Optimizing Resource-Efficiency for Federated Edge Intelligence in IoT Networks

Yong Xiao*, Yingyu Li*, Guangming Shi†, and H. Vincent Poor‡
*School of Elect. Inform. & Commun., Huazhong Univ. of Science & Technology, China
†School of Artificial Intelligence, Xidian University, Xi'an, China
‡School of Engineering & Applied Science, Princeton University, Princeton, NJ
E-mail: {yongxiao, liyingyu}@hust.edu.cn, gmshi@xidian.edu.cn, poor@princeton.edu

***Abstract*—This paper studies an edge intelligence-based IoT network in which a set of edge servers learn a shared model using federated learning (FL) based on the datasets uploaded from a multi-technology-supported IoT network. The data uploading performance of IoT network and the computational capacity of edge servers are entangled with each other in influencing the FL model training process. We propose a novel framework, called federated edge intelligence (FEI), that allows edge servers to evaluate the required number of data samples according to the energy cost of the IoT network as well as their local data processing capacity and only request the amount of data that is sufficient for training a satisfactory model. We evaluate the energy cost for data uploading when two widely-used IoT solutions: licensed band IoT (e.g., 5G NB-IoT) and unlicensed band IoT (e.g., Wi-Fi, ZigBee, and 5G NR-U) are available to each IoT device. We prove that the cost minimization problem of the entire IoT network is separable and can be divided into a set of subproblems, each of which can be solved by an individual edge server. We also introduce a mapping function to quantify the computational load of edge servers under different combinations of three key parameters: size of the dataset, local batch size, and number of local training passes. Finally, we adopt an Alternative Direction Method of Multipliers (ADMM)-based approach to jointly optimize energy cost of the IoT network and average computing resource utilization of edge servers. We prove that our proposed algorithm does not cause any data leakage nor disclose any topological information of the IoT network. Simulation results show that our proposed framework significantly improves the resource efficiency of the IoT network and edge servers with only a limited sacrifice on the model convergence performance.**

***Index Terms*—6G, edge intelligence, federated learning, IoT.**

I. Introduction

Data is at the heart of next generation IoT networks. With the dramatic growth in demand for smart services and applications that rely on cross-domain data processing and analysis, data privacy has become one of the key issues when deploying data-driven AI-based solutions in large-scale IoT networks. Traditional cloud-based solution in which IoT devices must upload all the data into a cloud data center for a centralized process is known to be difficult to meet the increasingly stringent requirements in service responsiveness as well as data security and privacy protection.

Edge intelligence has been recently promoted as one of the key solutions to unleash the full potential of data-driven IoT networks. It inherits the decentralized nature of edge computing and emphasizes more on performing and delivering AI-functions and solutions closer to the data source, e.g., IoT devices. One of the key challenges for applying edge intelligence in IoT networks is to implement distributed data processing and learning across a large number of decentralized datasets owned or managed by different user devices with privacy protection requirements. Federated learning (FL) is an emerging distributed AI framework that enables collaborative machine learning (ML) across decentralized datasets. It offers a viable solutions for data-driven learning and model training across privacy-sensitive datasets.

Despite of its huge potential, implementing FL in edge intelligence-based IoT networks is hindered by several challenges. In particular, edge intelligence relies on a large number of low-cost edge servers to perform AI tasks and solutions. The geographical distributions and data processing capabilities of edge servers can vary significantly causing highly unbalanced service demands and resource utilization. Also, the quality and size of dataset available at each edge server directly affects the performance of the model training. Unfortunately, the random nature of the wireless links connecting IoT devices and edge servers can exhibit high temporal and spatial fluctuation which may result in different numbers of data samples being uploaded from the IoT network to different edge servers. Currently, there is still a lack of a comprehensive framework that can coordinate and jointly optimize the data collection and transportation of the IoT network and the data processing and model training at edge servers.

In this paper, we investigate an edge intelligence-based IoT network in which an edge computing system consisting of a set of edge servers trains a shared model based on the data samples uploaded from an IoT network. We propose a novel framework, referred to as *federated edge intelligence (FEI)*, to allow jointly optimization of IoT networks and edge servers. In FEI, each edge server will only request a limited amount of data to be collected and uploaded from the IoT network as long as the requirement of model can be met. We consider a multi-technology-supported IoT network in which each IoT device can use licensed band IoT (LB-IoT) (e.g., NB-IoT) as well as unlicensed band IoT (UB-IoT) (e.g., Wi-Fi, 5G NR-U, Zigbee, LoRa, etc.) technologies to upload data. The main objective is to maximize the utilization of computational resources of edge servers and at the same time minimize energy cost of the IoT network. We propose a distributed algorithm that can address both problems. More specifically, to minimize the energy cost of IoT network, we evaluate the energy cost for uploading data samples with both LB-IoT and UB-IoT and prove that the energy cost minimization problem of the IoT network is separable and can be divided into a set of subproblems each can be solved by an individual edge server. For the edge computing system, we introduce a mapping function to quantify the convergence performance of model training under different combinations of three key parameters: size of the dataset, local batch size, and number of local training passes. We finally adopt an Alternative Direction Method of Multipliers (ADMM)-based approach to jointly optimize both energy cost minimization problem of the IoT network and computing resource utilization of the edge computing system. We prove that our proposed algorithm does not cause data leakage of any local dataset nor reveal any network-related information such as topological information about the IoT network. Numerical results show that our proposed framework can significantly reduce energy cost of the IoT network and at the same time improve the resource utilization of the edge computing system with only a slight degradation of the convergence performance of FL model training.

The remainder of this paper is organized as follows. Existing work that is relevant to this paper is reviewed in Section II. We introduce our system model and formulate the joint optimization problem in Section III. The FEI framework and a distributed optimization algorithm are proposed in Section IV. Numerical results are presented in Section V and the paper is concluded in Section VI.

II. Related Work

Edge computing-based IoT: Edge computing has already been recognized as one of the key components in the next generation IoT network to perform low-latency data processing and analysis. Most existing works focus on optimizing the communication resource (e.g., spectrum) allocation of IoT and computational load distribution of edge computing. For example, in [1], the authors studied the joint allocation of computing and networking resources for application provisioning in edge computing-enabled IoT networks requiring QoS guarantees. The task offloading and resource allocation problems for distributed edge computing networks have also been investigated under various time-varying scenarios [2], [3].

FL for Networking Systems: As mentioned earlier, two major challenges that are obstructing the wide spread implementation of FL in wireless networks are heterogeneity of datasets and highly unbalanced data processing capabilities between edge servers. Many recent works have already introducing solutions to address each of these challenges. For example, solutions such as user data sharing [4] and user selection for model updating [5] have been shown to be effective to reduce the adverse effect caused by the first challenge under certain scenarios. To address the second challenge, the authors in [6] have proposed to allow discarding and offloading of data among edge servers to balance the computational load among edge servers.

In this paper, we focus on addressing both of the above challenges simultaneously by investigating the relationship between the energy cost minimization of IoT network and resource utilization optimization of edge servers. We observe that these two problems are closely related to each other. A distributed and privacy-preserving framework that can optimize both of these problems is proposed. To the best of our knowledge, this is the first work to investigate the joint optimization and coordination of data collection and uploading of IoT and the computational resource utilization of edge servers for edge intelligence-based IoT.

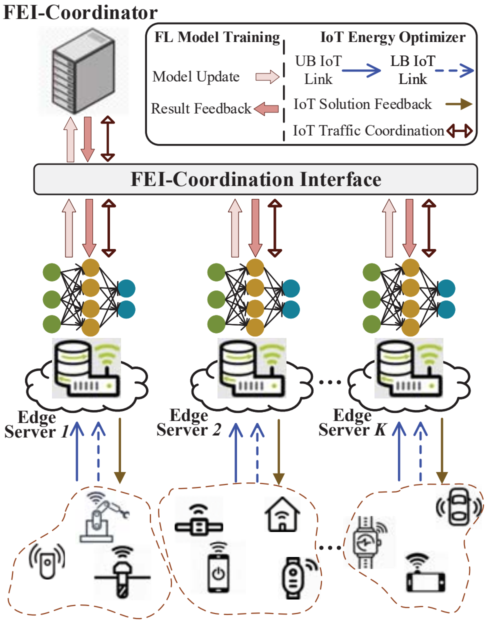

Fig. 1. System model for federated edge intelligence.

III. System Model and Problem Formulation

We propose federated edge intelligence (FEI) for an IoT network to upload data samples to a set $\mathcal{K}$ of edge servers for training a share model, as shown in Figure 1. The data uploading configuration of the IoT network and the data processing and training capability of the edge computing system are entangled with each other to influence the accuracy and convergence of the model training process.

A. IoT Network

Let $\mathcal{M}$ be the set of IoT devices in the considered IoT network. Suppose the data collected by each IoT device consists of propriety information and therefore must be encrypted and uploaded to its associated edge server. Let $\mathcal{M}_k$ be the set of IoT devices associated with edge server k for $k \in \mathcal{K}$ that satisfies $\mathcal{M}_k \subseteq \mathcal{M}$ and $\cup_{k\in\mathcal{K}}\mathcal{M}_k = \mathcal{M}$. We assume each IoT device can adopt two types of widely adopted IoT solutions, *LB-IoT* and *UB-IoT*, to upload its data to the associated edge server.

LB-IoT including 5G NB-IoT and its evolution towards B5G and 6G. In LB-IoT, each IoT device can pay for a dedicate channel in the licensed spectrum from a mobile network operator (MNO) to send its data samples. Let β_m be the price charged by the MNO to send each data sample from IoT device m to the associated edge server. We consider energy-sensitive IoT network and let ρ_m be the cost for IoT device m to consume each unit of energy in data uploading. This energy cost can be related to the real money spent by the IoT service operator to provide energy supplies for IoT nodes, e.g., the cost of replacing the battery or the electricity price charged by the utility company. Suppose each IoT device m uploads n''_m data samples in LB-IoT. The total cost for the IoT devices to upload the required data samples to edge server k can then be written as

$$c^l_k\left(\boldsymbol{n}''_k\right) = \sum_{m\in\mathcal{M}_k} \left(\rho_m \nu^l_m\left(n''_m\right) + \beta_m n''_m\right), \tag{1}$$

where $\boldsymbol{n}''_k = \langle n''_m\rangle_{m\in\mathcal{M}_k}$ and $\nu^l_m(n''_m)$ is the total number of energy units required for IoT device m to upload n''_m data samples using LB-IoT.

UB-IoT including 5G New Radio-Unlicensed (NR-U), LoRa and ZigBee as well as their future evolution variants. In UB-IoT, all the IoT devices share the same unlicensed band for free. In this case, each IoT device only needs to pay for the energy cost for data uploading. We can write the cost for uploading $\boldsymbol{n}'_k = \langle n'_m\rangle_{m\in\mathcal{M}_k}$ data samples from set $\mathcal{M}_k$ of IoT devices to edge server k as

$$c^u_k(\boldsymbol{n}'_k) = \sum_{m\in\mathcal{M}_k} \rho_m \nu^u_m(n'_m), \tag{2}$$

where $\nu^u_m(n'_m)$ is the number of energy units consumed by IoT device m using UB-IoT. Note that, since the unlicensed band is open to various wireless technologies, successful data delivery cannot always be guaranteed due to the possibility of collision between multiple coexisting devices.

The main objective for IoT network is to solve the following problem.

(P1) Energy Cost Minimization Problem: Each edge server has limited computational resource and can only process a limited number of data samples during a given time duration. If the volume of received data exceeds the computational capacity of an edge server, IoT network can reduce the data collection and uploading rate to reduce costs. Based on this observation, in our framework, edge servers will first evaluate their required size of dataset (the number of local data samples) according to the model training requirement. Each edge server will then decide the suitable data uploading configuration for its associated IoT devices including the number of data samples sent from each IoT device in each available IoT solution (LB-IoT or UB-IoT). Motivated by the fact that, in most large IoT networks, managing the energy efficiency of the entire network is more important than minimizing the energy cost of an individual IoT device, we focus on minimizing the total energy cost of an IoT network. Suppose there is the maximum energy constraint ϵ (the maximum number of energy units) for the IoT network, i.e., we have $\nu(\boldsymbol{n}', \boldsymbol{n}'') \le \epsilon$ where $\nu(\boldsymbol{n}', \boldsymbol{n}'') = \sum_{m\in\mathcal{M}} \left(\nu^u_m(\boldsymbol{n}') + \nu^l_m(\boldsymbol{n}'')\right)$ is the total energy consumption of the IoT network. We can then write the energy cost minimization problem of IoT network as

$$\textbf{(P1)} \quad \min_{\boldsymbol{n}', \boldsymbol{n}''} \quad c(\boldsymbol{n}', \boldsymbol{n}'') = \sum_{k\in\mathcal{K}} \left(c^u_k(\boldsymbol{n}'_k) + c^l_k(\boldsymbol{n}''_k)\right) \tag{3a}$$

$$\text{s.t.} \quad \nu(\boldsymbol{n}', \boldsymbol{n}'') \le \epsilon \;\text{ and }\; n'_m + n''_m \le \bar{n}_m, \tag{3b}$$

where $\boldsymbol{n}' = \langle n'_m\rangle_{m\in\mathcal{M}}$, $\boldsymbol{n}'' = \langle n''_m\rangle_{m\in\mathcal{M}}$, and $\bar{n}_m$ is the maximum number of data samples that can be collected by each IoT device m.

B. Edge Computing System

Recent results [7], [8] as well as our own experiments have suggested that for a given type of dataset (e.g., MNIST, CIFAR, etc.), the computational load in each edge server k to train an FL model is closely related to three key parameters: (1) size of the local dataset, denoted as n_k, (2) the number of local training passes in each coordination round, denoted as e_k, and (3) the minibatch size of each local training pass, denoted as b_k. Motivated by this observation, we introduce an empirical-based mapping function μ_k to quantify the relationship between computational load and combinations of parameters e_k, b_k, and n_k under a given model convergence requirement, i.e., there exists a computational load function for each edge server k given by $\tilde{\mu}_k = \mu_k\left(b_k, e_k, n_k | t, r\right)$ where t denotes the type of dataset and r is the convergence requirement of the model training (minimum required model accuracy under a given number of rounds of model updating). We will give a more detailed discussion in Section IV.

We investigate the following problem for edge computing system.

(P2) Resource Utilization Maximization of Edge Computing System: Let $\bar{\mu}_k$ be the maximum computational load that can be supported by the local computational resources of edge server k, i.e., we have $\tilde{\mu}_k < \bar{\mu}_k$. Suppose the number of data samples arrived at edge server k is given by $n_k = \sum_{m\in\mathcal{M}_k} \left(n'_m + n''_m\right)$. We can then define the average resource utilization maximization problem for an edge computing system as

$$\textbf{(P2)} \quad \min_{\boldsymbol{b}, \boldsymbol{e}, \boldsymbol{n}} \; \mu\left(\boldsymbol{n}\right) = \frac{1}{K} \sum_{k\in\mathcal{K}} \left(\bar{\mu}_k - \tilde{\mu}_k\right) \tag{4a}$$

$$\text{s.t.} \; n_k < \tilde{n}_k \;\text{ and }\; \tilde{\mu}_k \le \bar{\mu}_k, \forall k \in \mathcal{K}, \tag{4b}$$

where $\boldsymbol{b} = \langle b_k\rangle_{k\in\mathcal{K}}$, $\boldsymbol{e} = \langle e_k\rangle_{k\in\mathcal{K}}$, $\boldsymbol{n} = \langle n_k\rangle_{k\in\mathcal{K}}$, and $\tilde{n}_k$ is the maximum number of data samples that can be received and processed by edge server k.

C. Joint Optimization of Resource Efficiency

It can be observed that problems (P1) and (P2) are closely related to each other. To investigate the joint optimization of these problems, we introduce a weighing factor γ specifying the relative weights between energy cost of the IoT network and computational resource utilization of the edge computing system. In this paper, we focus on solving the following problem:

$$\textbf{(P3)} \quad \min_{\boldsymbol{n}', \boldsymbol{n}''} \; \mu\left(\boldsymbol{n}\right) + \gamma c(\boldsymbol{n}', \boldsymbol{n}'') \tag{5a}$$

$$\text{s.t.} \quad \nu(\boldsymbol{n}', \boldsymbol{n}'') \le \epsilon \tag{5b}$$

$$n'_m + n''_m \le \bar{n}_m, \; \forall m \in \mathcal{M} \tag{5c}$$

$$n_k < \tilde{n}_k \;\text{ and }\; \tilde{\mu}_k \le \bar{\mu}_k, \forall k \in \mathcal{K}, \tag{5d}$$

Solving problem (5) requires the full knowledge of both the IoT network and edge computing system which may not always possible, especially in a distributed networking system

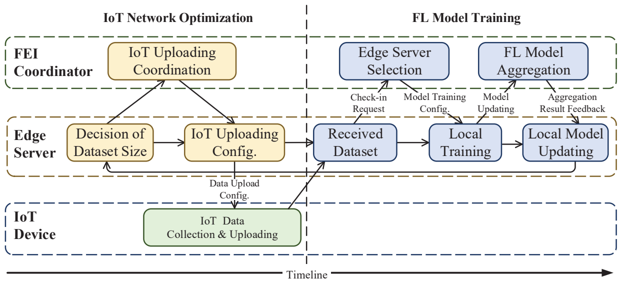

Fig. 2. Proposed framework and procedure of the FEI.

where the data information of each edge server cannot be revealed to others. In the rest of this paper, we introduce a novel distributed framework that can jointly coordinate edge servers to train a FL model across edge servers and at the same time minimize the energy cost of IoT network by optimizing the data uploading traffic distribution between UB-IoT and LB-IoT across all the IoT devices. Our framework will not disclose any data-relevant information.

IV. FEI Framework and Resource Efficiency Optimization

A. Framework Overview

The detailed procedure of the FEI framework is illustrated in Figure 2. In FEI, each edge server will first evaluate the required size of dataset to be uploaded from the IoT network. We propose a distributed coordination approach based on distributed ADMM for edge servers to coordinate via FEI coordinator and jointly decide the size of data samples to be uploaded from each IoT device. In this approach, each edge server will initially randomly generate a number of data samples to be uploaded from the IoT network. The generated number needs to satisfy the data collection constraint of IoT devices as well as its data processing limit. Edge servers will then submit their randomly selected numbers to the FEI coordinator. FEI coordinator will calculate an intermediate result based on the received numbers and send the result back to the edge servers. Each edge server will use this intermediate result to calculate an updated solution about the number of data samples to be requested from the IoT network. The above process will be repeated and the result will converge to the global optimal solution of dataset sizes to be uploaded from IoT devices with each IoT solution (UB-IoT or LB-IoT). We will give a more detailed discussion in Section IV-B).

After receiving the required number of data samples, each edge server will then proceed to the FL model training procedure. In this procedure, edge servers will first decide whether to register with the FEI coordinator to join the next round of model training. FEI coordinator will select a subset of edge servers to participate in the training and will send the detailed configuration information, such as data structure, sharing state, and model parameters, to each participating edge server. Each participating edge server will perform local training computations based on the received configuration information as well as its local dataset. After the local training, edge servers will send their updates to the FEI coordinator for model aggregation. Once FEI coordinator receives enough updates, it will send the aggregation result back to participating edge servers. The edge servers will update their respective models using the received results. The above process will be repeated in the next round with a newly selected subset of edge servers until the trained model converges nor the stopping criteria have been met.

As will be proved in Section IV-B, our proposed framework allows distributed coordination and collaborative model training between edge servers without requiring any exchange of datasets between edge servers or disclosing any information related to the IoT network including the topological information, spectrum usage, etc.

1) Computational Load and Resource Demand Mapping: We follow a commonly adopted setting [7], [8] and consider an FL process with synchronized model updating in which all the edge servers will use the data collected during the same period of time with duration τ to train a globally shared machine learning model. We assume the data samples collected by each individual IoT device at different time within each considered duration are equally important for model training. Let $\boldsymbol{x}_k$ be the dataset collected by edge server k during τ. Suppose for a given τ, the number of model updating rounds among edge servers can be considered as fixed and the model convergence performance can then be specified by the model accuracy that can be achieved under a fixed number of model updating rounds. We use r to denote the model convergence requirement, i.e., r is the minimum model accuracy required for training within a fixed number of updating rounds. As mentioned earlier, the computational load in each edge server k to train an FL model with a certain requirement r is closely related to three key parameters, including size of the local dataset n_k, the number of local training passes e_k in each coordination round, and the minibatch size b_k of each local update.

It has been shown in [7] that FL can only converge to an unbiased solution if all the edge servers have equal probability to participate in the model training. We therefore assume the probability for each edge server to participate each round of model update can be regarded as a constant, denoted as C. The value of C is closely related to the communication cost among edge servers to train the model. How to optimize the value of C is out of the scope of this paper and will be our future work. Suppose the size of dataset $\boldsymbol{x}_k$ (number of samples) is n_k. The number of local updates performed by edge server k for each round is given by $\frac{e_k n_k}{b_k}$, e.g., if each edge server uses all the received dataset for model training per pass, we have $n_k = b_k$ and, in this case, edge server k will perform e_k local passes for each coordination round. It is known that the values of e_k and b_k affect not only the computational load at edge server k, but also the convergence performance of the entire model training process. In particular, increasing e_k or decreasing b_k will generally result in more computational load at edge server k. The impact of e_k and b_k on the convergence of FL model training is more complicated and will also depend on the type of the dataset. For example, for some document recognition dataset consisting of samples with highly correlated features, such as MNIST [9], reducing b_k and increasing e_k will accelerate the convergence performance.

However, for some other types of datasets with more complex features, e.g., Shakespeare LSTM dataset [7], increasing e_k may result in diverged result due to the over-optimization. In this case, it is better to adopt a larger b_k and a smaller e_k. We present the average computational time of Federated Averaging algorithm proposed in [7] for each round of model updating using MNIST dataset under the same computation environment and setup in Table I.

To simulate the data generated by IoT network during multiple periods, we assume each edge server receives the different subsets of samples for training different FL models at different time periods. We can observe that for a given size of the dataset, different combinations of parameters result in different convergence rate and computational load, e.g., a small b_k and a large e_k achieve the fastest convergence speed with the highest computational load requirement. In this paper, we focus on a finite set $\mathcal{T}$ of dataset types for IoT networks. We assume data samples from the same type of dataset can be considered as independent and identically distributed and contribute equal to the model training. We also assume, for a given type of dataset, there exists a one-to-one mapping from each possible combination of b_k, e_k, and n_k to the computational load $\tilde{\mu}_k$ at each edge server k, i.e., we have $\tilde{\mu}_k = \mu_k\left(b_k, e_k, n_k | t, r\right)$ for $\tilde{\mu}_k < \bar{\mu}_k$, where $\bar{\mu}_k$ is the maximum computation power that can be offered by edge server k. We also assume for each given type of dataset $t \in \mathcal{T}$ with a fixed size n_k, there exists an optimal pair of b_k and e_k that offers the fastest convergence performance for FL model training, denoted as $\langle b^*_k, e^*_k \rangle = \mathcal{F}\left(n_k, t, r\right)$ for all $k \in \mathcal{K}$.

TABLE I
COMPUTATION TIME COMPARISON UNDER DIFFERENT PARAMETERS

n_k	e_k	b_k	Accuracy (in 50 rounds)	Accuracy (in 200 rounds)	Time per round (sec)
100	5	10	97.12%	98.39%	0.1216
100	5	50	95.34%	97.72%	0.0655
100	**20**	**10**	**97.22%**	**98.47 %**	**0.4772**
100	20	50	95.98%	97.81%	0.2610
200	5	10	98.00%	98.73%	0.2386
200	5	20	97.74%	98.67%	0.1693
200	5	50	96.98%	98.36%	0.1299
200	**20**	**10**	**98.04%**	**98.85 %**	**0.9571**
200	20	20	97.88%	98.74%	0.6784
200	20	50	97.60%	98.44%	0.5200
400	5	10	98.58%	99.01%	0.4775
400	5	20	98.27%	98.89%	0.3383
400	5	50	97.87%	98.78%	0.2597
400	**20**	**10**	**98.61%**	**99.16%**	**1.9128**
400	20	20	98.39%	98.97%	1.3539
400	20	50	98.22%	98.83%	1.0409

2) Energy Consumption of IoT Network: Due to the different spectrum sharing regulations, LB-IoT and UB-IoT have different characteristics in terms of spectrum access, energy consumption, as well as data uploading performance. In this paper, we adopt a commonly used formulation to model the energy consumption of IoT data uploading.

UB-IoT: Since unlicensed band is shared by all the devices without coordination, a certain channel access mechanism must be imposed to avoid collision among spectrum sharing devices. For example, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism has been adopted in 2.4 GHz and 5 GHz unlicensed bands for supporting UB-IoT such as Wi-Fi, ZigBee, Thread, Z-Wave, and Wi-SUN [10]. In this mechanism, each IoT device must first sense the vacancy of the channel and can only start transmission if the channel is sensed idle. If the channel is busy, it will go through a random backoff procedure and will only be able to send data if all the procedure is finished and the channel is clear. According to [11], [12], the probability of channel access is closely related to the topology (relative location) of all the spectrum sharing devices as well as the channel contending parameters adopted by each IoT device. If the topology and parameters of IoT devices are fixed, the probability for each IoT device to successfully occupy the channel to transmit data can be considered as fixed. Similarly, in the duty-cycle-based mechanism adopted in 433 MHz and 800 MHz bands, IoT devices do not have to sense the channel but will have to send its data packets according to a fixed duty-cycle specifying the maximum fraction of time that can be occupied by each device to access the channel, e.g., depending on the frequency bands, the maximum duty-cycle is generally between 0.1% and 10%.

Let P^u_m be the probability for IoT device m to successfully occupy unlicensed band to send data. If IoT node m needs to upload n'_m bits of data to edge server k within time duration τ, it will need to send the data at a rate of $\frac{n'_m}{\tau} = P^u_m B^u \log_2\left(1 + h^u_m \frac{\tilde{\nu}^{(1)}_m}{\sigma^u_m}\right)$ where B^u is the bandwidth of unlicensed channel, h^u_m is the channel gain between IoT node m and the associated UB-IoT access point (AP)[1], and σ^u_m is the received noise level at the AP. The energy consumed by IoT device m for sending n'_m data bits via a UB link can then be written as

$$\nu^u_m(n'_m) = \frac{\sigma^u_m}{h^u_m}\left(2^{\frac{n'_m}{\tau B^u P^u_m}} - 1\right). \tag{6}$$

LB-IoT: Each IoT device can purchase IoT connectivity service from an MNO to upload its data through the licensed band. In this case, a dedicate frequency band will be allocated to each IoT device for data transmission. Let β_m be the price paid by IoT device m for sending each bit of data via LB-IoT network infrastructure, i.e., the price paid by IoT m for sending n''_m bits of data is given by $\beta_m n''_m$. Similar to the previous case, we can write the energy consumed by IoT node m to upload n''_m bits of data to edge server k as

$$\nu^l_m(n''_m) = \frac{\sigma^l_m}{h^l_m}\left(2^{\frac{n''_m}{\tau B^l}} - 1\right), \tag{7}$$

where B^l is the bandwidth of each licensed channel, h^l_m is the channel gain between IoT node m and the associated LB-IoT base station (BS), and σ^l_m is the received noise level at the BS.

B. Resource Efficiency Optimization Algorithm Design

We propose a joint optimization algorithm to address Problem (P3) under the above framework based on an FEI coordinator. In this algorithm, we divide the global traffic allocation problem into K subproblems each of which can

[1] In this paper, we assume the major bottleneck for data transmission between IoT devices and edge servers is the wireless link connecting the IoT device and AP.

be solved by an edge server, and the intermediate calculation results will be sent to the FEI coordinator that can perform coordination among all edge servers.

To make our description clearer, we define $\boldsymbol{n}_k = \langle n'_m, n''_m \rangle_{m \in \mathcal{M}_k}$ to include all the data sample optimization variables of the IoT nodes associated to each edge server, i.e., $\boldsymbol{n}'_k = A_k \boldsymbol{n}_k$ and $\boldsymbol{n}''_k = B_k \boldsymbol{n}_k$, where A_k and B_k are matrices in $R^{M_k \times 2M_k}$ with elements

$$A_k(p,q) = \begin{cases} 1, & q = 2p-1, \\ 0, & \text{otherwise}, \end{cases} \text{ and } B_k(p,q) = \begin{cases} 1, & q = 2p, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

for all $p = 1, 2, ..., M_k$ and $q = 1, 2, ..., 2M_k$, where M_k is the number of IoT devices in $\mathcal{M}_k$.

It can be proved that by substituting (6)-(8) into (5), it becomes a convex optimization problem in which the objective function (5a) and constraints (5c)-(5d) are completely separated, while constraint (5b) is coupled across all the edge servers. In order to distributively solve problem (5) distributively without requiring each server to disclose any data or network information, we first introduce the following indicator functions to incorporate the inequality constraints (5b)-(5d) into the objective function (5a):

$$I_{\mathcal{D}}(\boldsymbol{n}) = \begin{cases} 0, & \boldsymbol{n} \in \mathcal{D}, \\ \infty, & \boldsymbol{n} \notin \mathcal{D}, \end{cases} \quad I_{\mathcal{D}_k}(\boldsymbol{n}_k) = \begin{cases} 0, & \boldsymbol{n}_k \in \mathcal{D}_k, \\ \infty, & \boldsymbol{n}_k \notin \mathcal{D}_k, \end{cases} \tag{9}$$

where $\boldsymbol{n} = (\boldsymbol{n}_1; \boldsymbol{n}_2; ...; \boldsymbol{n}_K)$, $\mathcal{D} = \{\boldsymbol{n} | \sum_{k \in \mathcal{K}} \nu_k(\boldsymbol{n}_k) \le \epsilon\}$, and $\mathcal{D}_k = \{\boldsymbol{n}_k | \boldsymbol{n}_k$ satisfies (5c-5d)$\}$. The objective function for each subproblem to be solved by an edge server can be rewritten as:

$$f_k(\boldsymbol{n}_k) = \frac{1}{K}(\bar{\mu}_k - \tilde{\mu}_k) + c^u_k(A_k \boldsymbol{n}_k) + c^l_k(B_k \boldsymbol{n}_k). \tag{10}$$

We can then reformulate problem (5) in a standard ADMM form as

$$\min_{\boldsymbol{n}, \boldsymbol{w}} \sum_{k \in \mathcal{K}} \left(f_k(\boldsymbol{n}_k) + I_{\mathcal{D}_k}(\boldsymbol{n}_k) \right) + I_{\mathcal{D}}(\boldsymbol{w}), \tag{11a}$$

$$\text{s.t.} \quad \boldsymbol{n} - \boldsymbol{w} = \boldsymbol{0}, \tag{11b}$$

where $\boldsymbol{w} = \langle \boldsymbol{w}_1; \boldsymbol{w}_2; ...; \boldsymbol{w}_K \rangle$ is the introduced auxiliary variable. We can therefore adopt the following updates to solve problem (11) iteratively:

$$\{\boldsymbol{n}^{i+1}_k\}_{k \in \mathcal{K}} = \underset{\boldsymbol{n}_k}{\operatorname{argmin}} \left(f_k(\boldsymbol{n}_k) + I_{\mathcal{D}_k}(\boldsymbol{n}_k) + \frac{\eta}{2} \|\boldsymbol{n}_k - \boldsymbol{w}^i_k + \boldsymbol{\theta}^i_k\|^2_2 \right), \tag{12a}$$

$$\boldsymbol{w}^{i+1} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left(I_{\mathcal{D}}(\boldsymbol{w}) + \frac{\eta}{2} \|\boldsymbol{n}^{i+1} - \boldsymbol{w} + \boldsymbol{\theta}^i\|^2_2 \right), \tag{12b}$$

$$\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i + \boldsymbol{n}^{i+1} - \boldsymbol{w}^{i+1}, \tag{12c}$$

where i is the iteration index, $\boldsymbol{\theta} = (\boldsymbol{\theta}_1; \boldsymbol{\theta}_2; ...; \boldsymbol{\theta}_K)$ is the dual variable, and η is the augmented Lagrangian parameter. We summarize the details in Algorithm 1 where the stopping criteria follow the same line as that in [13].

Theorem 1: Algorithm 1 converges to the optimal solution of Problem (P3) without requiring any exchange of local datasets among edge servers nor disclosing any IoT network topological information.

Algorithm 1 Resource-Efficiency Optimization Algorithm

```
*Initialization*: $\boldsymbol{n}^0 \in R^{2M \times 1}$, $\eta > 0$;
**for** $i = 0, 1, 2, ...$ **do**
  1. Each edge server $k$ executes the following steps using its local information:
    1) Calculate $\boldsymbol{n}^{i+1}_k$ according to (12a);
    2) Broadcast $\boldsymbol{n}'_m$ and $\boldsymbol{n}'_m$ to each IoT device $m \in \mathcal{M}_k$, so it can allocate its traffic distribution accordingly;
    3) **if** Stopping criteria have been met **then**
          Break;
       **else**
          Report $\boldsymbol{n}^{i+1}_k$ to the FEI coordinator;
       **end if**
  2. When the FEI Coordinator receives all the updates of $\boldsymbol{n}^{i+1}_k$, it will execute the following steps:
    1) Update the $\boldsymbol{w}^{i+1}$ and $\boldsymbol{\theta}^{i+1}$ according to (12b) and (12c);
    2) Feedback $\boldsymbol{w}^{i+1}_k$ and $\boldsymbol{\theta}^{i+1}_k$ to each edge server $k$.
**end for**
```

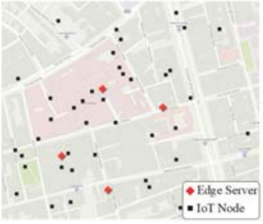

Fig. 3. Simulation setup.

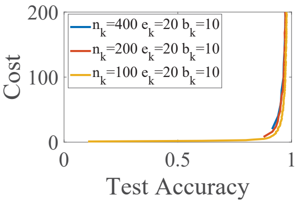

Fig. 4. Cost of FEI under different requirements of model accuracy.

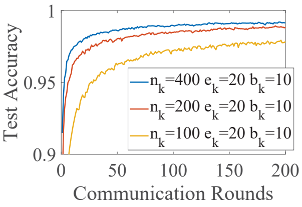

Fig. 5. Convergence performance under different model training parameters.

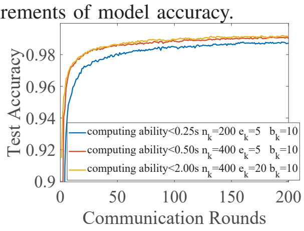

Fig. 6. Convergence performance under different computational capacities of edge server.

Proof: We briefly describe our proof as follows:

1) The FEI coordinator only receives the intermediate traffic allocation results from edge servers which cannot be used to reveal any content information of their local datasets.
2) Each edge server only receives subvectors $\boldsymbol{w}^{i+1}_k$ and $\boldsymbol{\theta}^{i+1}_k$ from the FEI coordinator. It cannot derive any information about other edge servers and IoT devices.
3) Each IoT device only receives traffic allocation results $\boldsymbol{n}'_m$ and $\boldsymbol{n}'_m$ from its associated edge server which do not contain any information about other IoT devices or edge servers.

■

V. Numerical Result

To evaluate the performance of our proposed framework, we simulate an IoT network that can upload data samples to 4 edge servers using the real locations of Starbuck's coffee shops in the city center of Dublin city as shown in Figure 3. We assume that each edge server can receive data from 10 IoT devices. We use the data samples from the EMNIST dataset to simulate the data uploaded from each IoT device to an edge server. We equally divide 341873 training examples and 40832 test examples among 4 groups of IoT devices and in each round of data uploading, each IoT device randomly selects data samples from its local training example pool to upload to the associated edge server. The average sample size of EMNIST is 0.67 KB. We assume each IoT device can use Zigbee or NB-IoT to submit data samples with transmit powers 63 mW and 100 mW, respectively. We simulate each edge server with a TITAN X GPU running on Ubuntu 16.04. The training process among 4 edge servers is simulated using Federated-Tensorflow framework 0.16.1. Each edge server trains a four-layer CNN model locally.

We first evaluate the impact of the data uploading performance of IoT network on the convergence rate of the FL model training. In Figure 4, we consider a fixed computational capacity of each edge server (computational time for each round of training is less than 0.25s, 0.5s, and 2s, respectively) and compare the total cost under different model accuracies when the IoT network can use only LB-IoT, only UB-IoT, and optimal data uploading configuration with both IoT technologies. We can observe that the optimal data uploading scheme significantly reduces the total cost. The LB-IoT however results in the highest cost among all the IoT solutions. We then present the model accuracy under different communication rounds in the same setup in Figure 5. In this case, we can observe that LB-IoT achieves the best convergence performance among all three solutions. This is because LB-IoT can always upload the required amount of data samples to edge servers during every round of training. The optimal uploading scheme however tries to balance the update uploading performance and the cost and therefore results in slightly reduced convergence performance. We can also observe that, with only a small degradation of convergence performance of the training process, the cost of the IoT network can be significantly reduced. In other words, our proposed optimal uploading scheme is more suitable for cost-sensitive IoT network that is not very sensitive to the accuracy or convergence of the model.

To evaluate the impact of the computational capacity of edge servers on the model training performance, we compare the convergence performance under different upper limits of the computational load (maximum computational time per round of training) in Figure 6. We can observe that the increasing speed of the model training performance in terms of convergence speed and accuracy improvement reduces with the computational capacity. Therefore, for IoT networks with limited model accuracy requirements, deploying low-cost edge servers with limited computational capacity may be sufficient for most FL model training tasks.

VI. Conclusion

This paper has introduced the FEI framework to jointly optimize the data collection and transportation of an IoT network and resource utilization of an edge computing system. FEI allows edge servers to first evaluate the required number of data samples according to its local data processing capacity and requirements of the FL model and will request a sufficient amount of data with minimized energy cost for the IoT network. The energy cost of the IoT network has been evaluated when the data uploading of each IoT device can be supported by two widely-used IoT solutions, LB-IoT and UB-IoT. The relationship between the convergence performance and accuracy of FL model training and the impact of three key parameters, size of the dataset, batch size, and number of local training passes, has also been investigated. A joint optimization problem that can maximize the utilization of the computational resources at edge servers and also minimize the energy cost of the IoT network has been formulated. A distributed algorithm based on ADMM has been proposed to address the formulated problems. We have proved that the proposed algorithm will not cause any data leakage of local dataset and will preserve the private information related to the topological property of IoT network. Simulation results have been presented to show that the proposed framework can significantly reduce the cost with only a limited degradation in model convergence.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62071193, the Key R & D Program of Hubei Province of China under Grant No. 2020BAA002, and the U.S. National Science Foundation under Grant No. CCF-1908308.

References

[1] R. Yu, G. Xue, and X. Zhang, "Application provisioning in fog computing-enabled Internet-of-Things: A network perspective," in *Proc. of IEEE INFOCOM*, Honolulu, HI, Apr. 2018.

[2] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," *IEEE J. Sel. Area Commun.*, vol. 37, no. 3, pp. 668–682, Mar. 2019.

[3] Y. Xiao and M. Krunz, "Dynamic network slicing for scalable fog computing systems with energy harvesting," *IEEE J. Sel. Area Commun.*, vol. 36, no. 12, pp. 2640–2654, Dec. 2018.

[4] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint 1806.00582*, Jun. 2018.

[5] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. of IEEE INFOCOM*, Virtual Conference, Apr. 2020.

[6] Y. Tu, Y. Ruan, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," in *Proc. of IEEE INFOCOM*, Virtual Conference, Apr. 2020.

[7] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of the International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, Apr. 2017, pp. 1273–1282.

[8] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, Mar. 2019.

[9] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[10] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless lans," in *Proc. of PIMRC*, vol. 2, Taipei, Taiwan, Oct. 1996, pp. 392–396.

[11] Y. Xiao and M. Krunz, "Distributed resource allocation for network slicing over licensed and unlicensed bands," *IEEE J. Sel. Area Commun.*, vol. 36, no. 10, pp. 2260 – 2274, Dec. 2018.

[12] M. Hirzallah, Y. Xiao, and M. Krunz, "On modeling and optimizing LTE/Wi-Fi coexistence with prioritized traffic classes," in *Proc. of IEEE DySPAN*, Seoul, South Korea, Oct. 2018, pp. 1–10.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, Jan. 2011.