epsilon K - A_1 + A_2 K)B_1 - A_2 B_0 K}{2A_2 B_1 K} \geq 1 \\ 1, & \frac{(\epsilon K - A_1 + A_2 K)B_1 - A_2 B_0 K}{2A_2 B_1 K} < 1 \end{cases} \tag{17}$$

Combining Lemma (1) and Lemma (2) directly leads to the following result.

Theorem 1: Objective function (13a) is a strictly biconvex function.

The biconvexity of problem (13a) allows us to adopt Alternate Convex Search (ACS) [15] to iteratively achieve the optimal solution for both K^* and E^*. We first set the search domain as $\mathcal{Z}_K \in \{\max\left(\frac{A_1}{\epsilon - A_2(E-1)}, 1\right), N\}$ and $\mathcal{Z}_E \in \{1, \frac{\epsilon K - A_1 + A_2 K}{A_2 K}\}$ based on (13c). When the difference of the calculated values obtained by two successive iterations is less than the target residual ξ, the solution is considered to be approximately optimal. The detailed algorithm is presented in Algorithm 1.

Algorithm 1 Parameter Optimization Algorithm

Input: Target residual ξ; $i = 0$; Initial point (K_0, E_0); Search domains $\mathcal{Z}_K$ and $\mathcal{Z}_E$.
Output: Solution point (K, E).
While $|\mathbb{E}[\hat{e}(K_i, E_i)] - \mathbb{E}[\hat{e}(K_{i+1}, E_{i+1})]| > \xi$ **do**
  Step 1: Substitute E_i into (15) to calculate K^*.
  Step 2: Substitute K_i into (17) to calculate E^*.
  Step 3: Set $i = i + 1$.

VI. Experimental Results

In this section, we develop a hardware prototype to measure the energy consumption of FEI and then conduct extensive simulations to evaluate the performance of our proposed optimization algorithm.

A. Experimental Setup

We develop a hardware prototype system consisting of N = 20 Raspberry Pis 4B mini-computers as edge servers as shown in Fig. 2. We use a laptop computer as the coordinator. Edge servers and the coordinator are connected via a TP-Link WiFi Router, and we use a multi-function USB multi-meter POWER-Z KM001C plugged into the power port of each Raspberry Pi to measure and keep track of voltage, current, and power during model training. We set the power sample rate to 1 kHz.

We conduct our experiments on a widely used dataset named MNIST containing 70,000 28×28 pixel gray-scale images (60,000 for training and 10,000 for testing) of hand-written digits. We adopt multinomial logistic regression as the loss function to train a classification model shared among edge servers.

We uniformly allocate 60,000 training data samples into 20 edge servers (each has 3000 samples stored in its local memory). We use full batch size for SGD and set the learning rate as 0.01 with a fixed decay rate of 0.99.

B. Preliminary Observations

We record data traces of energy consumption at each edge server during 100 rounds of global model coordination. We can observe the same pattern being repeated at each round of training. We can also observe a clear difference in the energy consumption at different steps of local training and model uploading at the edge server.

Since our experiments are based on dataset pre-loaded to each edge server, the local model training and uploading consist of the following four steps:

(1) **Waiting:** In FEI, each edge server will first wait for the global model and training parameters from the coordinator as well as data samples uploaded from the IoT network. In our experiment, we assume the dataset has been pre-loaded to the edge server. Therefore, in this first step, the edge server will not perform any data loading or computing tasks. The average power consumption in this step see in step (1) in Fig. 3 is 3.6W which is almost the same as that of the idle state of a Raspberry Pi.

(2) **Model Downloading:** Once the coordinator dispatches the global model parameters and model training setup to edge servers, each edge server will first download these model-related data and then replace its local model parameters with the global one, which can be reflected by

the two peaks in power consumption at the beginning of step (2) in Fig. 3. The average power consumption during the entire step (2) is around 4.286W.

(3) **Local Model Training:** Once an edge server receives all the required model and datasets, it will start to perform E rounds of local training. Surprisingly, we observe that the instantaneous power consumption does not increase with the values of E of n_k. In Table I we present the time duration of step (3) under different combinations of E and n_k. We can observe that the time duration of each local training step increases almost linearly with n_k. The average energy consumption during this step is around 5.553W.

(4) **Local Model Uploading:** When E steps of local model training have been finished, the edge server will upload the updated local model parameters to the coordinator. The power consumption in this step is recorded in (4) of Fig. 3. We can observe that the average power consumed for local model uploading is around 5.015W which is less than the computational power consumption for training the local model (step (3)) and higher than the power consumption consumed by model downloading (step (2)).

The above four-step power consumption pattern is repeated at every round of global model updating.

TABLE I
TIME DURATION OF STEP (3) UNDER DIFFERENT TRAINING PARAMETERS

E	n_k	Time of step (3)(sec)
10	100	0.0197
10	500	0.0749
10	1000	0.1471
10	2000	0.2855
20	100	0.0403
20	500	0.1508
20	1000	0.2912
20	2000	0.5721
40	100	0.0799
40	500	0.3026
40	1000	0.5554
40	2000	1.1451

Based on the above analysis, we observe that the energy consumption recorded in our experiments are consistent with the energy consumption model previously developed in Eq. (5) and we can fit our measured data in Table I to obtain the estimated values of c_0 and c_1 as $7.79 * 10^{-5}$ and $3.34 * 10^{-3}$, respectively using the least square method.

C. Numerical Results

Here we present our experimental results. Our simulation configuration is listed in Table II.

TABLE II
SIMULATION CONFIGURATION

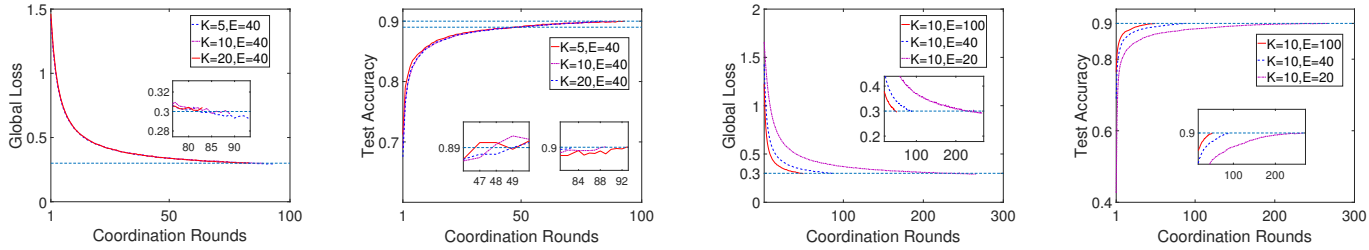
Model Type	Multinomial Logistic Regression
Input Size	784*1
Output Size	10*1
Activation Function	Sigmoid
Optimizer	SGD, learning rate 0.01 with decay rate 0.99

Fig. 4 presents the global loss value and test accuracy at different global coordination rounds with different combinations of E and K.

Fixed E: In Fig. 4(a) and Fig. 4(b), we fix E = 40 and compare the convergence performance of FEI under different K. In Fig. 4(a), we can observe that the global loss value drops dramatically in the first few rounds of global coordination (around T = 25), and then the convergence speed reduces when the value of T continues to increase. Fig. 4(b) compares test accuracy under the number of global coordination rounds with different values of K. We observe that when the required model accuracy level is relatively low (e.g., around 0.89), increasing K does not significantly affect the required value of T to reach the target accuracy (shown in the middle-left subfigure). However, when the required model accuracy becomes more stringent (e.g., around 0.9), increasing K results in a linear reduction of the required value of T to obtain the target accuracy, which is shown in the middle-right subfigure.

Fixed K: As shown in Fig.4(c) and Fig. 4(d), we fix K = 10 and compare the convergence performance of FEI under different E. In both figures, we can observe that E is closely related to the trade-off between communication and local computation at edge servers. It has already been proved that the convergence of FL is almost the same as that of mini-batch SGD (e.g., E = 1) in terms of the total gradients calculated by edge servers, which is determined by $E \cdot T$ for a given K. However, our result suggests that the above result does not apply when evaluating the combined energy consumption involving both communication and computation at edge servers. In particular, suppose the target accuracy level is 0.9, we can observe that when E = 20, the required value of T is given by 280, resulting in 5,600 rounds of local gradients being calculated in total to reach the target accuracy. When E increases to 40, the required value of T to reach the target accuracy is around 90, resulting in 3,600 local gradient rounds. If we continue to increase E to 100, we can observe that the required value of T becomes 60, which again results in a total of 6,000 rounds of local gradients being calculated by edge servers. This verifies the existence of an optimal value of E to achieve the minimized energy consumption of an FEI system.

In Fig. 5 and 6, we compare the theoretical bound of energy consumption in (13a) with the real traces collected in our experiments. We also highlight the optimal values K^* and E^* calculated using both the theoretical bound as well as real traces collected in our hardware prototype. Note that since we compare the total amount of energy consumed for training the model with a fixed accuracy level of 92% (calculated by power times the duration of the entire model training process), different combinations of E and K will result in different number of required global coordination rounds T. We can observe that although there is a gap between real traces and the theoretical bound, the theoretical bound curve shows the same trend as the real traces. In particular, in Fig. 5, we can observe that the optimal solution K^* is 1 which means that only one edge server needs to be selected to participate in each round of global coordination. This is because, in



(a) Loss with different K. (b) Accuracy with different K. (c) Loss with different E. (d) Accuracy with different E.

Fig. 4. Training performance with multinomial logistic regression and MNIST.

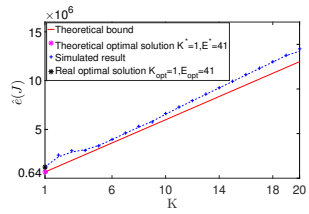


Fig. 5. Energy consumption estimated from theoretical bound (solid line) compared with real measurement traces (dash line) with the optimal solution K^* calculated using theoretical bound (red asterisk) compared with that obtained using practical traces (black asterisk).

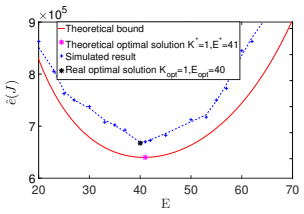


Fig. 6. Energy consumption estimated from theoretical bound (solid line) compared with real measurement traces (dash line) with the optimal solution E^* calculated using theoretical bound (red asterisk) compared with that obtained using practical traces (black asterisk).

our setup, we assume the distribution of data samples in all the edge servers is identical. In other words, the gradients calculated using datasets at different edge severs should show similar statistic features. In this case, our result shows that choosing one edge server to upload its locally trained model is a more communication-efficient solution. Similarly, in Fig. 6, we present the energy consumption of the optimal solution E^* compared with that of other E values. Note that there is a slight difference between the optimal E^* calculated using our theoretical result and real traces. This can be caused by the roundup operation applied in our theoretical bound where we use a continuous optimization solution to approximate the discrete value of E. We can also observe that by optimizing the value of E, our proposed EE-FEI can achieve around 49.8% reduction of energy consumption, compared to the case with $K = 1$ and $E = 1$.

VII. Conclusion

This paper studied an FEI-supported IoT network. An energy-efficient FL-based framework called EE-FEI was proposed to jointly optimize the number of model training participating edge servers K and the number of local model training rounds E to minimize overall energy consumption for training a satisfactory ML model. We investigated the energy consumption in each step of the model training process and proved the energy consumption minimization problem was biconvex. We then derived close-form optimal solutions for K, E, and T to solve our optimization problem. A hardware prototype has been developed to verify the performance of our proposed solution. Experiment results showed that our theoretical results were consistent with real energy measurement results and EE-FEI could reduce the energy consumption by 49.8%.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62071193, the National Key R & D Program of China under Grant No. 2020YFB1806605, and the Key R & D Program of Hubei Province of China under Grant No. 2020BAA002.

References

[1] N. Alliance, "5g white paper," *Next generation mobile networks, white paper*, vol. 1, Feb. 2015.

[2] Y. Xiao, G. Shi, Y. Li, W. Saad, and H. V. Poor, "Towards self-learning edge intelligence in 6G," *IEEE Communications Magazine*, vol. 58, no. 12, Dec. 2020.

[3] Y. Xiao, Y. Li, G. Shi, and H. Vincent Poor, "Optimizing resource-efficiency for federated edge intelligence in iot networks," in *WCSP*, Nanjin, China, Oct. 2020.

[4] R. Schwartz, J. Dodge, N. Smith, and O. Etzioni, "Green ai," *arXiv preprint arXiv:1907.10597*, 2019.

[5] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," *arXiv preprint arXiv:2012.08336*, 2020.

[6] F. Zhou and G. Cong, "On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization," *arXiv preprint arXiv:1708.01012*, 2017.

[7] S. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[8] 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 JSAC*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[9] Y. Zhan, P. Li, and S. Guo, "Experience-driven computational resource allocation of federated learning by deep reinforcement learning," in *IEEE IPDPS*, New Orleans, LA, Mar. 2020.

[10] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[11] C. T. Dinh, N. H. Tran, M. N. H. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 398–409, Feb. 2021.

[12] Y. Sun, S. Zhou, and D. Gündüz, "Energy-aware analog aggregation for federated learning with redundant data," in *IEEE ICC*, Dublin, Ireland, Jun. 2020.

[13] M. Perrone, H. Khan, C. Kim, A. Kyrillidis, and J. Quinn, "Optimal mini-batch size selection for fast gradient descent," *arXiv preprint arXiv:1911.06459*, 2019.

[14] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local sgd on identical and heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*, Virtual Conference, Aug. 2020.

[15] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Math Methods Oper. Res.*, vol. 66, no. 3, pp. 373–407, Jun. 2007.