

Energy or Accuracy? Near-Optimal User Selection and Aggregator Placement for Federated Learning in MEC

Zichuan Xu ¹, Member, IEEE, Dongrui Li ², Weifa Liang ³, Senior Member, IEEE, Wenzheng Xu ⁴, Member, IEEE, Qiufen Xia ⁵, Member, IEEE, Pan Zhou ⁶, Senior Member, IEEE, Omer F. Rana ⁷, Senior Member, IEEE, and Hao Li ⁸

Abstract—To unveil the hidden value in the datasets of user equipments (UEs) while preserving user privacy, federated learning (FL) is emerging as a promising technique to train a machine learning model using the datasets of UEs locally without uploading the datasets to a central location. Customers require to train machine learning models based on different datasets of UEs, through issuing FL requests that are implemented by FL services in a mobile edge computing (MEC) network. A key challenge of enabling FL in MEC networks is how to minimize the energy consumption of implementing FL requests while guaranteeing the accuracy of machine learning models, given that the availabilities of UEs usually are uncertain. In this paper, we investigate the problem of energy minimization for FL in an MEC network with uncertain availabilities of UEs. We first consider the energy minimization problem for a single FL request in an MEC network. We then propose a novel optimization framework for the problem with a single FL request, which consists of (1) an online learning algorithm with a bounded regret for the UE selection, by considering various contexts (side information) that influence energy consumption; and

(2) an approximation algorithm with an approximation ratio for the aggregator placement for a single FL request. We third deal with the problem with multiple FL requests, for which we devise an online learning algorithm with a bounded regret. We finally evaluate the performance of the proposed algorithms by extensive experiments. Experimental results show that the proposed algorithms outperform their counterparts by reducing at least 13% of the total energy consumption while achieving the same accuracy.

Index Terms—Energy minimization, federated learning, machine learning based algorithms, mobile edge computing, UE selection and aggregator placement.

I. INTRODUCTION

FEDERATED learning (FL) enables users to collaboratively learn a machine learning model, by performing local training in user equipments (UEs) and aggregating the local models to a central site to obtain a global model in a mobile edge computing (MEC) network. The process of conventional FL consists of multiple rounds of learning, and within each round, multiple chosen UEs perform their local model training and the trained local models are then aggregated by a parameter server for the next round of training. This procedure continues until an accurate global model is obtained. Each FL process aims to obtain a machine learning model. In reality, different users may require to learn different machine learning models. As such, in this paper we consider that users issue *FL requests* to train different datasets of UEs; while the service provider of the MEC network implements the FL requests via *FL services*. For example, users can issue different FL requests to train machine learning models for disease classification of different diseases in healthcare [7].

To perform high-quality implementation of FL requests, the service provider deploys FL services within the proximity of UEs, to implement FL requests with low latency and accurate global models. However, since UEs may have their own functionalities, they perform opportunistic local training for FL. That is, they may prefer to contribute local training when they are idle without any significant resource-consuming activities [50], [54]. For instance, intelligent vehicles have their own intensive computing tasks, such as object detection/tracking and traffic sign detection/classification [66], which make them only available for local training when they are in charging or parked. As such,

Manuscript received 15 April 2022; revised 2 January 2023; accepted 16 March 2023. Date of publication 29 March 2023; date of current version 5 February 2024. The work of Zichuan Xu and Qiufen Xia was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62172068 and 62172071, and the “Xinghai scholar” program. The work of Weifa Liang was supported in part by the City University of Hong Kong under Grant 9380137/CS. The work of Wenzheng Xu was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62272328. The work of Pan Zhou was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61972448. Recommended for acceptance by W. Gao. (Corresponding author: Hao Li.)

Zichuan Xu and Dongrui Li are with the School of Software, Dalian University of Technology, Dalian 116024, China, and also with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116024, China (e-mail: z.xu@dlut.edu.cn; dawncoming@mail.dlut.edu.cn).

Weifa Liang is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: weifa.liang@cityu.edu.hk).

Wenzheng Xu is with the Department of Computer Science, Sichuan University, Chengdu 610017, China (e-mail: wenzheng.xu@scu.edu.cn).

Qiufen Xia is with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, International School of Information Science & Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: qiufenxia@dlut.edu.cn).

Pan Zhou is with the Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: panzhou@hust.edu.cn).

Omer F. Rana is with the Physical Sciences and Engineering College, Cardiff University, CF10 3AT Cardiff, U.K. (e-mail: RanaOF@cardiff.ac.uk).

Hao Li is with the Ningxia Research Institute, China Coal Research Institute, Ningxia 750004, China (e-mail: hateif@163.com).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TMC.2023.3262829>, provided by the authors.

Digital Object Identifier 10.1109/TMC.2023.3262829

a UE may not be able to perform a sufficient number of local training epochs, thereby reducing the accuracy of the obtained model. Without sufficient number of UEs participating in FL, the local models will be trained based on insufficient data, which can reduce the accuracy of the global machine learning model. Thus, the availability of UEs is fundamentally important to ensure the accuracy of the model.

A. Motivations and Challenges

It is observed that the availability of UEs usually is uncertain, which typically depends on various factors, including the energy level, data quality, and available computation capacities on UEs. In particular, the energy level of a UE has a direct impact on the availability of the UE. Specifically, since UEs have limited battery lives, they usually perform opportunistic local training for a FL. This means that a UE may not be able to perform a sufficient number of local training epochs, thereby reducing the accuracy of the obtained model. It thus is critical to minimize the energy consumption of UEs to ensure more UEs to participate in FL, so that highly-accurate global models can be obtained in the end. In addition, the workload of each UE also plays a vital role too. For instance, if a UE is already busy with computation-intensive tasks such as gaming and AR rendering, it may not be available to train its local data. Besides, the computing resource constraints on base stations or cloudlets in an MEC network may also influence the availability of UEs. For instance, if a UE is registered to a congested base station, the uploading delay of its trained local model may be high, which makes the UE be reluctant to participate in FL. Therefore, a fundamental problem for a service provider of the MEC network is to minimize the energy consumption of FL requests by considering the uncertain availability of UEs, such that the accuracy of each FL request is guaranteed while meeting the resource constraints on base stations and cloudlets in the MEC network.

Minimizing the energy consumption of FL poses several challenges. First, besides the energy levels of each UE, its availability of participating in FL can be determined by various side information (defined as *context*), such as its workload, available amount of computing resource, and the quality of its data. The complex interplays of these contexts make the availability of UEs uncertain. For example, even if a UE chooses to participate in a FL according to its current energy level, its local training may not perform sufficient training epochs due to its insufficient available computing resource. As a result, local training models with low accuracy will degrade the accuracy of the global model. To guarantee the accuracy of the global model while minimizing the consumed energy, how to harness the complex interplay among contexts of UEs via selecting UEs with accurate local models? Second, given the distributive and dynamic nature of UEs, it is vital to ensure that accurate local models be aggregated timely in each training round of a FL, considering that each training round has a fixed time length. Otherwise, if the aggregators are far from UEs, the local model may not be transmitted to the aggregators due to high transmission latency and the limited time of each round of training. This consequently leads to a low

accurate global model. Further, UEs are distributed at different locations of the MEC network, different local models of UEs have different levels of accuracy. We can distribute a number of aggregators to base stations close to the UEs with highly accurate local models. This however can quickly deplete the computing resource of base stations, as base stations are attached with light-weight computing units, such as FPGAs and neural network accelerators [27]. Consequently, the communication efficiency can be degraded if base stations are too congested to provide efficient aggregations. Thus jointly selecting a set of UEs and placing a proper number of aggregators to a set of strategic locations is critical to aggregate high-quality local models timely. To tackle the afore-mentioned challenges, we enable FL in an MEC network from the perspective of context-aware UE selection and efficient aggregator placement, such that the energy consumption is minimized without degrading the accuracy of FL.

B. Novelty and Contributions

To the best of our knowledge, we are the first to minimize the energy minimization for FL in an MEC network while guaranteeing the accuracy of the global model with uncertain UE availability, by building the accuracy of models that consider not only just energy levels of UEs but also various other context information. Most existing studies either focused on the energy consumption of UEs [52], [60], [63] or assumed that UEs are always available to participate in FL [40], [42], [54]. There are rarely studies of FL that aim to strive for the finest trade-off between accuracy and energy levels of UEs. The basic ideas for the mentioned problems are as follows. For a single FL request, we propose novel online learning methods via leveraging context correlations through adopting the contextual multi-armed bandit technique. In other words, each context group consists of the contexts with similar losses, which represents similar influences on the energy consumption of UEs. As a result, each context group has a representative context. The expert of each context group follows the recommendation of the representative group. On the other hand, for multiple FL requests, we leverage the dynamics of contexts and propose a method to adjust the loss ranges of each context group.

The main contributions of this paper are summarized as follows.

- We formulate the problem of energy minimization for FL in an MEC network with uncertain UE availability, by finding a trade-off between the energy consumption and accuracy of the global model.
- We propose a performance-guaranteed optimization framework for the defined problem with a single FL request, which consists of an algorithm based on bandits with correlated contexts for UE selections and an approximation algorithm with an approximation ratio for aggregator placement of FL in base stations or cloudlets.
- We devise an online learning algorithm with a bounded regret for the problem with multiple FL requests, by adopting an adaptive grouping method to dynamically learn the correlation among contexts.

- We evaluate the performance of the proposed algorithms for FL based on real datasets. Experimental results demonstrate that the performance of the proposed algorithms outperform their counterparts by reducing energy consumption at least 13% while achieving the similar accuracy.

The rest of the paper is arranged as follows. Section II summarizes the related studies on the topic. Section III introduces models and defines optimization problems. The proposed optimization framework for the problem with a single FL request is described in Section IV. Section V devises an online learning algorithm for the problem with multiple FL requests. Section VI provides some experimental results on the performance of the proposed algorithms. The paper is concluded in Section VII.

II. RELATED WORK

Recently, FL has gained increasing interest and attentions [10], [40], [41], [53]. Most existing studies focused on distributed learning architectures to promote the accuracy of FL while speeding up the FL convergence [10], [41], [53]. For example, several FL frameworks including FedAvg [40] and FedOpt [26] and the hierarchy framework [5] were introduced as key AI techniques in the edge [10], [40], [41], [53].

There are studies on enabling FL in wireless or MEC networks. Most of them focused on minimizing the computing cost [33], [40], delay [59], or optimizing parameter updating. For example, Yang et al. [59] aimed to minimize the delay of FL in wireless networks. Qian et al. [42] studied the privacy-aware service placement for FL in MEC networks to meet user privacy requirements. Zhan et al. [62] designed the incentive mechanism to motivate UEs for participating in FL via an optimal pricing strategy. Wang et al. [54] analyzed the convergence bound of the distributed gradient descent for FL to minimize the global loss. McMahan et al. [40] conducted empirical studies for FL by considering different model architectures and datasets, to minimize the communication cost. Xu et al. [58] studied the problem of aggregator placement and UE assignment for hierarchical FL in MEC networks, by proposing approximation and heuristic algorithms. He et al. [15] jointly considered the privacy-preserving and low-cost requirements of FL in an MEC network, by proposing new efficient scheme and algorithms with the objective of reducing model training time while meeting the privacy requirement of users. Wang et al. [55] focused on the decentralized federated learning, by adopting peer-to-peer communication without maintaining a global model. They proposed an efficient algorithm to accelerate DFL by integrating optimization of topology construction and model compression. Houda et al. [16] proposed a cooperative framework to secure IoT applications by jointly leveraging the techniques of FL and game theory. Song et al. [51] focused on the user-level privacy issue and presented an attack framework based on attacks from malicious servers. Sun et al. [47] proposed a contract-based incentive mechanism to customize the payment for each participating worker considering personalized privacy preference. Chen et al. [8] investigated the problem of improving the performance of FL in wireless networks by considering user selection and wireless resource allocations. Most of these studies did not

consider the energy consumptions of both UEs and cloudlets. Conventional studies on energy consumptions usually focus on Internet-of-Things (IoT) networks with IoT devices performing sensing or data collection tasks, most of them do not consider FL requests [3], [34], [35]. Direct adopting such methods for the problems in this paper cannot efficiently implement FL requests.

Although works in [36], [52], [60] studied the energy minimization of FL, they did not consider the problem of UE selection and aggregator placement in an MEC, they only investigated the energy consumption of UEs. For instance, Yang et al. [60] aimed to minimize the learning and communication energy consumptions of UEs while meeting the latency requirement of a FL request. They however focused on solely the energy consumption of UEs, and ignored the energy consumption due to the model aggregation. Tran et al. [52] considered the similar optimization objective of minimizing energy consumption, where all UEs unconditionally transmit their learning results to base stations, through determining the optimal CPU-cycle frequency that can be adopted by local training in each UE. Li et al. [36] designed a flexible communication compression scheme to minimize the energy consumption of all UEs. Zhan et al. [63] improved the energy efficiency of FL by balancing CPU-cycle frequency of UEs. A closely related study with ours is conducted by Kim et al. [19], they tackled the energy efficiency of FL from the perspective of dataset and computation management by striving for a trade-off of the learning efficiency and the overall energy consumption of UEs. Although their study investigated the relation of accuracy and energy consumption, the user selection and aggregator placement are not the major concerns in their study. The dependency of energy levels on various contexts did not explore either. Huang [17] investigated the uncertainty of training and reporting phases of FL in MEC networks, by proposing a proactive context-aware FL with state prediction and decision-making based on multi-armed bandits. However, they did not consider the aggregator placement in their paper.

III. PRELIMINARY

In this section, we first describe the system model, notions and notations. We then formulate the problem precisely.

A. System Model

We consider an MEC network $G = (\mathcal{BS} \cup \mathcal{CL}, E)$ with a set \mathcal{BS} of base stations and a set \mathcal{CL} of cloudlets, as shown in Fig. 1. Let $bs_i \in \mathcal{BS}$ be a base station, which provides wireless connections for a set of UEs within its transmission range. Each base station bs_i can execute AI applications in its attached neural network accelerators, such as Intel Neural Stick 2 [30]. Denote by $cl_j \in \mathcal{CL}$ a cloudlet, which is located in the backhaul of the MEC network consisting of edge servers with CPU and GPU resources. Due to the size and space limitation of base stations and cloudlets, both computing and bandwidth resources in them are capacitated. Let $C(bs_i)$ and $C(cl_j)$ be the computing resource capacities of base station bs_i and cloudlet cl_j , respectively. Also, $B(bs_i)$ and $B(cl_j)$ represent the bandwidth resource capacities of bs_i and cl_j , respectively. The base stations and cloudlets are

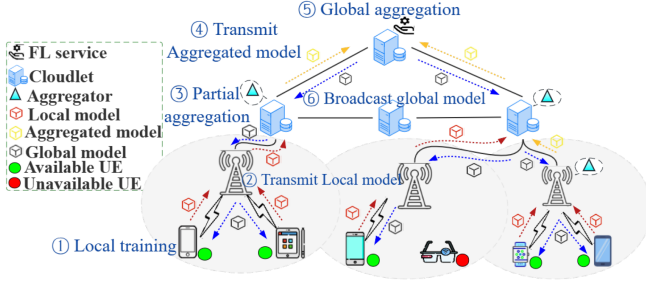


Fig. 1. An example of FL in an MEC network.

interconnected by links/paths in E . Each UE generates a dataset and trains a learning model on its dataset locally, instead of transmitting its dataset to a central site for training. Let ds_k be the dataset of UE ue_k . An example of FL in the MEC network is shown in Fig. 1.

B. FL Requests, Local Training, and Aggregation Framework

Users of the MEC network issue multiple *FL requests* to train the datasets that are distributed in UEs. Note that each FL request aims to obtain a single accurate machine learning model; therefore, we assume that they are independent and not correlated. The implementation of a FL request is referred to as a *FL process*, which consists of multiple rounds of local training and aggregation, until the global model converges to a certain accuracy.

Each UE performs a number of epochs of local training in each round of FL to obtain a local model. We adopt a L -Lipschitz continuous and γ -strongly convex loss function for the local training of each FL process [59], [60]. In particular, let A be the accuracy gap between the accuracy of the current epoch with the targeted accuracy. A can be obtained if at least a number

$$\lambda = \left\lceil \frac{2}{(2 - L\psi)\psi\gamma} \log_2(1/A) \right\rceil \quad (1)$$

of epochs is performed at a UE, where ψ is the learning rate and $\psi < 2/L$ [60]. Note that the number of epochs is a lower bound, which represents the minimum number of epochs needed to obtain the target accuracy A . We use it to approximate the number of epochs each UE needs to perform local training at each round. Besides, due to the heterogeneity of computation capabilities and workloads, the amount of time needed for each UE performing an epoch varies and dynamically changes. Specifically, given that each round of local training usually has a fixed length of time, each round of FL consists of a local training epoch, local model updating and aggregation. After local training, the trained local model has to be uploaded to its aggregator for aggregation before the round expires [5], [37], [65]. In particular, when the length of a round is very small or the local training takes too much time, the trained local models have to be uploaded in real time. Considering that the availability of a UE is not known in advance, whether it can update its trained local model in each round of FL on time is uncertain as well.

The trained local models in UEs are required to be aggregated to obtain a global model for each FL request. Following the study in [5], we consider a hierarchy aggregation framework of FL [38], which consists of a set of *aggregators* and a FL service, as shown in Fig. 1. In each round, a number of UEs perform local training, and each *aggregator* is responsible to collect the local models from a number of UEs. Let $A_{m,o}$ be the o th aggregator of request r_m . The aggregated models of all aggregators are sent to FL service S_m for final aggregation. In the next round, the FL service broadcasts the current global model to all UEs for further training. Let S_m be the FL service, and denote by r_m a FL request that requires S_m .

Due to resource capacity constraints on the MEC network, we consider that the aggregators of each FL request r_m can be distributed at different locations of the network, such as base stations or cloudlets. By distributing multiple aggregators into locations within the proximity of UEs, the local models can be aggregated in their close locations, thereby avoiding sending them around.

C. Energy Consumption

The service provider of the MEC network wants to minimize its cost of provisioning FL services while encouraging more UEs participating in FL processes. To this end, the energy consumption of UEs, cloudlets and base stations needs to be minimized, which are defined in the following.

Each UE consumes its energy to perform local training and transmit its trained local model to an aggregator in each round of each FL process. On one hand, in each round of implementing a FL request, UE ue_k needs to perform a necessary number of local training epochs to obtain a local model with required accuracy, computing energy thus is consumed in it. However, each UE ue_k may need different lengths of time to perform an epoch of local training for different FL requests. Denote by $t_{m,k}$ the time that ue_k used to train its dataset ds_k for an epoch of local training for FL request r_m . Notice that $t_{m,k}$ is uncertain as it depends on varying factors, including the local training minibatch size and the energy status of the ue_k . Let P_k^{comp} be the processing power of UE ue_k . The energy consumption of local training in ue_k for each round of implementing FL request r_m is

$$e_{m,k}^{local} = x_{m,k} \cdot t_{m,k} \cdot P_k^{comp} \cdot \lambda, \quad (2)$$

where λ is the necessary number of epochs to obtain a local model with a targeted accuracy (as shown in (1)), and $x_{m,k}$ is an indicator decision variable that shows whether UE ue_k participates the implementation of FL request r_m . On the other hand, each UE ue_k needs to update its trained local model to an aggregator for aggregation and receives the updated global model from FL service S_m , which consumes transmission energy. Let $e_{k,i}^{upt}$ be the energy consumption of UE ue_k for updating its local model w_m via bs_i . Denote by w_m the local model trained in a UE for FL request r_m . We then have

$$e_{k,i}^{upt} = y_{m,i,k} (2 \cdot |w_m| / R_{k,i}) P_k^{comm}, \quad (3)$$

where $y_{m,i,k}$ is an indicator variable shows whether the trained model of ue_k of request r_m is uploaded via base station bs_i ,

P_k^{comm} is the data transmission power of the ue_k that participates in r_k and $R_{k,i}$ is the achieved data rate of the wireless channel between ue_k and bs_i , which can be calculated by Shannon formula [29].

The energy consumptions of base stations and cloudlets of the MEC network are mainly due to model aggregations for different FL requests [39]. Let location Loc_q be a potential location for placing an aggregator, which can be either a base station or a cloudlet.

Following existing studies [22], [28], the energy consumption of a neural network processing unit is proportional to the rate of aggregation, which is also proportional to the amount of data to be aggregated and the resource allocated to process a unit amount of data [28], [61]. Let κ_q be the amount of computing resource assigned to aggregate a unit amount of data in location Loc_q . The latency $\delta_{m,o,q}$ of executing aggregation task in location Loc_q is

$$\delta_{m,o,q} = z_{m,o,q} (|UE_{m,o}| \cdot |w_m|) \beta_q \cdot \kappa_q, \quad (4)$$

where $UE_{m,o}$ is the set of UEs that are assigned to aggregator $A_{m,o}$ of FL request r_m and have trained parameters to update, β_q is a constant identifying how long the aggregator at location Loc_q takes to aggregate a unit amount of data, and $z_{m,o,q}$ is a binary indicator variables shows whether there is an aggregator of FL request r_m in location Loc_q . Let $e_{m,o,q}^{agt}$ be the energy consumed due to executing the aggregation task of request r_m in Loc_q , consisting of the energy consumption of its neural network processing unit, idle power, and leakage power. That is,

$$e_{m,o,q}^{agt} = z_{m,o,q} \cdot \delta_{m,o,q} \cdot \left(\frac{\xi_j \cdot |A_{m,o}|}{\delta_{m,o,q}} P_q^{max} + P_q^{idle} + P_q^{leak} \right), \quad (5)$$

where ξ_j is used to calculate the access rate of neural network processing units of cloudlet cl_j as shown in [22], P_q^{max} is the maximum power of all processing units of location Loc_q , P_q^{idle} is the idle power, P_q^{leak} is the leakage power of the GPU units at location Loc_q , and $|A_{m,o}|$ is the number of instructions of the aggregation task of request r_m in Loc_q .

D. Resource Consumption of Implementing FL Requests

Implementing FL requests in MEC networks consumes both computing and bandwidth resources. Since each service S_m is already running in the system, we assume that its resource has been preserved. Instead, we focus on the resource consumptions of aggregators distributed in the MEC network.

The computing resource consumed due to the aggregation of trained local models by UEs in $UE_{m,o}$ that are assigned to aggregator $A_{m,o}$ is proportional to the amount of data that needs to be aggregated [12], [60], i.e.,

$$CR_{m,o,q} = z_{m,o,q} \cdot \kappa_q \cdot |UE_{m,o}| \cdot |w_m|. \quad (6)$$

Given the computing capacity C_q on each location Loc_q and a set R of FL requests, we then have

$$\sum_{r_m \in R} CR_{m,o,q} \leq C_q. \quad (7)$$

for each $Loc_q \in \mathcal{CL} \cup \mathcal{BS}$.

Bandwidth resource is consumed in order to transfer the local models of UEs and aggregated models to service S_m . Let ω_q be the amount of bandwidth resource used to transfer a unit amount of data. The bandwidth resource needed to transfer $|w_m|$ amount of trained model of each FL request r_m thus is $\omega_q \cdot |w_m|$. The bandwidth requirement can be modeled as

$$\sum_{r_m \in R} BR_{m,o,q} \leq B_q, \quad (8)$$

where $BR_{m,o,q} = z_{m,o,q} \cdot \omega_q \cdot |UE_{m,o}| \cdot |w_m|$ and B_q is the bandwidth resource capacity of location Loc_q .

E. Problem Formulation

Given an MEC network $G = (\mathcal{BS} \cup \mathcal{CL}, E)$, we consider efficient and highly-accurate FL in the network, by implementing a set R of FL requests in FL services provisioned by a service provider. Each request $r_m \in R$ requires to train a global model with a guaranteed accuracy. We formulate the following two optimization problems.

Problem 1. Assuming that the UE availability is uncertain, the energy minimization problem for a single FL request in an MEC network is to jointly select a number of UEs and find an appropriate number of locations for aggregators in each round of the FL process for the FL request, such that the total energy consumption of base stations, cloudlets and UEs is minimized, subject to their resource capacities and the accuracy requirement of the FL request.

The energy minimization problem for a single FL request problem can be formerly formulated as follows.

$$\mathbf{P1} : \quad \min \left(e_{m,k}^{local} + e_{k,i}^{upt} + e_{m,o,q}^{agt} \right), \quad (9)$$

subject to the resource constraints (7) and (8) on each location $Loc_q \in \mathcal{BS} \cup \mathcal{CL}$, and the following constraints,

$$CR_{m,o,q} \leq C_q, \quad \forall Loc_q \quad (10)$$

$$BR_{m,o,q} \leq B_q, \quad \forall Loc_q \quad (11)$$

$$\sum_{bs_i \in \mathcal{BS}} y_{m,i,k} = 1, \quad \forall ue_k \quad (12)$$

$$x_{m,k}, y_{m,i,k}, z_{m,o,q} \in \{0, 1\}, \quad (13)$$

where constraints (10) and (11) ensure that the computing and bandwidth capacities on each location Loc_q must be met. Constraint (12) indicates that the uploading of each local model can only be forwarded via a single base station. Constraints (13) guarantee that the indicator variables should be either ones or zeros.

Problem 2. In reality, FL requests arrive into the system dynamically, and the admissions of requests may impact the availability of UEs. Assuming that all requests in R can be implemented by the MEC network, the energy minimization problem for multiple FL requests in an MEC network G is to dynamically implement each FL request in R , jointly select a number of UEs and find an appropriate number of locations for aggregators in each round of the FL process for each FL request, such that the total energy consumption of base stations,

TABLE I
 SYMBOLS

Symbols in system model	Meaning
$G = (\mathcal{BS} \cup \mathcal{CL}, E)$	an MEC network with a set \mathcal{BS} of base stations and a set \mathcal{CL} of cloudlets.
bs_i and cl_j	a base station in \mathcal{BS} and a cloudlet in \mathcal{CL} .
$C(bs_i)$ and $C(cl_j)$	the computing resource capacities of base station bs_i and cloudlet cl_j , respectively.
$B(bs_i)$ and $B(cl_j)$	the bandwidth resource capacities of base station bs_i and cloudlet cl_j , respectively.
Loc_q	a potential location for placing an aggregator, which can be either a base station or a cloudlet.
E	a set of backhaul links interconnecting the base stations and cloudlets.
ue_k, ds_k	an UE and its dataset, where k is a constant with $1 \leq k \leq K$.
λ	the number of epochs of local training in each round of FL.
r_m and S_m	a FL request and a FL service.
$A_{m,o}$ and $ A_{m,o} $	the o th aggregator of FL request r_m and the number of instructions of the aggregation task of request r_m in Loc_q .
P_k^{comp}	the processing power of the ue_k .
P_k^{comm}	the data transmission power of the ue_k .
$t_{m,k}$	the time that ue_k used to train its dataset ds_k for an epoch of local training for FL request r_m .
w_m and $ w_m $	the local model trained in a UE for FL request r_m and the size of the local model.
$R_{k,i}$	the achieved data rate of the wireless channel between ue_k and bs_i .
$e_{m,k}^{local}$	the energy consumption for local training of r_m in ue_k .
$e_{k,i}^{upt}$	the energy consumption for updating local model w_m via bs_i in ue_k .
$e_{m,o,q}^{agt}$	the energy consumption for executing the aggregation task of request r_m in Loc_q .
$\mathcal{UE}_{m,o}$	the set of UEs that are assigned to aggregator $A_{m,o}$ of FL request r_m .
\mathcal{A}_m	the set of deployed aggregators for FL request r_m .
κ_q	the amount of computing resource assigned to aggregate a unit amount of data in Loc_q .
ω_q	the amount of bandwidth resource used to transfer a unit amount of data in Loc_q .
$\delta_{m,o,q}$	the latency of executing aggregation task in Loc_q .
ξ_j	the access rate of neural network processing units of cloudlet cl_j .
$CR_{m,o,q}$	the amount of computing resource need by the set of UEs that are assigned to aggregator $A_{m,o}$.
Symbols in algorithms	Meaning
θ, Θ, Θ_g and $\Theta_{w,t}$	a context, a collection of contexts, the set of representative contexts of all context groups, and the representative context of group g_w in round t .
$l_{k,t}$ and $\tilde{l}_{k,t}$	the loss of an arm (a.k.a. ue_k) if the arm is played in round t and the estimated loss of an arm according to context θ .
$\tilde{l}_{k,t}^{cum}$	the estimated cumulative loss of the ue_k in round t .
g_w and $ g_w $	a context group and the number of contexts in group g_w .
\mathcal{P}_t^θ	the probabilities of selecting UEs according to the θ th context.
$p_{\theta,t}^w$	the probability that expert of group g_w selects the θ th context as the representative context in round t .
η_t	the learning rate of the algorithm OL_MAB in round t .
$p_{k,t}$	the probability of selecting UE ue_k .
$L_{w,\theta,t}$ and $L_{w,\theta,t}^{cum}$	the losses of the θ th context and the accumulative loss of context θ of group g_w in round t .
$\tilde{L}_{w,\theta,t}^{cum}$	the estimated cumulative loss of context θ of group g_w in round t .
$L_{w,t}$ and $\tilde{L}_{w,t}$	the losses of an expert and the estimated loss of an expert of group g_w in round t .
RS_q	the number of resource splits in Loc_q .
$s_{z,q}$	the z th resource split of location Loc_q .
ζ and σ	the increment that each expert uses to expand its lose range, and the probability that an expert expands its loss range.

cloudlets and UEs is minimized, subject to resource capacities in the MEC network and the accuracy requirement of each admitted FL request.

In other words, the energy minimization problem for multiple FL requests can be formulated as follows.

$$\mathbf{P2}: \quad \min \sum_{r_m \in \mathbf{R}} \left(e_{m,k}^{local} + e_{k,i}^{upt} + e_{m,o,q}^{agt} \right), \quad (14)$$

subject to the resource constraints (7) and (8) on each location $Loc_q \in \mathcal{BS} \cup \mathcal{CL}$, and the following constraints,

$$\sum_{bs_i \in \mathcal{BS}} y_{m,i,k} = 1, \quad \forall ue_k \quad (15)$$

$$x_{m,k}, y_{m,i,k}, z_{m,o,q} \in \{0, 1\}. \quad (16)$$

The differences of *Problem 1* and *Problem 2* are given in the following. In *Problem 1*, we deal with a single FL request by selecting UEs and placing aggregators for the FL request. Difference UEs may have different availabilities in each round of the FL request, the selecting of appropriate numbers of UEs is thus important to ensure that the FL learn an accurate model. Therefore, the problem aims to capture the impact of various

factors on the availabilities of UEs, such that UEs with high quality local model are selected for the FL request. However, there may be multiple FL requests arriving into the system, and the admission of currently-arrived requests influence the implementation of future FL requests, as the resource availabilities in base stations and cloudlets change dynamically. *Problem 2* thus is to investigate the impacts of request admissions on the performance of FL training in an MEC network with limited resource availability.

For clarity, the symbols used in this paper are summarized in Table I.

IV. A PERFORMANCE-GUARANTEED OPTIMIZATION FRAMEWORK BASED ON CONTEXTUAL BANDITS FOR A SINGLE FL REQUEST

In this section, we propose an efficient optimization framework with performance guarantees for the energy minimization problem for a single FL in an MEC network, with the aim of minimizing the energy consumption while meeting resource capacity constraints of the MEC network.

A. Motivation and Basic Idea

To guarantee the accuracy of the learnt global model of a FL request r_m , the key is to select UEs that can train their local models for a sufficient number of local training epochs. However, the UE availabilities depend on various contexts of the MEC network, including the energy level of each UE, data quality, the available computation capacity of each UE, the achieved data rate of the wireless channel between the UE and its base station, etc. In particular, the energy status of UE ue_k significantly impacts on time $t_{m,k}$ of each epoch of local training. A UE with low energy levels may need more local training epochs that are greater than the length of a FL round, to train a model with the required accuracy. In addition, the data quality of each UE will affect the convergence time of the FL because the higher the data quality (e.g., iid), the faster convergence of the FL model [18]. Therefore, these contexts should be jointly considered in the UE selection for implementing a FL request.

We also observe that the contexts of different UEs are highly correlated. For instance, the low energy level of a UE usually leads to a reluctant participation of FL processes or slow pace of training. Also, low quality data will increase the energy consumption of local training. Such correlated contexts jointly affect the accuracy of each FL as they eventually have impact on the time of each epoch of each UE. In extreme cases, a UE may not be able to upload its local model for aggregation in each round due to that it takes a long time of local training epochs. We thus propose an online learning algorithm via contextual bandits, by exploring the correlations among the contexts to predict time $t_{m,k}$ of each local training epoch for FL request r_m in UE ue_k . The novelty of algorithm OL_MAB is to leverage the correlations among contexts by forming context groups according to the loss ranges, unlike conventional context-aware MAB method.

B. An Online Learning Algorithm for the UE Selection Based on Bandits With Correlated Contexts

We now devise an online learning algorithm for correlated contexts, by considering each UE ue_k as an arm. There is a *loss* value of playing each arm (selecting a UE), which is defined as the energy consumption of UE ue_k , cloudlets and base stations, due to admitting UE ue_k for local training. Denote by $l_{k,t}$ the loss of an arm (a.k.a, a UE) if the arm is played in round t of algorithm OL_MAB. Without loss of generality, we assume that the loss $l_{k,t}$ of each UE is normalized in the range of $[0, 1]$.

The key of the proposed online learning algorithm is to jointly learn the correlations among contexts and the UE choices (arms). We adopt a multi-level non-stochastic online learning method without stochastic distributions on the contexts and UEs. Specifically, we assume that different contexts with similar influences on the UE choices are grouped into a *context group*. The reason is that there may be a group of factors having similar influences on the availability of UEs. For instance, the energy levels of UEs may depend on the workload of UEs [2], [25], [49]. Namely, when the workload of a UE is high and executing computing-intensive tasks, it may deplete its energy quickly, thereby decreasing its probability of participating a FL process [5], [48]. There is an expert in each context group to select

the most influencing context, referred to as a *representative context*, and recommends UEs according to the representative context, as shown in Fig. 2.

The online learning algorithm is described as follows.

Context Groups. Given a collection Θ of contexts $\{\mathcal{S}_\theta : 1 \leq \theta \leq |\Theta|\}$, each context θ has a different value in different rounds, and thus recommends different UEs for each FL request. Specifically, in each round t of implementing a FL request, let \mathcal{P}_t^θ be the probabilities of selecting UEs according to the θ th context, i.e., $\mathcal{P}_t^\theta = \{p_{1,t}^\theta, \dots, p_{k,t}^\theta, \dots, p_{|UE|,t}^\theta\}$ with $1 \leq k \leq |UE|$.

Correlated contexts may have similar probabilities of recommending UEs. We then divide the contexts into different context groups. The expert of each group g_w chooses a representative context and recommends UEs according to the probability of its representative context. Intuitively, the context with the minimum loss is preferred. However, selecting the context with the minimum loss may miss opportunities of exploration in an exploitation-and-exploration procedure. Also, the minimum loss of the current round may not lead to a minimum loss in future. We thus consider that each expert of group g_w dynamically learns the probability of the chosen representative context. Each context in the beginning has the a uniform probability to be chosen. Let $p_{\theta,t}^w$ be the probability that expert of group g_w selects the θ th context in round t of the algorithm, and $\mathcal{P}_t^w = \{p_{\theta,t}^w | 1 \leq \theta \leq |g_w|\}$ with $|g_w|$ denoting the number of contexts in group g_w . Each expert dynamically adjusts its own probabilities of selecting contexts to minimize its loss. Specifically, in each round, the expert of g_w updates its probability according to the losses of contexts in the group. The rationale behind is to adopt an exponential weighting method [11], [43]. Let $L_{w,\theta,t}^{cum}$ be the accumulative loss of context θ of group g_w in round t , $p_{\theta,t}^w$ then is updated by

$$p_{\theta,t}^w = \frac{\exp(-\eta_t L_{w,\theta,t}^{cum})}{\sum_{1 \leq k \leq |g_w|} \exp(-\eta_t L_{w,k,t}^{cum})}$$

where η_t the learning rate of the algorithm in round t .

UE Selection. Once the expert selected a representative context θ , it will recommend UEs according to the probabilities of context θ , i.e., the probabilities of \mathcal{P}_t^θ of θ . Let Θ_g be the set of representative contexts of all context groups. The probability of selecting UE ue_k is the average of the corresponding probability of all representative contexts in Θ_g , i.e.,

$$p_{k,t} = \mathbb{E}_{k \sim \mathcal{P}_t^\theta} p_{k,t}^\theta, \quad \forall \theta \in \Theta_g. \quad (17)$$

Then, each UE ue_k is randomly selected with probability $p_{k,t}$.

Loss and Probability Updating Policies. Let $L_{w,\theta,t}$ and $L_{w,t}$ be the losses of the θ th context and expert of group g_w in round t , respectively.

The loss of a context is the sum of the losses of the selected UEs, if it is selected as a representative context. Note that the unselected contexts and UEs do not contribute its loss at the current round t of a FL process. Let UE_t be the set of selected UEs in round t , the estimated loss of an arm (UE) according to context θ is

$$\tilde{l}_{k,t} = (l_{k,t} / (p_{k,t} \cdot \mathbb{E}_{\theta \sim \mathcal{P}_t^w} p_{\theta,t}^w)) \mathbb{1}_{k \in UE_t}, \quad (18)$$

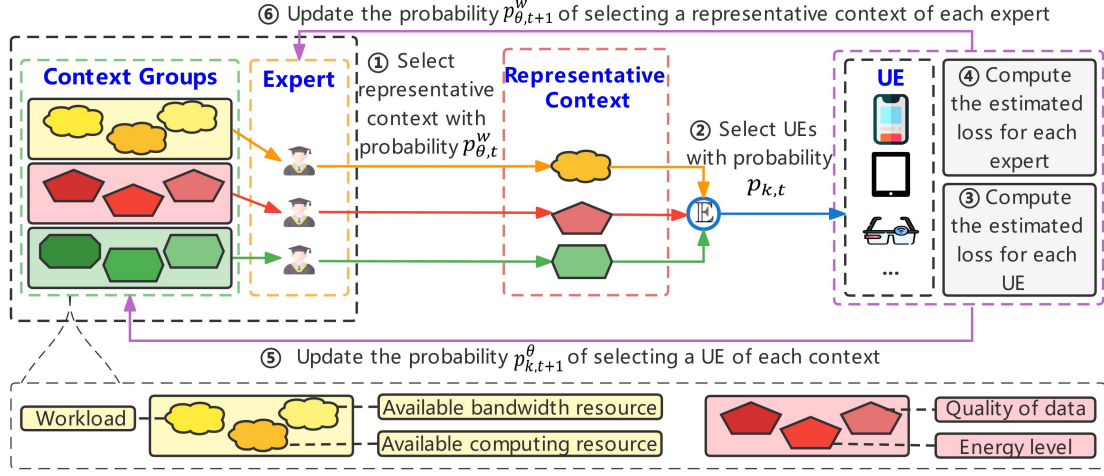


Fig. 2. An illustration of the steps of the proposed online learning algorithm for the UE selection based on bandits with correlated contexts.

where $\mathbb{1}_{k \in \mathcal{UE}_t} = 1$ if $k \in \mathcal{UE}_t$. The rationale behind is to construct an unbiased forecaster that estimates the loss of selecting an arm. Such an unbiased forecaster needs to jointly consider the expected probability of selecting a context and choose an arm according to the representative context. Once the UEs are selected according to context θ , we update the probabilities of selecting UEs of context θ by

$$p_{k,t+1}^{\theta} = \frac{\exp(-\eta_t \tilde{l}_{k,t}^{cum})}{\sum_{1 \leq k' \leq |\mathcal{UE}|} \exp(-\eta_t \tilde{l}_{k',t}^{cum})}, \quad (19)$$

where $\tilde{l}_{k,t}^{cum}$ is the estimated cumulative loss of ue_k in round t .

The loss of an expert depends on which context in the group is selected as the representative context. Since each context belongs to a single context group and only a single context is selected as a representative context of each group, the estimated loss of an expert is

$$\tilde{L}_{w,t} = (L_{w,\theta,t} / p_{\theta,t}^w) \mathbb{1}_{\Theta_{w,t} = \theta}, \quad (20)$$

where condition $\Theta_{w,t} = \theta$ indicates that context θ is selected as the representative context of group g_w and $L_{w,\theta,t}$ denotes the actual loss of context θ of group g_w in round t . We update the probabilities of selecting representative contexts. Let $\tilde{L}_{w,\theta,t}^{cum}$ be the estimated cumulative loss of context θ in group g_w in round t , we then have

$$p_{\theta,t+1}^w = \frac{\exp(-\eta_t \tilde{L}_{w,\theta,t}^{cum})}{\sum_{1 \leq \theta' \leq |g_w|} \exp(-\eta_t \tilde{L}_{w,\theta',t}^{cum})}. \quad (21)$$

The reason why we adopt these updating policies is that if the current loss of a UE is high, the probability of selecting the UE should be reduced. On the other hand, a UE with less loss may intend to have less loss in future, too.

Steps of the Algorithm. Given the contexts, context groups, and loss updating strategy, the problem is reduced to an exponential weighting algorithm with experts shown in [4], referred to as *Algorithm Exp4*. The basic steps of *Algorithm Exp4* consists

Algorithm 1: OL_MAB.

Input: An MEC network G , a set of UEs, a FL request r_m , and a finite time horizon T .

Output: The selected UEs for FL request r_m .

- 1: Let $(\eta_t)_{1 \leq t \leq T}$ be a nonincreasing sequence of real numbers;
 - 2: Initially, each UE is selected with a uniform distribution, i.e., $p_{k,0}^{\theta}$ for all UEs and contexts;
 - 3: Similarly, each context is selected by an expert with the same probability $p_{\theta,0}^w$;
 - 4: **for** $t \leftarrow 1, \dots, T$ **do**
 - 5: Each expert selects a representative context in its group g_w with probability $p_{\theta,t}^w$;
 - 6: Each UE ue_k (a.k.a. an arm) is selected with probability $p_{k,t}$, where $p_{k,t} = \mathbb{E}_{k' \sim \mathcal{P}_t^{\theta}} p_{k',t}^{\theta}$;
 - 7: Compute the estimated loss for each UE by (18);
 - 8: Compute the estimated loss for each expert by (20);
 - 9: Update the probability of selecting an UE of each context θ by (19);
 - 10: Update the probability of selecting a context in group g_w by (21);
 - 11: **end for**
-

of (1) collecting the probabilities of all representative contexts; (2) calculating the expected probability of selecting each UE, according to the collected probability sets of all experts; (3) selecting UEs according to the expected probabilities in step (2); and (4) updating the losses of both the selected UEs, representative contexts and experts. The detailed algorithm is given in *Algorithm 1*, referred to as *Algorithm OL_MAB*.

C. An Approximation Algorithm With Selected UEs

So far, we have shown how to select UEs for a FL process in an MEC network. We now place the proper number of aggregators to collect trained local models from UEs and send them to the FL service for the final aggregation.

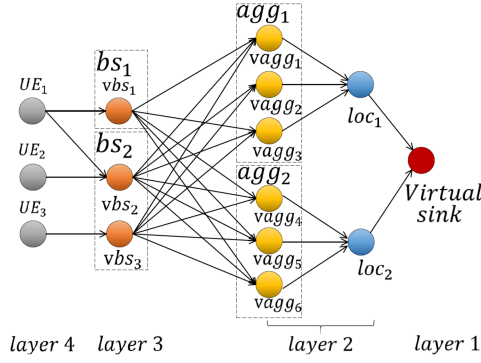


Fig. 3. An example of the auxiliary graph $G' = (V', E')$.

It must be mentioned that each base station or cloudlet has capacitated resource for local model aggregations. Given the available computing $C(Loc_q, t)$ and bandwidth resources $B(Loc_q, t)$ in each round t , we divide each location Loc_q into a number of ‘resource splits’ with each resource split having the computing and bandwidth resources to process a single local model w_m . Specifically, each location Loc_q has

$$RS_q = \left\lfloor \min \left\{ \frac{C(Loc_q, t)}{|w_m| \max_q \kappa_q}, \frac{B(Loc_q, t)}{|w_m| \max_q \omega_q} \right\} \right\rfloor \quad (22)$$

numbers of resource splits. Let $s_{z,q}$ be the z th resource split of location Loc_q . Each resource split is required to have sufficient amount of resource to aggregate a single local model. It then is crucial to determine the amount of resource in each resource split to be allocated to aggregate a single local model. Recall that the total size of local models of the UEs that are assigned to an aggregator for aggregation determines its resource demands. We thus use a *virtual aggregator* to represent action of aggregating a local model, and the resource demand of a virtual aggregator is calculated by

$$|w_m| \cdot \max_q \kappa_q \quad (23)$$

of computing resource and the amount

$$|w_m| \cdot \max_q \omega_q \quad (24)$$

of bandwidth resource. The rationale behind is that the trained model of each UE can be assigned to a location with enough computing resource for its aggregation.

Given RS_q resource splits of each location Loc_q and the concept of virtual aggregators, we reduce the problem of aggregator placements to an unsplittable minimum-cost multicommodity flow problem in an auxiliary graph $G' = (V', E')$, illustrated in Fig. 3. The rationale behind the construction of the auxiliary graph is to jointly determine the assignments of UEs to base stations and place aggregators to locations in $BS \cup CL$ without violating the resource capacities of locations. To this end, we first construct a layered auxiliary graph and distribute such decision makings to different layers of the graph. In particular, the auxiliary graph G' has 4 layers:

- *Layer 1* contains a *virtual sink* that represents the location of service S_m FL request r_m , denoted by v_t .

Algorithm 2: Placement.

Input: An MEC network G , a set UE_t of selected UEs for model updating of a FL request r_m , and finite time horizon T .

Output: A number of aggregators and their locations for r_m .

- 1: **for** $t \leftarrow 1, \dots, T$ **do**
 - 2: Construct an auxiliary graph G' as shown in Fig. 3;
 - 3: Find a unsplittable minimum-cost multicommodity flow f in the constructed auxiliary graph, with each commodity having source ue_k , sink v_t , and demand 1;
 - 4: **for each** UE ue_k **do**
 - 5: Find its path $(ue_k, bs_{i,p}, s_{o,q}) \in f$;
 - 6: Assign ue_k to base station bs_i ;
 - 7: Send the trained model of ue_k to Loc_q for aggregation;
 - 8: For the trained model w_m that is sent to base station bs_i , assign amounts $|w_m| \cdot \min_q \kappa_q$ and $|w_m| \cdot \min_q \omega_q$ of computing and bandwidth resources;
 - 9: **end for**
 - 10: **end for**
-

- *Layer 2* is composed of resource splits of locations in $BS \cup CL$.
- *Layer 3* is considered as the access layer with base stations in BS that relay their received local models for aggregation. In particular, we create a number of *virtual base stations* for each base station bs_i . Each virtual base station provides wireless access for a single UE within the transmission range of bs_i . Let $bs_{i,p}$ be the p th virtual base station of bs_i with $1 \leq p \leq RS_q$.
- *Layer 4* has the selected UEs.

We add edges into G' by interconnecting nodes in different layers. The virtual root is connected to every resource split of *layer 2*. That is, there is an edge from virtual sink v_t to the o th resource split of each location Loc_q , i.e., $(v_t, s_{o,q})$. The weight of edge $(v_t, s_{o,q})$ is set to the energy consumption of Loc_q for processing an amount $|w_m|$ of the trained model of r_m , i.e., $w(v_t, s_{o,q}) = e_{m,o,q}^{agt}$. We interconnect the resource splits in *layer 2* with the base stations in *layer 3*. Specifically, there is an edge between a resource split of each Loc_q in *layer 2* and a virtual base station $bs_{i,p}$ of bs_i , and its weight is set to the energy consumption for bs_i to relay the trained model of w_m , i.e., $w(s_{o,q}, bs_{i,p}) = e_{k,i}^{upt}$. Afterwards, we connect each virtual base station in *layer 3* with UEs in *layer 4*. The weight of each edge is set to the energy consumption of ue_k for local training, that is, $w(ue_k, bs_{i,p}) = e_{m,k}^{local}$.

We then find an unsplittable minimum-cost multicommodity flow in G' . Specifically, there is a ‘commodity’ for each UE with source ue_k , sink v_t , and demand 1. Let f be the obtained unsplittable flow. If path $(ue_k, bs_{i,p}, s_{o,q}) \in f$, ue_k will update its trained model via base station bs_i for the aggregation in Loc_q . The detailed steps are shown in *Algorithm 2*, referred to as *Algorithm Placement*.

D. Algorithm Analysis

In the following, we analyze the regret incurred by the online learning algorithm `OL_MAB`. We consider an oblivious adversary in the online learning process, where experts select representative contexts and make recommendations, the regret is defined as the maximum regret of all experts.

Let $R(T, w)$ be the regret of an expert of group g_w in time horizon T , we have its expectation $\mathbb{E}(R(T, w))$ by

$$\mathbb{E}(R(T, w)) = \mathbb{E}\left(\sum_{t=1}^T L_{w, \Theta_{w,t}, t}\right) - \min_{\theta=1, \dots, |g_w|} \mathbb{E}\left(\sum_{t=1}^T L_{w, \theta, t}\right). \quad (25)$$

Given W groups, the definition of regret is

$$R(T) = \max_{w=1, \dots, W} \mathbb{E}(R(T, w)). \quad (26)$$

Theorem 1. Let \mathcal{C} be the set of all contexts and $|\mathcal{C}|$ be the number of contexts in \mathcal{C} . The regret of *Algorithm OL_MAB* is $\frac{|\mathcal{C}|}{2}T\eta^2 + \frac{\ln|\mathcal{C}|}{\eta}$ if $\eta_1 = \eta_2 = \dots = \eta_T = \eta$. The time complexity of `OL_MAB` is $O(T(\Theta + |\mathcal{U}E|))$.

Please see Appendix, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TMC.2023.3262829>, for the detailed proof.

Theorem 2. Given the selected UEs $\mathcal{U}E_t$ in a single time slot t , *Algorithm Placement* delivers a feasible solution to the energy minimization problem for a single FL request in an MEC network. The approximation ratio of the obtained solution is ϑ , and the number of admitted local models is $(0.075 - \epsilon)|\mathcal{U}E|$, where $\vartheta = \max\left\{\frac{\max_q \kappa_q}{\min_q \kappa_q}, \frac{\max_q \omega_q}{\min_q \omega_q}\right\}$, and $0 < \epsilon \leq 0.075$. Its time complexity is $O(|\mathcal{U}E| \cdot (|\mathcal{B}S| + |\mathcal{C}L|)^2)$.

Please see Appendix, available in the online supplemental material, for the detailed proof.

V. AN ONLINE LEARNING HEURISTIC FOR THE PROBLEM WITH MULTIPLE FL REQUESTS

In this section we consider the energy minimization problem for FL in an MEC network with multiple FL requests, where each FL request requires to train a global model via a FL process.

A. Algorithm Overview

We devise an efficient algorithm to implement FL requests in a set R . The admission of a FL request in the MEC network influences different contexts of the network, such as resource availabilities and energy statuses of UEs. Consequently, the contexts change dynamically with implementations of FL requests one by one. As such, the selected UEs in the previous contexts may not be optimal in the new context of the network. That is, each context may change its recommendation probabilities on UE selections dynamically with the implementation of FL requests in each round t . Unfortunately, simply admitting each request one by one by invoking algorithm `OL_MAB` may not be able to accurately capture such dynamics of contexts.

Our basic idea to capture the afore-mentioned accumulative impact of multiple FL requests on the contexts by proposing an adaptive online learning algorithm to dynamically admit FL requests, by allowing each context group in *Algorithm OL_MAB* to be dynamically updated according to the latest context information. In particular, we observe that the dynamics of each context group needs to maintain a certain level of stability and embrace exploring the optimality of decisions made by experts (analog to explore and exploit process). It must be mentioned that `OL_MAB` actually learns different aspects of the implementation of FL requests. On the other hand, we here focus on the dynamical changes of contexts, by dynamically updating the context group after the admission of each FL request.

B. An Online Learning Algorithm

We aim to classify contexts with similar losses on energy consumptions into a context group g_w , where w is the index of group g_w with $1 \leq w \leq W$. Initially, we divide the range of losses of contexts into W equal intervals, and the contexts whose losses lie in an interval w are considered as the members in context group g_w . As the admission of FL requests proceeds, the losses of contexts in each group g_w may change. To maintain a certain level of stability, we assume that the expert of group g_w monitors and controls the group dynamics of g_w . In particular, the expert of group g_w oversees the losses of its contexts and makes recommendations based on the observed losses. To this end, it adjusts the loss range of its context group so that only the context with losses in the range can be included in the group.

A simple method is to allow each expert to freely adjust its range of losses, such that contexts with lower losses may be included in its loss range. This however may lead to all experts converging to similar ranges in which contexts have the lowest losses. Also, this method does not favor the contexts that may have a sudden low loss in future, thereby missing the opportunities of minimizing the losses of experts. To avoid such cases, we limit the adjustment steps of each expert. Let $[LB_w, UB_w]$ be the range of losses that the contexts are in g_w . Note that this range is equally divided in the very beginning, and each expert is in charge of only one interval.

To allow a certain level of exploration, we enable each expert to expand its range by a given increment ζ , before the admission of each FL request. That is, in each round t , if the expert of g_w chooses to expand its range by ζ , its new range will be $[LB_w - \zeta, UB_w + \zeta]$. This will include more contexts into the context group of the expert. Thus, the expert will have more opportunities to explore ‘better’ contexts. Specifically, in each round t , each expert decides whether to expand its range with probability σ with $0 \leq \sigma \leq 1$. After determining the ranges of all experts, we obtain the context group. Note that such an expansion strategy may lead to the overlapping of loss ranges of experts, making a context being included into multiple context groups. The incurred benefit is leading some valuable contexts to be selected with higher probability by different experts.

Given the adjusted context groups decided by the experts, we admit FL requests one by one, by invoking the proposed algorithms `OL_MAB` and `Placement`. Each expert then updates

Algorithm 3: OL_MULTI.

Input: An MEC network G , a set of UEs, a set R of FL requests, and a finite time horizon.

Output: Implementing all requests in R .

- 1: Let $(\eta_t)_{1 \leq t \leq T}$ be a non-increasing sequence of real numbers.
- 2: Initially, each UE is selected with a uniform distribution, i.e., $p_{k,0}^\theta$ for all UEs and contexts;
- 3: **for** $t \leftarrow 1, \dots, T$ **do**
- 4: **for** $r_m \in R$ **do**
- 5: Each expert decides whether to expand its loss range by ζ with probability σ ;
- 6: Group contexts according to the loss range of each expert.
- 7: **if** $t = 1$ **then**
- 8: Each context in the group is selected by its expert with the same probability $p_{\theta,0}^w$;
- 9: **end if**
- 10: Execute steps from step 5 to step 10 in **Algorithm** OL_MAB;
- 11: Execute steps from step 2 to step 9 in **Algorithm** Placement to find a number of aggregators and their locations for FL request r_m ;
- 12: **end for**
- 13: **end for**

its loss range after the admission of a FL request, as this may impact the context significantly. The detailed steps are shown in *Algorithm 3*, referred to as *Algorithm* OL_MULTI.

Theorem 3. The regret of the *Algorithm* OL_MULTI is $\frac{|C|}{2}T\eta^2 + \frac{\ln|C|}{\eta} + |R|\sigma\chi\zeta$, where χ is a given constant with $0 < \chi \leq 1$, σ is the probability that an expert expands its loss range, and ζ is the increment that each expert uses to expand its lose range. Its time complexity is $O(|R| \cdot T(2\Theta + |\mathcal{UE}|))$.

Please see Appendix, available in the online supplemental material, for the detailed proof.

VI. EXPERIMENTS

In this section, we first evaluated the performance of the proposed algorithms for the energy minimization problem for FL in an MEC network. We then investigated the impacts of important parameters on the performance of different algorithms.

A. Parameter Settings

We consider an MEC network with 100 to 500 UEs, 5 to 15 base stations and cloudlets. The MEC network generated by tool GT-ITM [14]. We consider the following settings of contexts of UEs: the achieved data rate of the wireless channel between UEs and base stations varies within the range of [1, 5] Mbps [45], and the available computation capacity of all UEs is also sampling from the same uniform distribution within [50%, 200%] [24]. The computing and bandwidth resource capacities of a location are in the range of [1,000, 5,000] MHz and [5, 20] Mbps [42], respectively. The data transmission power of a UE and a base station is randomly withdrawn from [5, 33] dBm [13]. The amount

of computing resource assigned to aggregate a unit amount of data in a location is randomly drawn from [0.1, 0.3] MHz [1]. The maximum power of all processing units of location is randomly withdrawn from [60, 100] Watt and the idle power is 5 Watt [21]. The minibatch size is drawn from [5, 10] [32]. The learning rate η , L of the L -Lipschitz, ψ and γ of the γ -strong convex loss function are set to 0.5, 4, 0.1, and 2, respectively [60]. Our experiment is based on an open source framework FedML [23]. The number of global rounds is fixed at 150, and the number of FL requests is fixed to 10. Unless otherwise specified, the afore-mentioned parameters are considered as default parameters.

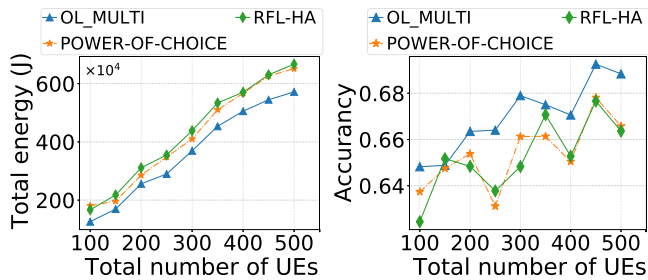
Models and Datasets. Each FL is performed based on datasets MNIST [31], FedEMNIST [44], and Shakespeare [6] with three different models: 1) Logistic regression on the MNIST dataset; 2) CNN on the FedEMNIST dataset; and 3) RNN on the Shakespeare dataset. We adopt FedAvg [40] and FedOpt [26] FL frameworks. By default, we use the FedAvg based on the FedEMNIST dataset. The training model is CNN, which uses the AlexNet architecture, and the model size $|w_m|$ is around 230M. Data samples are assigned to each UE uniformly.

Benchmarks. We compare the proposed algorithm OL_MULTI against the following algorithms.

- The first benchmark is the POWER-OF-CHOICE algorithm in [9], which speeds up the global model convergence by selecting UEs with higher local losses in each round of FL. For the sake of fairness, the number of UEs selected by algorithm POWER-OF-CHOICE is consistent with the number of UEs selected by the proposed UE selection of this paper in each round.
- The second benchmark is algorithm RFL-HA in [56], which divides UEs into K clusters with each cluster having an aggregator to aggregate its local models. Note that RFL-HA does not consider the UE selection, and all UEs can participate in the FL process. RFL-HA clusters UEs according to their data sizes, and uses an aggregator to collect the local model of UEs in each cluster. For the sake of fairness, the number of UEs selected by algorithm RFL-HA is consistent with the number of UEs selected by the our algorithms in each round.

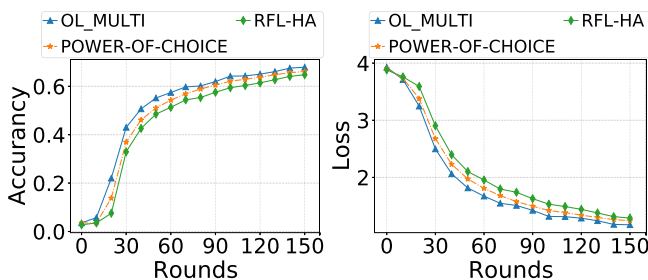
B. Results

Energy, Accuracy, and Convergence Speed. We first compared the performance of algorithm OL_MULTI against that of POWER-OF-CHOICE and RFL-HA in terms of total energy consumption and the average global accuracy of all FL requests, by varying the number of UEs from 100 to 500. Fig. 4 shows the results, from which we can see that algorithm OL_MULTI has around 13% and 15.5% lower energy consumption compared with those of POWER-OF-CHOICE and RFL-HA, while delivering the highest model accuracy. This is because OL_MULTI minimizes the energy consumptions of both UEs and cloudlets/base stations. In addition, OL_MULTI considers various contexts relating to energy consumption to harness the uncertain local training time of each UE. It also selects UEs with low energy consumption and necessary model accuracy to participate in each round of FL. Also, Fig. 4(a) indicates that the energy consumption of OL_MULTI increases with the



(a) The energy consumption of multiple FL requests by varying the total number of UEs. (b) The average global accuracy of multiple FL requests by varying the total number of UEs.

Fig. 4. The impact of the total number of UEs on the performance of algorithms OL_MULTI, POWER-OF-CHOICE and RFL-HA.

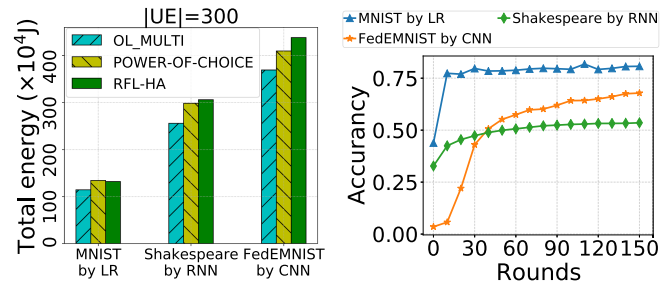


(a) The average global accuracy of multiple FL requests by different algorithms. (b) The average global loss of multiple FL requests by different algorithms.

Fig. 5. The accuracy and loss of algorithms OL_MULTI, POWER-OF-CHOICE and RFL-HA.

growth of the number of participating UEs. The reason is that more UEs imply more local trainings and aggregating a higher accurate local models, pushing up the energy consumptions of both UEs and cloudlets/base stations in the MEC network. Also, Fig. 4(b) shows that with the increase on the number of UEs, the accuracy of the global model fluctuates. The reason is that when the number of participating UEs reaches a certain number in the FedAvg, the influence of simply increasing its number on the global model training is trivial.

We then showed the accuracy and loss of algorithms OL_MULTI, POWER-OF-CHOICE, and RFL-HA, while fixing the number of UEs at 300. The results are shown in Fig. 5. We can see that OL_MULTI converges faster and obtains a higher accuracy than the other comparison algorithms. This is because OL_MULTI selects UEs that perform well in local training by algorithm OL_MAB, and finds a suitable aggregator for each UE by algorithm Placement. In contrast, algorithm POWER-OF-CHOICE selects UEs to participate in FL process based on higher local losses in each round of FL without considering the uncertain UE availability. This may lead some unavailable UEs with low energy to be chosen, and the failing of uploading their local models slows down the convergence and impacts the accuracy of the FL. Besides, algorithm RFL-HA takes the amount of data as the single metric of clustering without considering the aggregator placements. On one hand, the amount



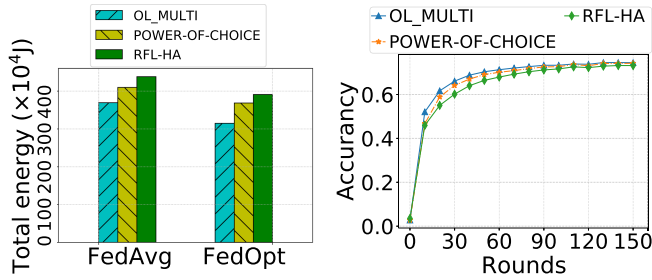
(a) The energy consumption of multiple FL requests by different FL models in different algorithms. (b) The average global accuracy of multiple FL requests by different FL models in OL_MULTI.

Fig. 6. The performance of Algorithms OL_MULTI, POWER-OF-CHOICE and RFL-HA based on different datasets to train FL models with different structures.

of data does not lead to high accurate models. On the other hand, the trained local models may not be uploaded successfully if the aggregators are placed far from UEs, which consequently reduces the convergence speed and accuracy.

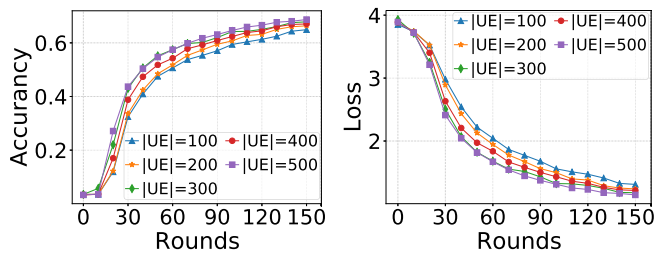
We third evaluated the energy consumption and convergence speeds of algorithms OL_MULTI, POWER-OF-CHOICE and RFL-HA based on different datasets to train FL models with learning algorithm of linear regression (LR), CNN, and RNN, by fixing the number of UEs to 300. Fig. 6(a) shows that algorithm OL_MULTI consumes the lowest amount of energy among the three mentioned algorithms no matter which learning algorithm is adopted. The reason is that algorithm OL_MULTI carefully finds a trade-off of the energy consumption and accuracy of FL, and its benchmark algorithms do not jointly consider energy and accuracy. Further, the gap of energy consumptions is enlarging if more complex learning algorithms are adopted. For instance, the gap of energy consumptions between algorithms OL_MULTI and RFL-HA under MNIST by LR is smaller than that under Shakespeare by RNN. The reason is that higher energy consumption leads to higher performance gaps, which depends on the complexity of performing a round of local training, and LR is the simplest one and consumes less energy for local training and aggregation than RNN and CNN. In addition, Fig. 6(b) shows the convergence speeds of algorithm OL_MULTI based on three applications scenarios with different datasets, from which we can see that OL_MULTI converges quickly no matter what learning algorithm is used for local training.

Performance in Real FL Frameworks. To show that algorithm OL_MULTI can be applied to real FL frameworks, we extend OL_MULTI to the FedOpt FL framework [26], which explores the interplay between the UE heterogeneity and communication efficiency [26]. Fig. 7 shows the results, from which we can see that our algorithm OL_MULTI consumes less energy than algorithms POWER-OF-CHOICE and RFL-HA while guaranteeing the accuracy of the global model. In addition, Fig. 7(a) shows that the overall energy consumption of the algorithms based on FedOpt is lower than FedAvg, because FedOpt improves communication efficiency, resulting in a low energy consumption due to data transmissions. Fig. 7(b) illustrates that

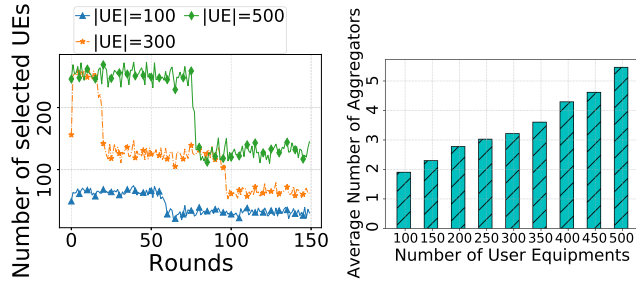


(a) The energy consumption of different algorithms under FedOpt and FedAvg. (b) The average global accuracy of different algorithms under FedOpt.

Fig. 7. The performance of algorithms OL_MULTI, POWER-OF-CHOICE and RFL-HA under FedOpt and FedAvg.



(a) The average global accuracy. (b) The average global loss.



(c) The number of selected UEs. (d) The average number of aggregators.

Fig. 8. The impact of the number of UEs on the performance of algorithm OL_MULTI.

the performance of OL_MULTI based on FedOpt. We can also see that algorithm OL_MULTI achieves higher accuracy and faster convergence than algorithms POWER-OF-CHOICE and RFL-HA.

Impact of Parameters. We finally investigated the impact of the number of UEs on the performance of algorithm OL_MULTI in terms of average global accuracy, loss, the number of selected UEs and the average number of aggregators. Fig. 8(a) and (b) plot the curves of accuracy and loss of CNN on the FedEMNIST dataset, from which we observe that the average accuracy of the global model generally increases gradually as the number of UEs increases. The reason is that more UEs imply that more highly available UEs can be selected to perform local training, which consequently contribute to highly accurate local models. The reason is that $|UE|$ represents the number of UEs that can be selected to participate in FL process. The larger the $|UE|$,

the larger the number of UEs selected by OL_MULTI, which means training more data and aggregating more local models in each round, leading to a higher accuracy. However, the accuracy increase is much slower when the number of UEs varies from 300 to 400. The rationale behind is that the FedAvg is adopted. As the number of UEs keeps increasing from 300 to 400, some relatively low-accuracy local models may also participate in the aggregation, thereby reducing the accuracy of the global model, such overfitting of the global model should be avoided.

Fig. 8(c) shows the number of selected UEs participating in FL by algorithm OL_MULTI in each round, when the number of UEs is set to 100, 300 and 500, respectively. We can see that a larger number of UEs are initially selected, and the number drops with the growth of the number of rounds of FL. The rationale behind is that algorithm OL_MULTI selects each UE with a uniform probability, and it updates the probability of selecting each UE progressively as the rounds of FL grow. Initially, a higher number of UEs are selected, as OL_MULTI selects a UE with a low probability and a higher number of UEs are needed to guarantee the accuracy. As the FL continues in the rest of rounds, some UEs are identified as those who could contribute to high accurate local models with low energy consumption. Therefore, the number of UEs drops as the FL proceeds.

Fig. 8(d) illustrates the number of aggregators needed in the MEC network. It can be seen that the number of aggregators placed increases with the increase on the number of UEs. The reason is that as more UEs participate in FL process, more local models need to be aggregated, so more aggregators need to be placed in the MEC network to collect trained local models. In addition, when the number of UE increases from 450 to 500, the number of aggregators increases significantly. This is because the resources of potential locations in the MEC network are limited, aggregators may be distributed in more locations to meet their capacity requirements, thus significantly increasing the number of aggregators needed.

VII. CONCLUSION

In this paper, we studied the energy minimization problem for FL in an MEC network with uncertain UE availability. We first proposed a novel optimization framework for the problem with a single FL request that consists of two parts: (1) An optimization framework is proposed, and an online learning algorithm for the UE selection built upon the framework then is developed, which considers various contexts (side information) that may have impact on energy consumptions; and (2) an approximation algorithm with an approximation ratio for finding an appropriate number and locations of aggregations in each FL process is proposed. We then devised an online learning algorithm with a bounded regret for the problem with multiple FL requests. We finally evaluated the performance of the proposed algorithms by extensive experiments. Experimental results demonstrate that the performance of the proposed algorithms outperform comparison benchmarks by at least 13% lower energy consumption while achieving the similar accuracy of the global model.

ACKNOWLEDGMENT

We appreciate the three anonymous referees and the associate editor for their constructive comments and suggestions, which helped us improve the quality and presentation of the paper greatly.

REFERENCES

- [1] Amazon Pricing, 2021. Accessed: Apr.–Jul. 2021. [Online]. Available: <https://aws.amazon.com/emr/pricing/>
- [2] M. Abbasi et al., “Efficient resource management and workload allocation in fog–cloud computing paradigm in IoT using learning classifier systems,” *Comput. Commun.*, vol. 153, pp. 217–228, 2020.
- [3] S. Benhamaid, A. Bouabdallah, and H. Lakhlef, “Recent advances in energy management for Green-IoT: An up-to-date and comprehensive survey,” *J. Netw. Comput. Appl.*, vol. 198, 2022, Art. no. 103257.
- [4] S. Bubeck and N. Cesa-Bianchi, “Regret analysis of stochastic and non-stochastic multi-armed bandit problems,” 2012, *arxiv:1204.5721*.
- [5] K. Bonawitz et al., “Towards federated learning at scale: System design,” in *Proc. Mach. Learn. Syst.*, 2019, pp. 374–388.
- [6] S. Caldas et al., “LEAF: A benchmark for federated settings,” 2018, *arxiv:1812.01097*.
- [7] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, “FedHealth: A federated transfer learning framework for wearable healthcare,” *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Jul./Aug. 2020.
- [8] 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,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.
- [9] Y. J. Cho, J. Wang, and G. Joshi, “Client selection in federated learning: Convergence analysis and power-of-choice selection strategies,” 2020, *arxiv:2010.01243*.
- [10] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, and L. Liang, “Self-Balancing federated learning with global imbalanced data in mobile systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 59–71, Jan. 2021.
- [11] A. Dalalyan and A. B. Tsybakov, “Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity,” *Mach. Learn. J.*, vol. 72, no. 1, pp. 39–61, 2008.
- [12] C. T. Dinh et al., “Federated learning over wireless networks: Convergence analysis and resource allocation,” *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.
- [13] ETSI, 2021. Accessed: Jul. 2021. [Online]. Available: <https://www.etsi.org/>
- [14] GT-ITM, 2021. Accessed: Jul. 2021. [Online]. Available: <http://www.cc.gatech.edu/projects/gtitm/>
- [15] C. He, G. Liu, S. Guo, and Y. Yang, “Privacy-preserving and low-latency federated learning in edge computing,” *IEEE Internet of Things J.*, vol. 9, no. 20, pp. 20149–20159, Oct. 2022.
- [16] Z. A. E. Houda, B. Birk, A. Ksentini, L. Khoukhi, and M. Guizani, “When federated learning meets game theory: A cooperative framework to secure IIoT applications on edge computing,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7988–7997, Nov. 2022.
- [17] H. Huang, R. Li, J. Liu, S. Zhou, K. Lin, and Z. Zheng, “ContextFL: Context-aware federated learning by estimating the training and reporting phases of mobile clients,” in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst.*, 2022, pp. 570–580.
- [18] G. Kim and C. Wu, “AutoFL: Enabling heterogeneity-aware energy efficient federated learning,” in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture*, 2021, pp. 183–198.
- [19] J. Kim, D. Kim, J. Lee, and J. Hwang, “A novel joint dataset and computation management scheme for energy-efficient federated learning in mobile edge computing,” *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 898–902, May 2022.
- [20] P. Guo, P. Wang, J. Zhou, S. Jiang, and V. M. Patel, “Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2423–2432.
- [21] M. D. Hossain et al., “Collaborative task offloading for overloaded mobile edge computing in small-cell networks,” in *Proc. Int. Conf. Inf. Netw.*, 2020, pp. 717–722.
- [22] S. Hong and H. Kim, “An integrated GPU power and performance model,” in *Proc. 37th Annu. Int. Symp. Comput. Architecture*, 2010, pp. 280–289.
- [23] C. He et al., “FedML: A research library and benchmark for federated machine learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, MIT Press, 2020, pp. 1–18.
- [24] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, “An efficiency-boosting client selection scheme for federated learning with fairness guarantee,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.
- [25] S. Ilager, K. Ramamohanarao, and R. Buyya, “Thermal prediction for efficient energy management of clouds using machine learning,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 5, pp. 1044–1056, May 2021.
- [26] S. J. Reddi et al., “Adaptive federated optimization,” in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: [OpenReview.net](https://openreview.net)
- [27] S. Jiang et al., “Accelerating mobile applications at the network edge with software-programmable FPGAs,” in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 55–62.
- [28] H. Jeong et al., “IONN: Incremental offloading of neural network computations from mobile devices to edge servers,” in *Proc. ACM Symp. Cloud Comput.*, 2018, pp. 401–411.
- [29] Shannon formula, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Shannon-Hartley_theorem
- [30] X. Kong et al., “Real-time mask identification for COVID-19: An edge computing-based deep learning framework,” *IEEE Internet of Things J.*, vol. 8, no. 21, pp. 15929–15938, Nov. 2021, doi: [10.1109/JIOT.2021.3051844](https://doi.org/10.1109/JIOT.2021.3051844).
- [31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [32] S. Liu et al., “Zeroth-order stochastic variance reduction for nonconvex optimization,” in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, MIT Press, 2018, pp. 3731–3741.
- [33] Y. Liu et al., “A communication efficient vertical federated learning framework,” 2019, *arXiv:1912.11187*.
- [34] Y. Li et al., “Energy-constrained D2D assisted federated learning in edge computing,” in *Proc. 25th Int. ACM Conf. Model. Anal. Simul. Wireless Mobile Syst.*, 2022, pp. 33–37.
- [35] Y. Li et al., “Data collection maximization in IoT-sensor networks via an energy-constrained UAV,” *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 159–174, Jan. 2023.
- [36] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, “To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices,” in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
- [37] L. Li, J. Wang, X. Chen, and C. Xu, “Multi-layer coordination for high-performance energy-efficient federated learning,” in *Proc. IEEE/ACM 28th Int. Symp. Qual. Serv.*, 2020, pp. 1–10.
- [38] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, “Client-edge-cloud hierarchical federated learning,” in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.
- [39] C. X. Mavroumoustakis, G. Kormetzas, G. Mastorakis, A. Bourdena, E. Pallas, and C. D. Dimitriou, “Joint energy and delay-aware scheme for 5G mobile cognitive radio networks,” in *Proc. IEEE Glob. Commun. Conf.*, 2014, pp. 2624–2630.
- [40] H. B. McMahan et al., “Communication-efficient learning of deep networks from decentralized data,” 2016, *arXiv:1602.05629*.
- [41] J. Mills, J. Hu, and G. Min, “Communication-efficient federated learning for wireless edge intelligence in IoT,” *IEEE Internet of Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020.
- [42] Y. Qian, L. Hu, J. Chen, X. Guan, M. M. Hassan, and A. Alelaiwi, “Privacy-aware service placement for mobile edge computing via federated learning,” *Inf. Sci.*, vol. 505, pp. 562–570, 2019.
- [43] P. Rigollet and A. B. Tsybakov, “Sparse estimation by exponential weighting,” *Statist. Sci. J.*, vol. 27, no. 4, pp. 562–570, 2012.
- [44] S. Reddi et al., “Adaptive federated optimization,” 2020, *arxiv:2003.00295*.
- [45] A. Rahmati and L. Zhong, “Context-for-wireless: Context-sensitive energy-efficient wireless data transfer,” in *Proc. 5th Int. Conf. Mobile Syst. Appl. Serv.*, 2007, pp. 165–178.
- [46] V. Smith et al., “Federated multi-task learning,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, MIT Press, 2017, pp. 4427–4437.
- [47] P. Sun et al., “Pain-FL: Personalized privacy-preserving incentive for federated learning,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3805–3820, Dec. 2021.

- [48] S. Shen et al., "Computation offloading with multiple agents in edge-computing-supported IoT," *ACM Trans. Sensor Netw.*, vol. 16, no. 1, pp. 1–27, 2019.
- [49] S. Salamat, B. Khaleghi, M. Imani, and T. Rosing, "Workload-aware opportunistic energy efficiency in multi-FPGA platforms," in *Proc. Int. Conf. Comput.-Aided Des.*, 2019, pp. 1–8.
- [50] S. Shukla et al., "On-device malware detection using performance-aware and robust collaborative learning," in *Proc. IEEE/ACM 58th Des. Automat. Conf.*, 2021, pp. 967–972.
- [51] M. Song et al., "Analyzing user-level privacy attack against federated learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2430–2444, Oct. 2020.
- [52] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.
- [53] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on Non-IID data with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1698–1707.
- [54] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
- [55] L. Wang, Y. Xu, H. Xu, M. Chen, and L. Huang, "Accelerating decentralized federated learning in heterogeneous edge computing," *IEEE Trans. Mobile Comput.*, early access, May 27, 2022, doi: [10.1109/TMC.2022.3178378](https://doi.org/10.1109/TMC.2022.3178378).
- [56] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
- [57] Z. Xu et al., "Energy-aware inference offloading for DNN-Driven applications in mobile edge clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 4, pp. 799–814, Apr. 2021.
- [58] Z. Xu et al., "HierFedML: Aggregator placement and UE assignment for hierarchical federated learning in mobile edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 1, pp. 328–345, Jan. 2023.
- [59] Z. Yang et al., "Delay minimization for federated learning over wireless communication networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1–7.
- [60] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
- [61] Q. Yang et al., "Computation offloading for fast CNN inference in edge computing," in *Proc. Conf. Res. Adaptive Convergent Syst.*, 2019, pp. 101–106.
- [62] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.
- [63] Y. Zhan, P. Li, and S. Guo, "Experience-driven computational resource allocation of federated learning by deep reinforcement learning," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, 2020, pp. 234–243.
- [64] X. Zhu et al., "Empirical studies of institutional federated learning for natural language processing," in *Proc. Findings Assoc. Comput. Linguistics, ACL*, 2020, pp. 625–634.
- [65] W. Zhang et al., "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.
- [66] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the Internet of Vehicles," *Proc. IEEE Proc.*, vol. 108, no. 2, pp. 246–261, Feb. 2020.



Zichuan Xu (Member, IEEE) received the BSc and ME degrees in computer science from the Dalian University of Technology, China, in 2008 and 2011, respectively, and the PhD degree in computer science from the Australian National University, in 2016. From 2016 to 2017, he was a research associate with the Department of Electronic and Electrical Engineering, University College London, U.K. He is currently a full professor and PhD advisor with the School of Software, Dalian University of Technology. He is also a 'Xinghai Scholar' in Dalian University of Technology. His research interests include edge computing, cloud computing, network function virtualization, software-defined networking, algorithmic game theory, and optimization problems.



Dongrui Li received the BE degree from the North China University of Science and Technology, China, in 2020. He is currently working toward the MS degree with the School of Software, Dalian University of Technology, Dalian, China. His current research interests include federated learning, mobile-edge computing, and resource allocation.



Weifa Liang (Senior Member, IEEE) received the BSc degree in computer science from Wuhan University, China, in 1984, the ME degree in computer science from the University of Science and Technology of China, in 1989, and the PhD degree in computer science from the Australian National University, in 1998. He is a professor with the Department of Computer Science, City University of Hong Kong. Prior to that, he was a professor with the Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, mobile edge computing, network function virtualization, Internet of Things, software-defined networking, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He currently serves as an associate editor in the Editorial Board of the *IEEE Transactions on Communications*.

He currently serves as an associate editor in the Editorial Board of the *IEEE Transactions on Communications*.



Wenzheng Xu (Member, IEEE) received the BSc, ME, and PhD degrees in computer science from Sun Yat-Sen University, Guangzhou, P.R. China, in 2008, 2010, and 2015, respectively. He currently is an Associate Professor with Sichuan University. Also, he was a visitor at both the Australian National University and the Chinese University of Hong Kong. His research interests include wireless ad hoc and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.



Qiufen Xia (Member, IEEE) received the BSc and ME degrees in computer science from the Dalian University of Technology, China, in 2009 and 2012, respectively, and the PhD degree in computer science from the Australian National University, in 2017. She is currently an Associate Professor with the Dalian University of Technology. Her research interests include mobile cloud computing, query evaluation, big data analytics, big data management in distributed clouds, and cloud computing.



Pan Zhou (Senior Member, IEEE) received the BS degree from the Advanced Class of HUST, in 2006, the MS degree from the Department of Electronics and Information Engineering, HUST, Wuhan, China, in 2008, and the PhD degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, USA, in 2011. He is currently a full professor and PhD advisor with Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology (HUST), Wuhan, P.R. China. He held an honorary degree in his bachelor's and merit research award of HUST in his master study. He was a senior technical member with Oracle Inc., America, during 2011 to 2013, and worked on Hadoop and distributed storage system for big data analytics at Oracle Cloud Platform. He received the "Rising Star in Science and Technology of HUST" in 2017. He is currently an associate editor of *IEEE Transactions on Network Science and Engineering*. His current research interests include: Security and privacy, Big Data analytics, machine learning, and information networks.

He is currently an associate editor of *IEEE Transactions on Network Science and Engineering*. His current research interests include: Security and privacy, Big Data analytics, machine learning, and information networks.



environments for computational science and commercial computing, data analysis and management for large-scale computing, and scalability in high performance agent systems.

Omer F. Rana (Senior Member, IEEE) received the BS degree in information systems engineering from the Imperial College of Science, Technology and Medicine, London, U.K., the MS degree in microelectronics systems design from the University of Southampton, Southampton, U.K., and the PhD degree in neural computing and parallel architectures from the Imperial College of Science, Technology and Medicine. He is a professor of performance engineering with Cardiff University, Cardiff, U.K. His current research interests include problem solving environments for computational science and commercial computing, data analysis and management for large-scale computing, and scalability in high performance agent systems.



Hao Li is currently an associate researcher with the Institute of Ningxia, China Coal Research Institute. His research interests include Coal mine intellectualization, automation of fully mechanized mining face, edge computing, AI, and Internet of Things.