Loss and Energy Tradeoff in Multi-access Edge Computing Enabled Federated Learning

Chit Wutyee Zaw, Choong Seon Hong
Department of Computer Science and Engineering, Kyung Hee University,
Yongin-si, Republic of Korea
Email: {cwyzaw, cshong}@khu.ac.kr

***Abstract*—Federated learning (FL) encourages users to train statistical models on their local devices. Since mobile devices have the limited power and computing capabilities, the users are rational in minimizing their energy consumption with the cost of the model's accuracy. Multi-access Edge Computing (MEC) enabled FL is a prominent approach where users can offload a fraction of their dataset to the MEC server where the training of the statistical model is performed with the help of the powerful MEC server in parallel with the local training at the mobile users. With the size of dataset offloaded to the MEC server, both the performance of the model and the energy consumption of the system are varied. We analyze this tradeoff between the performance of the system and the energy consumption at the MEC server and mobile users. The time consumption can also be saved by managing the size of the dataset offloaded to the MEC server. Since the MEC server and mobile users have the conflicting interest in saving the energy consumption with the constraint on the time taken for one computing round where the performance of the model fluctuates across the size of offloaded dataset, we analyze the tradeoff by formulating the resource management problem as a penalized convex optimization problem. We propose a distributed resource management problem for MEC enabled FL system where the global model is responsible for radio resource management and each local model performs a dataset offloading decision. Then, we perform the simulation to show the tradeoff and performance of the proposed algorithm.**

***Index Terms*—Data offloading, Federated learning, Penalized convex optimization problem, Multi-access edge computing, Resource management**

I. Introduction

Multi-access Edge Computing (MEC) enables the enengy constrained mobile devices to offload their computation intensive tasks to an edge server so that the time and energy consumption of the users are minimized. By deploying an edge server at a basestations, the providers can grant the low latency services to the users with the help of the high computation capabality of the edge server. The resource management in MEC system has been analyzed in the recent works. MEC for 5G network is surveyed in [1]. In addition, Federated Learning (FL) allows users to train a local model on the datasets generated by their mobile devices. A statistical model is built by combining the local models trained by the users. Challenges and future directions for FL are discussed in [2]. Authors in [3] describes the advances and open problems in FL. The challenges and existing works of FL in mobile edge networks are surveyed in [4].

Since the mobile users have the limitation in battery and computation capacity, the computing power of the edge server can be utilized by training a model on the edge server. The mobile users can offload a portion of the local datasets to the edge server where the training is performed on the datasets offloaded by the users. There is a tradeoff between the model training loss, energy and time consumption of the mobile users with respect to the data offloaded to the edge server. Thus, the resource management has become a comfronting issue in MEC enabled FL systems. Due to a large number of participants in FL, the distributed solution approaches are required for the resource management for FL.

A. Related Works

The wireless resource allocation with respect to the model training is widely addressed in recent works. Authors in [5] considered the wireless resource allocation and the learning model jointly where the convergence of the algorithm is derived. An optimization model for FL is proposed in [6] in which authors discussed a tradeoff between learning time and the energy consumption of users. The edge association of users to the servers is considered in [7] where the energy and time are jointly minimized by allocating the wireless bandwidth and local computing resources. Authors in [8] considered a joint data sampling and participant selection for FL in mobile edge computing system where the model training loss is jointly minimized with the cost of the training. However, authors did not take the time consumption into account which is an important factor to consider in the resource allocation. Moreover, the distributed resource allocation approach for the MEC system is discussed in [9] and [10].

B. Our Contribution

In this paper, we propose a MEC enabled federated learning model by utilizing the computation capabality of the edge server. The participating users decide the fraction of the local dataset to offload to the edge server to train a statistical model while training the local model on the remaining dataset simultaneously. The edge server trains a model on the datasets

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing). *Dr. CS Hong is the corresponding author.

offloaded from the users in parallel with the local training. We formulate a distributed resource management problem where the edge server controls the uplink bandwidth allocation for the dataset offloading and weight transmission while the mobile users decide the datasets to offload to the edge server. The objective of the edge server and mobile users is to minimize the cost which is the combination of the model training loss and the energy consumption within the specified time duration. A distributed resource management algorithm is proposed to solve the formulated problem. The simulation is performed to show the performance of the algorithm.

II. System Model

In this paper, we consider the MEC enabled federated learning model in which an edge server is responsible for training a statistical model simultaneously with the participating mobile users. By utilizing the edge sever in MEC system, users can offload a portion of their dataset to train a model at the edge server so that the users can save the energy consumption of their devices. Our MEC enabled FL system consists of an edge server, E, and a set of mobile users, $\mathcal{I} = \{1, 2, \cdots, n\}$. Each user, i, has a local dataset $\mathcal{D}_i$ to train a statistical model by participating in FL. The MEC enable FL system model is shown in Fig. 1 where user i offloads a portion of their dataset, δ_i to the edge server, E, and train the local model with the remaining portion of dataset $(1 - \delta_i)$. User i determines δ_i independently with other users and the edge server, E. The bandwidth allocation is performed by the edge server, E, to offload the dataset and paramenters of the local training model at users.

The time consumption of our proposed MEC enabled FL model and traditional FL model is shown in Fig. 2. In traditional FL, mobile users participating in FL train their local models and transfer the weights of local model after training. The edge node, which is considered the edge server in our paper, aggregates the weights from all participating users after receiving from all users. In our proposed model, user i can upload δ_i portion of the local dataset and train the local model with $(1-\delta_i)$ portion of dataset simultaneously. After receiving the offloaded dataset from all users, the edge server trains a model with the offloaded dataset from all users. The model aggregation is performed after the edge server and mobile users finish the training, and the weight transmission. For both traditional and proposed MEC enabled FL, the synchronous update model is considered.

A. Federated Learning Model

In FL, user i trains a statistical model with the local dataset, $\mathcal{D}_i$, by minimizing the loss function which is defined as follows.

$$\underset{\mathbf{w}_i}{\text{minimize}} \sum_{j \in \mathcal{D}_i} l(\mathbf{w}_i, x_j, y_j),$$

where $\mathbf{w}_i$ is the weight vector which captures the local model. x_j and y_j are the features and label of sample j in the local dataset $\mathcal{D}_i$. In this paper, we perform the linear regression model where the Mean Squared Error is considered as the loss function. After training the local model, users transfer the model which is the weight vector to the edge server. The edge server then aggregate the weight vectors from all participating users and the final model, $\mathbf{w}_f$ is obtained as follows.

$$\mathbf{w}_f = \frac{\sum_{i \in \mathcal{I}} |\mathcal{D}_i| \mathbf{w}_i}{\sum_{i \in \mathcal{I}} |\mathcal{D}_i|},$$

where the local weights, $\mathbf{w}_i$ are adjusted by the contribution of the user i in the model training which is the proportion of local dataset with respect to sum of all local datasets of users. However, in our proposed MEC enabled FL system, the edge server perform the model training as well with the data offloaded from the participating users which is described in following subsections.

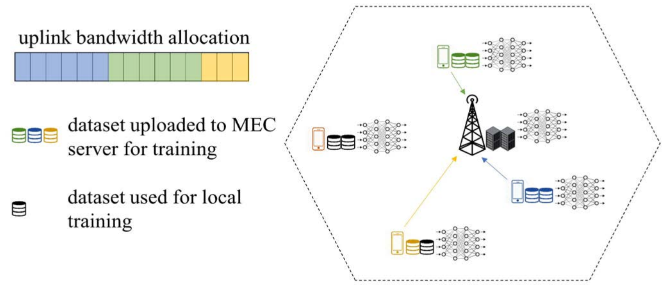

Fig. 1. Multi-access edge computing enabled federated learning system.

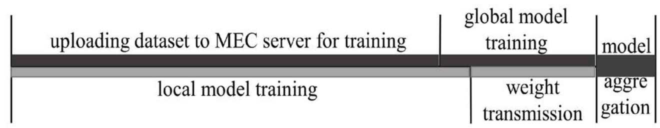

Fig. 2. Time consumption for one computing round.

B. Local Model

In this section, we define the local model training, energy and time consumption of the participating mobile devices. In our proposed MEC enabled FL system, user i is allowed to offload δ_i portion of the local dataset to the edge server E. The remaining $(1 - \delta_i)$ portion of dataset is used for the local training. Let $\bar{\mathcal{D}}_i$ be the dataset offloaded to the edge server E where the data samples are randomly chosen from $\mathcal{D}_i$ and $|\bar{\mathcal{D}}_i| = \delta_i |\mathcal{D}_i|$. Let $\tilde{\mathcal{D}}_i$ be the dataset to train the local model of user i where $\tilde{\mathcal{D}}_i = \mathcal{D}_i - \bar{\mathcal{D}}_i$, $|\tilde{\mathcal{D}}_i| = (1 - \delta_i)|\mathcal{D}_i|$ and $\mathcal{D}_i = \bar{\mathcal{D}}_i \cup \tilde{\mathcal{D}}_i$. Thus, user i performs the training of the local model as follows.

$$\underset{\mathbf{w}_i}{\text{minimize}} \sum_{j \in \tilde{\mathcal{D}}_i} l(\mathbf{w}_i, x_j, y_j). \tag{1}$$

To participate in our MEC enabled FL system, user i needs to perform three steps which are the dataset transmission, local model training and model (weight) transmission. In this paper, we consider Orthogonal Frequency Division Multiple Access for communication. Thus, the energy consumption of user i is calculated as follows.

$$e_i(\bar{\beta}_i, \tilde{\beta}_i, \delta_i) = p_i \left(\frac{\delta_i c_d |\mathcal{D}_i|}{\bar{\beta}_i \log_2(1+\frac{p_i h_i}{n_0})} + \frac{c_w |\mathbf{w}_i|}{\tilde{\beta}_i \log_2(1+\frac{p_i h_i}{n_0})} \right) + \xi(1-\delta_i) c_d |\mathcal{D}_i| c_f^2, \tag{2}$$

where p_i is the trasmission power of user i, h_i is the channel gain, n_0 is noise power, ξ is the chip capacitance, c_f is the CPU cycles required for computing one byte data, c_d and c_w are the size of a data sample and weight vector respectively. $\bar{\beta}_i$ and $\tilde{\beta}_i$ are the uplink bandwidth allocation for the dataset offloading and weight transmission respectively.

The time consumption for the dataset offloading, local training and weight transmission is defined as follows. The time taken for offloading the dataset of the size $\delta_i|\mathcal{D}_i|$ is

$$t_i^{\text{offload}}(\bar{\beta}_i, \delta_i) = \frac{\delta_i c_d |\mathcal{D}_i|}{\bar{\beta}_i \log_2(1+\frac{p_i h_i}{n_0})}. \tag{3}$$

The time required to perform the local traning with the dataset of size $(1-\delta_i)c_d|\mathcal{D}_i|$ is

$$t_i^{\text{training}}(\delta_i) = \frac{(1-\delta_i) c_d |\mathcal{D}_i| c_f}{F_i}, \tag{4}$$

where F_i is the CPU cycle speed of the mobile device of user i. The time consumption for the weight transmission is calculated as follows.

$$t_i^{\text{trans}}(\tilde{\beta}_i) = \frac{c_w |\mathbf{w}_i|}{\tilde{\beta}_i \log_2(1+\frac{p_i h_i}{n_0})}. \tag{5}$$

C. Edge Model

After receiving the offloaded dataset, $\bar{\mathcal{D}}_i, \forall i \in \mathcal{I}$, the edge server performs the model training where the loss function with respect to the offladed datasets is described as follows.

$$\underset{\mathbf{w}_E}{\text{minimize}} \sum_{j \in \bar{\mathcal{D}}_E} l(\mathbf{w}_E, x_j, y_j), \tag{6}$$

where $\bar{\mathcal{D}}_E = \cup_{i\in\mathcal{I}} \bar{\mathcal{D}}_i$. The energy consumption of the edge server depends on the traffic and computing load at the edge server. The traffic load on the uplink transmission for the dataset offloading and weight transmission is defined as $\bar{\alpha} = \frac{\sum_{i\in\mathcal{I}} \delta_i c_d |\mathcal{D}_i|}{\sum_{i\in\mathcal{I}} \bar{\beta}_i \log_2(1+\frac{p_i h_i}{n_0})}$ and $\tilde{\alpha} = \frac{\sum_{i\in\mathcal{I}} c_w |\mathbf{w}_i|}{\sum_{i\in\mathcal{I}} \tilde{\beta}_i \log_2(1+\frac{p_i h_i}{n_0})}$ respectively. The computing load at the edge server for training the model is same as the traffic load for dataset offloading because the offloaded datasets are used in the model training.

$$e_E(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) = (\bar{\alpha} + \tilde{a}) P^{\text{BS}} + \bar{\alpha} P^{\text{MEC}}, \tag{7}$$

where P^{BS} and P^{MEC} are the maximum available power of the edge base station and MEC edge server, and $\bar{\boldsymbol{\beta}} = [\bar{\beta}]_{i\in\mathcal{I}}^T, \tilde{\boldsymbol{\beta}} = \left[\tilde{\beta}\right]_{i\in\mathcal{I}}^T, \boldsymbol{\delta} = [\delta]_{i\in\mathcal{I}}^T$ which are the vector representation of the resource management variables. The time consumption for training the model at the edge server with respect to all the offloaded dataset is calculated as follows.

$$t_E^{\text{training}}(\boldsymbol{\delta}) = \frac{\sum_{i\in\mathcal{I}} (1-\delta_i) c_d |\mathcal{D}_i| c_f}{F_E}, \tag{8}$$

where F_E is CPU cycle speed of the edge server.

D. Model Aggregation

Once the mobile users and the edge sever finish the training their models, the model aggregation is performed which is the summation of the weights with respect to their contribution in FL. The model aggregation is performed as follows.

$$\mathbf{w}_f = \frac{1}{\sum_{i\in\mathcal{I}} |\mathcal{D}_i|} \left(\sum_{i\in\mathcal{I}} |\tilde{\mathcal{D}}_i| \mathbf{w}_i + \sum_{i\in\mathcal{I}} \bar{\mathcal{D}}_i \mathbf{w}_E \right).$$

III. Problem Formulation

In this section, the resource management problem is formulated where the mobile users and the edge server minimize their cost which is defined by the loss function and their energy consumption. The mobile users and the edge server manage their resources which are the portion of dataset offloaded to the edge server and the uplink bandwidth allocation respectively so as to minimize their cost. Let t_i be the time taken for one computing round which is the time taken for the traditional FL approach.

$$t_i(\tilde{\beta}_i, \delta_i) = t_i^{\text{training}}(\delta_i) + t_i^{\text{trans}}(\tilde{\beta}_i). \tag{9}$$

Let t_E be the time taken for the edge server to train the model with all the offloaded datasets.

$$t_E(\bar{\boldsymbol{\beta}}, \boldsymbol{\delta}) = \max_i \left\{ t_i^{\text{offload}}(\bar{\beta}_i, \delta_i) \right\} + t_E^{\text{training}}(\boldsymbol{\delta}), \tag{10}$$

where $\max_i\{t_i^{\text{offload}}(\bar{\beta}_i)\}$ is taken since the edge server can begin the model training after receiving all the datasets offloaded from the mobile users. Thus, the total time taken for the proposed MEC enabled FL model is defined as

$$T(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) = \max_{i,E} \left\{ t_i(\tilde{\beta}_i, \delta_i), t_E(\bar{\boldsymbol{\beta}}, \boldsymbol{\delta}) \right\}. \tag{11}$$

Thus, the resource management problem for the MEC enabled FL to minimize the cost fuction by managing the uplink bandwith, the offloaded dataset is formulated as follows.

The objective of the edge server is to minimize the cost function which is the sum of the model training loss and energy consumption by allocating the uplink bandwith for dataset offloading and weight transmission within the specified time period, Δt, which is defined as follows.

$$\begin{aligned} \underset{\mathbf{w}_E, \bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}}{\text{minimize}} \quad & \sum_{j\in\bar{\mathcal{D}}_E} l(\mathbf{w}_E, x_j, y_j) + e_E(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) \\ \text{subject to} \quad & T(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) \le \Delta t. \end{aligned} \tag{12}$$

User i management the portion of dataset offloaded to the edge server, δ_i so as to minimize the loss of the model training and energy consumption. The resource management problem for user i is formulated as follows.

$$\begin{aligned} \underset{\mathbf{w}_i, \delta_i}{\text{minimize}} \quad & \sum_{j\in\tilde{\mathcal{D}}_i} l(\mathbf{w}_i, x_j, y_j) + e_i(\bar{\beta}_i, \tilde{\beta}_i, \delta_i) \\ \text{subject to} \quad & T(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) \le \Delta t, \end{aligned} \tag{13}$$

where the total time required for the model training is constrained by Δt.

The formulated resource management problem is hard to solve due to the coupling in $T(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta})$ between the edge server and users. In addition, the loss function is not guaranteed to be convex with respect to the dataset used for training which is later shown in section V. Thus, we propose a penalized distributed algorithm where the model learning and the resource management problem are solved alternatively.

IV. Distributed Resource Management Algorithm

In this section, a distributed resource management algorithm is proposed for the MEC enabled FL system. Since there is a coupling between the model training and the portion of datasets offloaded to the edge server, δ_i, we decouple the model training and the resource management problem.

User $i, \forall i \in \mathcal{I}$ and the edge server perform the model training by solving (1) and (6) with the fixed $\boldsymbol{\delta}$. The offloaded decision, $\boldsymbol{\delta}$ is updated by solving the following problems which are solved by the edge server and user $i, \forall i \in \mathcal{I}$ simultaneously. The resource management problem for the edge server is defined as follows.

$$\begin{aligned} \underset{\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}}{\text{minimize}} \quad & e_E(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) \\ \text{subject to} \quad & T(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) \leq \Delta t. \end{aligned} \tag{14}$$

The resource management problem for the user i is defined as

$$\begin{aligned} \underset{\delta_i}{\text{minimize}} \quad & e_i(\bar{\beta}_i, \tilde{\beta}_i, \delta_i) \\ \text{subject to} \quad & T(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) \leq \Delta t. \end{aligned} \tag{15}$$

Since there is a resource coupling in $T(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta})$ in (14) and (15), the formulated resource management problem is transformed by setting a penalty parameter to the time constraint as follows. Let $\hat{T}(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) = T(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) - \Delta t$. The penalized resource management problem for the edge server is

$$\underset{\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}}{\text{minimize}} \quad e_E(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}) + \lambda_E \hat{T}(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}), \tag{16}$$

where λ_E is the penalty parameter controlled by the edge server with respect to the time constraint. (16) is convex with respect to $\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}$ for the fixed $\boldsymbol{\delta}$ and λ_E. The penalized resource management problem for user i is formulated as

$$\underset{\delta_i}{\text{minimize}} \quad e_i(\bar{\beta}_i, \tilde{\beta}_i, \delta_i) + \lambda_i \hat{T}(\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}), \tag{17}$$

where λ_i is the penalty parameter controlled by user i to prevent the violation of time constraint. (17) is a convex problem with respect to δ_i for the fixed $\bar{\beta}_i, \tilde{\beta}_i$ and λ_i. Thus, (16) and (17) can be solved efficiently. The distributed resource management algorithm is proposed in algorithm 1.

Algorithm 1 Distributed Resource Management Algorithm

Output: $\bar{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}, \boldsymbol{\delta}$

set initial values for $\bar{\boldsymbol{\beta}}^0, \tilde{\boldsymbol{\beta}}^0, \boldsymbol{\delta}^0, \lambda_E^0, \lambda_i^0$.
$k \leftarrow 0$
repeat
  the edge server trains the model as in (6) with $\boldsymbol{\delta}^0$.
  user i performs the model training as in (1) with $\delta_i^0, \forall i \in \mathcal{I}$.
  $k \leftarrow k + 1$
  $\left[\bar{\boldsymbol{\beta}}^k, \tilde{\boldsymbol{\beta}}^k\right] \leftarrow \left[\bar{\boldsymbol{\beta}}^*, \tilde{\boldsymbol{\beta}}^*\right]$ by solving (16) with respect to $\delta_i^{k-1}, \lambda_i^{k-1}, \forall i \in \mathcal{I}$.
  $\delta_i^k \leftarrow \delta_i^*$ by solving (17) with respect to $\bar{\beta}_i^{k-1}, \tilde{\beta}_i^{k-1}, \lambda_i^{k-1}$.
  update λ_E^k according to (18).
  update λ_i^k according to (19), $\forall i \in \mathcal{I}$.
until $\epsilon_E^k < \epsilon$ and $\epsilon_i^k < \epsilon$.

Let ϵ_E^k and ϵ_i^k be the difference of the objective values between iteration k and $k-1$ for the edge server and user $i, \forall i \in \mathcal{I}$ respectively which are calculated as follows.

$$\begin{aligned} \epsilon_E^k = |(& \sum_{j \in \bar{\mathcal{D}}_E^{k-1}} l(\mathbf{w}_E, x_j, y_j) + e_E(\bar{\boldsymbol{\beta}}^{k-1}, \tilde{\boldsymbol{\beta}}^{k-1}, \boldsymbol{\delta}^{k-1})) \\ & - (\sum_{j \in \bar{\mathcal{D}}_E^k} l(\mathbf{w}_E, x_j, y_j) + e_E(\bar{\boldsymbol{\beta}}^k, \tilde{\boldsymbol{\beta}}^k, \boldsymbol{\delta}^k))| \\ \epsilon_i^k = |(& \sum_{j \in \tilde{\mathcal{D}}_i^{k-1}} l(\mathbf{w}_i, x_j, y_j) + e_i(\bar{\beta}_i^{k-1}, \tilde{\beta}_i^{k-1}, \delta_i^{k-1})) \\ & - (\sum_{j \in \tilde{\mathcal{D}}_i^k} l(\mathbf{w}_i, x_j, y_j) + e_i(\bar{\beta}_i^k, \tilde{\beta}_i^k, \delta_i^k))| \end{aligned}$$

The penalty parameters, λ_E and $\lambda_i \forall i \in \mathcal{I}$ are updated as follows.

$$\lambda_E^k = \begin{cases} \lambda_E^{k-1}, & \text{if } T(\bar{\boldsymbol{\beta}}^*, \tilde{\boldsymbol{\beta}}^*, \boldsymbol{\delta}^{k-1}) \leq \Delta t \\ \lambda_E^{k-1} + \gamma_E, & \text{otherwise} \end{cases} \tag{18}$$

$$\lambda_i^k = \begin{cases} \lambda_i^{k-1}, & \text{if } T(\bar{\beta}_i^{k-1}, \tilde{\beta}_i^{k-1}, \delta^*) \leq \Delta t \\ \lambda_i^{k-1} + \gamma_i, & \text{otherwise} \end{cases}, \tag{19}$$

where γ_E and γ_i are the increment for the penalty parameters so that the penalty values will keep increasing if the constraint is violoated.

V. Simulation Results

In this paper, we consider a single cell scenario with an edge server is deployed together with a macro basestation. We perform a linear regression model on 100 mobile users which are randomly located in 1 km radius. The total available uplink bandwidth is 20 MHz and the power density thermal noise is used which is -174 dBm/Hz. The edge server is equipped with a 16 GHz CPU and the value of Δt is 30 s. The edge server is denoted as global model and the mobile users are denoted as local model in the following figures.

Fig. 3 shows the training loss and time consumption between the traditional and proposed FL. Since the data samples

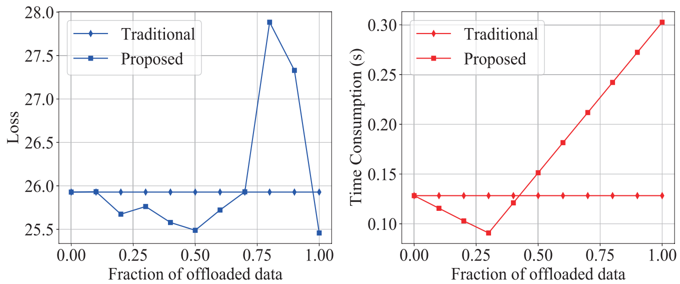

Fig. 3. Loss and time consumption with respect to fraction of offloaded data.

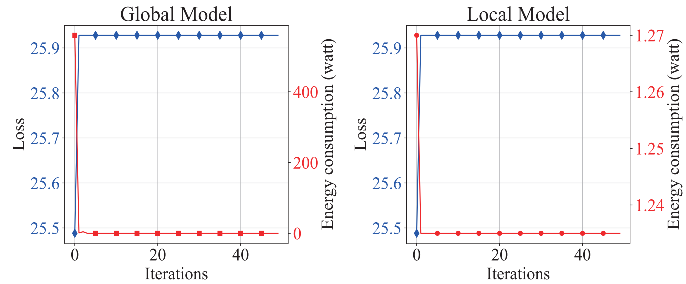

Fig. 5. Performance of the algorithm on loss and energy consumption.

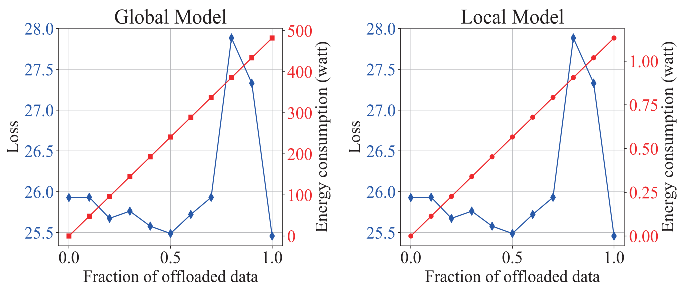

Fig. 4. Loss and energy consumption with respect to fraction of offloaded data.

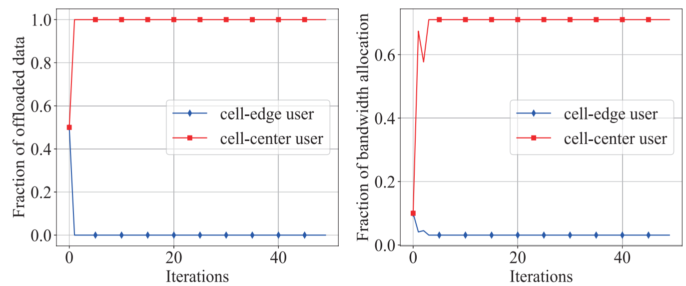

Fig. 6. Performance of the algorithm on offloaded dataset and bandwidth allocation.

for the offloaded dataset is chosen randomly, the training loss of the models varied. The fraction of offloaded data of value 0.0 means that the local dataset is not offloaded to the edge server and trained at the local model whereas the value 1.0 means the whole dataset is offloaded to the edge server. The training loss of the most cases, except for values 0.8 and 0.9, is less than the traditional FL in which the dataset is not offloaded to the edge server. When a small fraction of dataset is offloaded to the edge server, the resulting time consumption is less than that of traditional FL because the edge server is equipped with powerful computation capability and the local model is training the model with less data samples. However, as the fraction of offloaded dataset increases after a threshold, the time consumption also increases due to the increased load on the uplink transmission.

Fig. 4 shows the training model loss and energy consumption of the global and local model. The energy consumption of the global model increases with the fraction of offloaded dataset while the local model's energy consumption decreases. In some cases, loss is also decreasing with the energy consumption of local model where the global model's energy consumption is compromised. Since both the edge server and mobile users want to minimize the energy consumption, the fraction of dataset offloaded needs to be optimized to satisfy the objectives.

Fig. 5 shows the convergence of the algorithm on the training loss and energy consumption of the global and local model. The training loss has to be increased by a small amount in order to save the energy consumption due to the tradeoff between them. The alogrithm converges in a few iteration. Fig. 6 shows the fraction of offloaded dataset and bandwidth allocation of the cell-edge and cell-center users. Since the cell-edge user is situated far from the edge server, only a small fraction of the dataset is offloaded to the edge server so that the only few bandwidth resources are needed for the offloading. However, for the cell-center user, almost all the dataset is offloaded to the edge server in which a large fraction of bandwidth resources is requiered for the transmission.

VI. Conclusion

In this paper, we propose a Multi-access Edge Computing enabled Federated Learning where the computing power of the edge server is utilized by performing the model training at the server with the datasets offloaded by the participating users. The resource management problem is formulated where the edge server optimizes the uplink bandwidth allocation for dataset offloading and weights transmission by minimizing its cost function which is the combination of the traning model loss and energy consumption. The participating users control the fraction of dataset offloaded to the edge server where the local training loss and energy consumption are minimized. For the formulated problem, we propose a distributed resource management algorithm by setting a penalty parameter to the coupling constraints so as to solve the formulated problem independently. The simulation is then performed to show the performance of the proposed algorithm, the coupling in the resource management, time and the energy consumption.

References

[1] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, May 2017.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020.
[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, Dec 2019.
[4] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, Apr 2020.
[5] 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*, Sep 2019.
[6] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, Paris, Apr 2019, pp. 1387–1395.
[7] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, "Hfel: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning," *arXiv preprint arXiv:2002.11343*, Feb 2020.
[8] C. Feng, Y. Wang, Z. Zhao, T. Q. Quek, and M. Peng, "Joint optimization of data sampling and user selection for federated learning in the mobile edge computing systems," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, Dublin, Jun 2020, pp. 1–6.
[9] C. W. Zaw, N. N. Ei, H. Y. R. Im, Y. K. Tun, and C. S. Hong, "Cost and latency tradeoff in mobile edge computing: A distributed game approach," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, Kyoto, Feb 2019, pp. 1–7.
[10] C. W. Zaw, N. H. Tran, W. Saad, Z. Han, and C. S. Hong, "Generalized nash equilibrium game for radio and computing resource allocation in co-located mec," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, Dublin, Jun 2020, pp. 1–6.