Age-Aware Data Selection and Aggregator Placement for Timely Federated Continual Learning in Mobile Edge Computing

Zichuan Xu, *Member, IEEE*, Lin Wang, Weifa Liang, *Senior Member, IEEE*, Qiufen Xia, *Member, IEEE*, Wenzheng Xu, *Member, IEEE*, Pan Zhou, *Senior Member, IEEE*, and Omer F. Rana, *Senior Member, IEEE*

***Abstract*—Federated continual learning (FCL) is emerging as a key technology for time-sensitive applications in highly adaptive environments including autonomous driving and industrial digital twin. Each FCL trains machine learning models using newly-generated datasets as soon as possible, to obtain a highly accurate machine learning model for new event predictions. The *age of data*, defined as the time difference between the generation time of a dataset and the current time, is widely adopted as a key criterion to evaluate both timeline and quality of training. In this paper, we study the problem of age-aware FCL in a mobile edge computing (MEC) network. We not only investigate optimization techniques that optimize the data selection and aggregator placement for FCL but also implement a real system as a prototype for age-aware FCL. Specifically, we first propose an approximation algorithm with a provable approximation ratio for the age-aware data selection and aggregator placement problem for FCL with a single request. In real application scenarios, there are usually multiple FCL requests that require to train models, and delays in the MEC network are usually uncertain. We then study the problem of age-aware data selection and aggregator placement problem for FCL with uncertain delays and multiple requests, by devising an online learning algorithm with a bounded regret based on contextual bandits. We finally implement a prototype for FCL in an MEC network, with various heterogeneous user equipments (UEs) and cloudlets with different computing capabilities in the network. Experiment results show that the performance of the proposed algorithms outperform existing studies, by achieving 47% lower age of data and 12% higher model accuracy.**

***Index Terms*—Mobile edge computing, federated continual learning, data selection and aggregator placement, approximation and online algorithms.**

I. Introduction

FEDERATED learning is emerging as the key technique of distributed machine learning to unveil valuable information hidden in various user equipments (UEs), without violating the privacy requirements of users. It enables UEs to collaboratively learn a shared machine learning model while keeping all the training data on UE, decoupling the ability to do machine learning from the need to store the data in a central location. Meanwhile, mobile edge computing (MEC) is envisioned as a fundamental enabler for machine learning in close locations of UEs. In particular, each federated learning admits a set of UEs to train their datasets locally, and its *aggregators* aggregate the trained parameters of each UE to obtain a global machine learning model [2], [40]. Such model aggregation by aggregators can be done timely within the proximity of UEs, by deploying neural network accelerators or GPUs in edge locations of MEC.

A. Motivation

A strong trend of the joint fields of federated learning and MEC is federated continual learning (FCL), which provides real-time guarantee for time-sensitive AI applications [44]. That is, given a newly-changed environment of an MEC network, how to adaptively train an accurate machine learning model and use the model for inference in the new environment. For example, FCL is being deployed for the driver assistance in multiple autonomous vehicles, such as the perception and understanding of the world around the vehicle, which chiefly involves the use of camera- and LiDAR-based systems to detect and classify objects. As the vehicles move to a new location, the current machine learning model needs to be updated online and used for inference in no time. Another example is the digital twin applications in industrial applications, where each digital twin is defined as a real time synchronization of the physical object. The datasets synchronized from a physical object to its

Manuscript received 30 December 2022; revised 6 September 2023; accepted 29 October 2023. Date of publication 21 November 2023; date of current version 15 January 2024. The work of Zichuan Xu and Qiufen Xia was supported in part by the National Natural Science Foundation of China under Grants 62172068, 62172071, and T2350710232, and in part by the LiaoNing Revitalization Talents Program under Project 2022RG04. The work of Weifa Liang was supported by a grant from City University of Hong Kong under Projects 9380137/CS and 7005845. Recommended for acceptance by M. Annavaram. *(Corresponding author: Qiufen Xia.)*

Zichuan Xu, Lin Wang, and Qiufen Xia are with the School of Software and the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology, Dalian 116620, P. R. China (e-mail: z.xu@dlut.edu.cn; l.wang@mail.dlut.edu.cn; qiufenxia@dlut.edu.cn).

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

Wenzheng Xu is with the Department of Computer Network and Communication, Sichuan University, Chengdu, Sichuan 610000, P. R. China (e-mail: wenzheng.xu@scu.edu.cn).

Pan Zhou is with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 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).

Digital Object Identifier 10.1109/TC.2023.3333213

digital twin need to be trained timely and used for inference immediately when a new status of the physical object is present.

In this paper, we aim to enable timely FCL in an MEC network, by timely training the newly-generated datasets when the current model is no longer suitable for the new environment. The *age of data* (AoD) is emerging as a key criterion to measure the timeliness of such AI applications, with the aim of guaranteeing the most fresh datasets are trained timely. Specifically, the age of data is defined as the difference between the current time and the generation time of the dataset. The age of data is particularly important in FCL. In particular, given a newly-changed environment of an MEC network, the newly-generated datasets are valuable to enable the fast converge of machine learning model in the previous environment. Therefore, in this paper, we aim to enable efficient age-aware FCL in an MEC network.

B. Challenges and Novelty

Enabling age-aware FCL in the MEC network poses significant challenges. First, there are multiple user requests with each requiring to train a machine learning model. Implementing each FCL request in the MEC network thus involves dataset selection and the placement of aggregators. However, the MEC network is fully dynamic and heterogeneous with UEs, base stations, and cloudlets having various capacities of computing resource. Therefore, how to select datasets of which UEs, and how to place the aggregators of FCL requests in the MEC network are challenging. Second, minimizing the total AoD in the FCL request requires complex tuning of various delays, such as local training delays, trained parameter transmission delays, and parameter aggregation delays. For example, a UE may not have enough computing resource to train its most fresh yet large datasets. Further, the datasets generated may be postponed for training if the transmission path from UEs to aggregators of FCL is congested. Third, the delays of local training, transmission, and parameter aggregation are usually uncertain in dynamic MEC environments. Particularly, UEs usually have their main functionalities and local training may be done opportunistically when they have idle computing resources. Since the resource availabilities of UEs are uncertain, the local training delays are uncertain as well. Besides, the resources of the MEC network are fully dynamic, since there are multiple FCL requests under implementation in it. Different FCL requests can have influence on each other, thereby leading to uncertain network delays.

To the best of our knowledge, we are the first to investigate the age-aware data selection and aggregator placement problems for FCL in an MEC network, by designing approximation and online learning algorithms for them. Although there are studies on federated learning in MEC networks, most of them ignored the AoD and uncertain delays and did not consider the implementation of multiple FCL requests [20], [32], [33], [39], [40], [42].

The main contributions of this paper are listed as follows.

- We formulate the age-aware data selection and aggregator placement problems for FCL in an MEC network, with the aim to minimize the total age of the analyzed datasets required by FCL requests.
- For the problem with a single FCL request, we devise an exact solution and an approximation algorithm with a provable approximation ratio.
- For the problem with multiple FCL requests and uncertain delays, we propose an online learning algorithm with a bounded regret, by adopting a novel contextual bandit learning technique.
- To evaluate the performance of the proposed algorithms, we implement a real test bed with a number of real edge devices, switches, and physical servers. Experimental results indicate that the proposed algorithms outperform existing ones.

The rest of the paper is arranged as follows. Section II summarizes the state-of-the-arts on this topic. Section III introduces the system mode and problem formulation. Section IV proposes exact and approximate solutions to the age-aware data selection and aggregator placement problem for FCL with a single request. Section V describes the online learning algorithm for the problem with multiple FCL requests and uncertain delays. Section VI implements a real testbed for FCL in MEC networks and provides some experimental results on the performance of the proposed algorithms in the testbed. Section VII concludes this paper and gives the future work related to the paper.

II. Related Work

Studies on improving the efficiency of federated learning in MEC networks usually focus on improving the convergence speed of federated learning by motivating edge users participating the learning, and optimizing the delays of each round of federated learning in the MEC network. To the best of our knowledge, we are not aware of studies that focus on data selection and aggregator placement for continual federated learning in an MEC network, with the aim of minimizing the AoD. Specific differences are summarized in the following subsections.

A. Convergence Optimization

Improving the convergence speed of federated learning is a major research direction in optimizing the efficiency of federated learning in MEC networks [8], [9], [14], [17], [21], [22], [29], [30], [36], [38]. Most of them are not suitable for time-sensitive applications in MEC networks as they do not optimize the delays of training and transmitting data in each federated learning process. For example, Duan et al. [8] designed a self-balancing federated learning framework to solve the accuracy degradation caused by imbalanced distributed training datasets in the federated learning process. He et al. [9] designed automating federated learning to improve model accuracy and reduce the manual design effort. Mills et al. [21] aimed to reduce the convergence speed of FedAvg by reducing convergence time via compressing the trained parameters that needed to be transmitted. Prakash et al. [29] mitigated stragglers by injecting structured coding redundancy into federated learning. Ren et al. [30] designed a scheduling policy for unbiased gradient aggregation in federated learning. Wu et al. [38] proposed a multi-layer federated learning protocol to adopt two-level (the edge and cloud levels) aggregation of trained parameters with different aggregation strategies. Wang et al. [36] proposed a control resource algorithm that determines the best trade-off between local update and global parameter aggregation to minimize the loss function under the resource-limited MEC network.

B. Efficient Incentive Mechanisms

Another direction of optimizing the accuracy of FCL in MEC networks is to devise incentive strategies to motivate UEs with high quality training parameters to contribute in the federated learning [3], [6], [15], [24], [25], [28], [35], [45]. Most of such studies however did not consider the impact of data selection and aggregator placement on the accuracy of FCL.

For example, Ding et al. [6] proposed a mechanism to motivate UEs to participate in federated learning, by considering the training cost and communication delay of UEs. Mohammed et al. [24] investigated the problem of promoting the accuracy of federated learning by selecting the best candidate clients. Nguyen et al. [25] proposed a fast-convergent algorithm that performs intelligent sampling of UEs in each round of model training. Pandey et al. [28] proposed a novel crowdsourcing framework to establish an incentive mechanism to motivate users participating in the federated learning, so that the communication efficiency during parameters exchanges is improved. Wang et al. [35] selected UEs to counterbalance the bias introduced by non-IID data.

C. Delay Optimization

Although the communication delay is a vital part of AoD, existing studies on optimizing communication delay for federated learning cannot be directly applied to FCL in MEC [20], [32], [33], [39], [41], [42]. The reason is that most of them largely ignored the age of local datasets and complex interplays between different delays.

For instance, Luo et al. [20] proposed a novel hierarchical federated edge learning framework by focusing on the joint allocation of computation and communication resources with an aim to minimize global cost. Shi et al. [32] devised algorithms for task scheduling and resource allocation to maximize the model accuracy within a given total training time budget for delay-constrained federated learning. Tran et al. [33] explored the trade-offs between computation and communication latencies to accelerate the federated learning in MEC networks. Xia et al. [39] designed a multi-armed bandit-based framework for online client scheduling to minimize the training latency in federated learning without knowing wireless channel state information and statistical characteristics of clients. Yang et al. [42] studied the problem of scheduling requests in a wireless cellular environment, by considering the inter-cell interference in the network.

D. Age Optimization

Although Age of Information (AoI) is a critical metric in evaluating real-time communication systems, current researches on AoI may not be applied to optimize FCL in MEC directly [1], [10], [19], [34]. This is primarily because these works often overlook the age of data that is fundamentally different with AoI. Specifically, the AoI mainly considers the freshness of data packets. The primary goal of studies on age of information is to minimize the time difference between the generation and consumption of a data packet. In our study, simply considering AoI may not work for federated continual learning, because the performance of learning depends on which datasets are used for training instead of which packets. Specifically, datasets may consist of many data packets, and the optimization of age of packets of a dataset may not lead to a minimum age for the dataset.

For example, Asheralieva and Niyato [1] investigated the problem of payoff maximization of Internet of Everything applications in an MEC network, by jointly optimizing the AoI, security, and resilience to maximize the expected long-term system payoff. Liu et al. [19] considered the promoting the timeliness of tasks and data delivery in an MEC network with both sensors and edge servers, by considering the age that consists of energy harvesting duration, waiting and service delays in edge servers. He et al. [10] investigated the problem of AoI minimization for time sensitive applications such as real-time monitoring in an MEC environment, by proposing a Reinforcement Learning algorithm for the problem. Wang et al. [34] designed a federated multi-agent reinforcement learning algorithm to minimize the average AoI while maximizing the number of tasks that can be executed by edge nodes, subject to resource capacities in a dynamically-changing MEC network.

III. Preliminary

In this section, we first introduce the system model and notations. We then define the problems precisely.

A. System Model

We consider an MEC network $G = (\mathcal{BS} \cup \mathcal{CL}, E)$ consisting of a set $\mathcal{BS}$ of base stations and a set $\mathcal{CL}$ of cloudlets, which are deployed in locations within the proximity of UEs. As shown in Fig. 1, a set of base stations provide wireless connections with nearby UEs and computing resources to accelerate AI applications by attaching neural network accelerators, such as Intel Neural Stick 2 [13]. Let bs_i be a base station in $\mathcal{BS}$. Cloudlets in $\mathcal{CL}$ are usually located in the backhaul of the MEC. Each cloudlet, denoted by cl_j, contains several servers with computing resources that can be leveraged by FCL to perform parameter aggregation.

Each UE in the MEC network continuously generates datasets over time. To timely train such datasets while guaranteeing the privacy requirements of UEs, an FCL request is issued to train the datasets of each UE locally and the locally trained parameters are sent to the MEC network for aggregation via nearby base stations. Let ue_k be a UE with $1 \le k \le K$. Considering that the trained parameters of UEs can be aggregated in

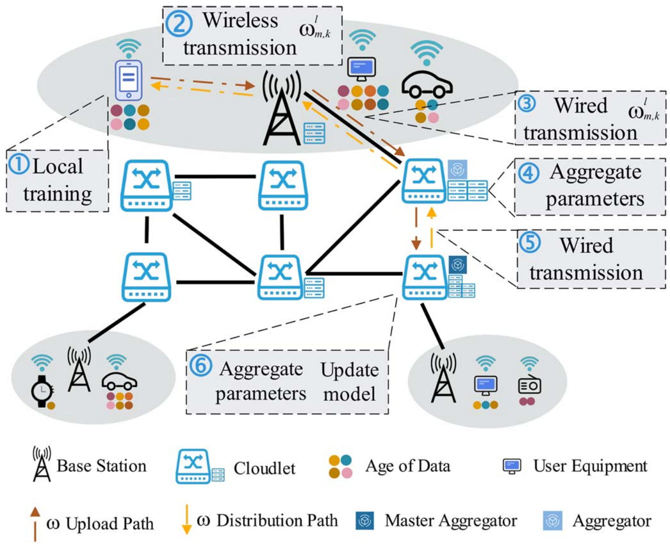


Fig. 1. An example of the MEC network. The federated learning framework involves the coordination among UEs, aggregators, and master aggregators. Specifically, each UE first performs local training (step 1), and then transmits the trained parameters to its aggregators or master aggregators via wireless and wired transmissions (step 2,3 and 5). On receiving the trained parameters from UEs, the aggregators and master aggregators aggregate the parameters to obtain a global model (step 4 and 6).

either a base station or a cloudlet, we consider a cloudlet cl_j or a base station bs_i as a potential location Loc_q ($\in \{\mathcal{BS} \cup \mathcal{CL}\}$) for parameter aggregation. The aggregation of trained parameters can be performed in containers of potential locations. However, the computing and bandwidth resources of each potential location are capacitated, and denoted by C_q and B_q the computing and bandwidth capacities of Loc_q, respectively. An example of the MEC network for FCL is shown in Fig. 1.

B. Age of Data

Each UE ue_k has a collection of datasets that are generated in different time slots, and let $\mathcal{DS}_k$ be such a collection of UE ue_k. Each dataset $ds_{k,t} \in \mathcal{DS}_k$ is generated at time slot t. Considering that each UE usually has limited storage capacity, not all historical datasets can be stored. We thus assume that each UE ue_k maintains a fixed volume of data generated in the past. However, the size of a dataset of each UE may change at different times. For example, given a UE that monitors temperature and humidity in a greenhouse, its generated dataset may be much larger and more variable during daytime. On the other hand, the dataset could be smaller and less variable, during nighttime. This means that both the number and age of datasets that each UE maintains may be different in different time slots.

The staleness of a dataset has a significant negative impact on the accuracy of its machine learning model [18], [31], [41]. For example, in a camera network for intelligent surveillance, the dataset of a new moving object (a vehicle or a person) captured by a camera needs to be trained timely. Otherwise, the object may not be recognized correctly in other cameras. The reason behind this is that the newly generated dataset is typically more relevant to the event prediction in the future. Thus, the generated datasets of each UE need to be trained as soon as possible, such that the most timely training parameters are aggregated to obtain a global machine learning model.

We thus consider the *age of data* (AoD) as a fundamental criterion to indicate the timeliness of training. The AoD is defined as the elapsed time between the current time slot τ and the time when the dataset is generated, then the age of data $ds_{k,t}$ is

$$\tau - t_{k,t}, \tag{1}$$

where $t_{k,t}$ is the time point when $ds_{k,t}$ is generated.

C. Federated Continual Learning in MEC

An FCL request to train a machine learning model based on the datasets of UEs. Implementing an FCL request requires multiple rounds of local training on datasets distributed in UEs and aggregation in locations of the MEC network. Following the hierarchy aggregation framework in [2], [40], we assume that a service coordinates the aggregations of each FCL request. Let S_m and r_m be the service and an FCL request. As shown in Fig. 1, in each round of the learning, each participating UE ue_k trains its data locally to obtain a local model for FCL request r_m. Also, service S_m uses a set of *aggregators* that are located in different locations to collect their local models from UEs. Without loss of generality, we assume that one of the major functionalities of each service S_m is to aggregate the parameters received from its aggregators, serving as a *master aggregator*. Therefore, service S_m finally aggregates the parameters from all aggregators to obtain the global model of the current round. Service S_m assigns the current global model to participating UEs for the training of the next round. Note that in each round of training, FCL has the same parallelism as that of federated learning. Specifically, each UE performs computations (local training) in parallel. After the training of their local models, UEs send their trained model to aggregators for aggregation in parallel. The FCL also considers data parallelism by continually training data that is generated dynamically. In particular, datasets with different generation times in each UE are fully utilized to train the fresh data by jointly considering the computing and communication capabilities of UEs.

Each UE ue_k consumes computing resource to train its dataset $ds_{k,t}$, which is proportional to the volume of $ds_{k,t}$. Thus, the amount of computing resource required by UE ue_k to train dataset $ds_{k,t}$ in each round is

$$\alpha_k \cdot |ds_{k,t}|, \tag{2}$$

where α_k is the amount of computing resource that is used to process a unit amount of data. Let C_k be the capacity of computing resource that each UE ue_k uses to train a unit amount of data, then,

$$\alpha_k \cdot |ds_{k,t}| \leq C_k. \tag{3}$$

To fully utilize the heterogeneous UE capabilities, different neural networks with different width of hidden channels are used to perform local training in UEs [5]. As such, the sizes of the obtained trained parameters of UEs are different. Let $w^l_{m,k}$ be the trained parameters by ue_k with the lth hidden channel size, where $l = 1$ represents that the local model has the same size as the global model and $1 \leq l \leq L$, and $|w_{m,k}|$ the volume of the trained parameters.

Aggregators are deployed close to the UEs such that the transmission cost of trained parameters is minimized. Let $A_{m,o}$ denote the oth aggregator of service S_m. Note that each FCL request r_m may not need to require the participation of all UEs to obtain a global model. Besides, a UE may not be able to transmit its trained parameters if its network delay is large at the current time. Therefore, we let $\mathcal{UE}_{m,o}$ be the set of UEs whose trained parameters will be aggregated by the aggregator $A_{m,o}$ of FCL request r_m. The computing resource consumed by aggregator $A_{m,o}$ due to aggregation of the trained parameters from the UEs $\mathcal{UE}_{m,o}$ is

$$\sum\nolimits_{ue \in \mathcal{UE}_{m,o}} |w^l_{m,k}| \cdot \beta_q, \tag{4}$$

where β_q is the amount of computing resource assigned to aggregate a unit amount of data in location $Loc_q \in \mathcal{CL} \cup \mathcal{BS}$.

D. Delay Model

The delay incurred during each round of an FCL is due to local training in UEs, transmitting and aggregating trained parameters. Let γ_k be the time that ue_k takes in training a unit amount of data. If $ds_{k,t}$ is selected to train the model in ue_k, the time spent in training $ds_{k,t}$ is

$$|ds_{k,t}| \cdot \gamma_k. \tag{5}$$

The trained parameters of each UE ue_k are transmitted to its registered base station first, and then from the base station to its assigned aggregator that is located in a location of the MEC network. As such, both wireless and wired transmission delays are incurred. First, let $d^{wls}_{m,k}$ be the wireless transmission delay of transmitting trained parameters $w^l_{m,k}$ from ue_k to its registered base station, then,

$$d^{wls}_{m,k} = \frac{|w^l_{m,k}|}{r^{wls}_k}, \tag{6}$$

where r^{wls}_k is the achievable wireless transmission rate of link between ue_k and its base station that can be obtained following studies in [4], [11]. Second, assuming that UE ue_k is registered to an aggregator $A_{m,o}$ that is placed to location Loc_q, the delay of updating its trained parameters from its base station to $A_{m,o}$ of service S_m is $\delta_{k,q} \cdot |w^l_{m,k}|$, where $\delta_{k,q}$ is the delay of transferring a unit amount of data from the base station of ue_k to location Loc_q.

Model aggregation by $A_{m,o}$ in Loc_q incurs a delay of

$$\xi_q \cdot |w^l_{m,k}| + \xi_m \cdot |w_m|, \tag{7}$$

where ξ_q and ξ_m are the delays of aggregating a unit amount of data in location Loc_q and service S_m, and w_m denotes the aggregated parameters.

After aggregation, each aggregator sends its aggregated parameters to service S_m for final aggregation. If $A_{m,o}$ is placed to Loc_q, the delay of updating w_m from $A_{m,o}$ to service S_m can be calculated by

$$\delta_{q,m} \cdot |w_m|, \tag{8}$$

where $\delta_{q,m}$ is the delay of transferring a unit amount of data from location Loc_q to the location of service S_m.

Each round of FCL has a fixed length of time [2], [43]. Let T_r be the time length of each round. The total delay experienced by each UE in a round has to be smaller than T_r; otherwise, its trained parameters will not be aggregated.

E. Problem Formulation

Given an MEC network $G = (\mathcal{BS} \cup \mathcal{CL}, E)$ with a set $\mathcal{BS}$ of base stations, a set $\mathcal{CL}$ of cloudlets, and a set $\mathcal{UE}$ of UEs, each UE continuously generate datasets for training in a finite time horizon. There are multiple FCL requests that are issued to train the generated datasets to obtain global model. It is very important to select the proper datasets of each UE for each FCL request. Our metric is to find the datasets with the minimum age. The rationale behind is that the fresh dataset may have valuable information that contribute a high quality global model. For instance, in a real-time traffic monitoring scenario, utilizing the most recent traffic data for model training is crucial, because the most recent data indicates the real-time road conditions and traffic flows and enables the model to generate more accurate and timely predictions on upcoming traffic patterns or potential congestions. Specifically, we focus on improving the efficiency of the learning process of each FCL request, by adopting the age of data as a key metric to guide the implementation of each FCL request. To this end, we formulate the following optimization problems.

Problem 1: the *age-aware data selection and aggregator placement problem for FCL in an MEC network with a single request* is to jointly place a number of aggregators for FCL request r_m, assign each UE to a placed aggregator, and select a dataset of each UE for local training, with an aim to minimize the age of all trained datasets, subject to the resource capacities on base stations, cloudlets and the delay requirement of training the selected dataset of each UE.

Problem 2: given a set R of FCL requests and uncertain delays, the *age-aware data selection and aggregator placement problem for FCL in an MEC network with multiple requests and delay uncertainty* is to implement a subset of requests in R, by jointly placing a number of aggregators of each FCL request r_m, assigning each UE to a placed aggregator, and select a dataset of each UE for local training, with an aim to minimize the age of all trained datasets of FCL requests, subject to the resource capacities on base stations and cloudlets and the delay requirement of training the selected dataset of each UE.

The relationship between **Problem 1** and **Problem 2** is in the following: (1) **Problem 1** deals with a single FCL request with a focus on selecting the right datasets, while **Problem 2** deals with multiple FCL requests, and both problems are complement with each other, (2) **Problem 2** considers a more complicated scenario based on **Problem 1** by incorporating the delay uncertainty. As such, the solution to **Problem 1** forms a base for that of **Problem 2**.

Note that the optimization problems focus on optimizing the efficiency of FCL in an MEC network, given the limited resources in the MEC network. On the other hand, promoting

TABLE I
SYMBOLS

Symbols	Meaning
$G=(\mathcal{BS}\cup\mathcal{CL},E)$	an MEC network with a set $\mathcal{BS}$ of base stations, a set $\mathcal{CL}$ of cloudlets and a set $\mathcal{E}$ of links.
bs_i, cl_j	a base station in BS and a cloudlet in $\mathcal{CL}$.
ue_k, K	a UE and the number of UEs, where $1\le k\le K$.
Loc_q $(\in\{\mathcal{BS}\cup\mathcal{CL}\})$	a potential location to aggregate the trained parameters of UEs.
C_q, B_q	the capacities of computing and bandwidth resources of each potential location Loc_q.
$\mathcal{DS}_k$, $ds_{k,t}$	the collection of datasets generated by UE ue_k and the dataset generated by ue_k in time slot t.
$\|ds_{k,t}\|$, $t_{k,t}$	the volume of dataset $ds_{k,t}$, and the generation time slot of dataset $ds_{k,t}$.
r_m, R, S_m	an FCL request, a set R of FCL requests, and the service that coordinates the parameter aggregations of each request.
α_k	the amount of computing resource that is used to process a unit amount of data by each UE ue_k.
C_k	the capacity of computing resource that each UE ue_k uses to train a unit amount of data.
$w^l_{m,k}$, $w^l_{m,k}$	the trained parameters by ue_k with the lth hidden channel size, where $l=1$ represents that the local model has the same size as the global model and $1\le l\le L$, and its volume (size).
$A_{m,o}$	the task of the oth aggregator of service S_m.
$\mathcal{UE}_{m,o}$	the set of UEs whose trained parameters will be aggregated by the aggregator $A_{m,o}$ of FCL request r_m.
β_q	the amount of computing resource assigned to aggregate a unit amount of data in location $Loc_q\in\mathcal{CL}\cup\mathcal{BS}$.
γ_k	the time that ue_k takes in training a unit amount of data.
$d^{wls}_{m,k}$	the wireless transmission delay of transmitting trained parameters $w^l_{m,k}$ from ue_k to its registered base station.
r^{wls}_k	the achievable wireless transmission rate of link between ue_k and its base station.
$\delta_{k,q}$	the delay of transferring a unit amount of data from the base station of ue_k to location Loc_q.
ξ_q, ξ_m	the delays of aggregating a unit amount of data in location Loc_q and service S_m, respectively.
w_m, $\|w_m\|$	the aggregated parameters by service S_m, and its size.
T_r	the time length of each round of an FCL.
$x_{k,t,q}$	an indicator variable to show whether dataset $ds_{k,t}$ is selected for local training in ue_k and assigned to location Loc_q to aggregate the trained parameters.
$y_{m,q}$	a binary indicator variable that shows whether there is an aggregator in location Loc_q for FCL request r_m.
x^*, y^*	the optimal fractional solution to the **LP**.
$x'_{k,t,q}$	the filtered value of $x^*_{k,t,q}$.
DS_{max}, σ	$\max_k\{\|\mathcal{DS}_k\|\}$ and a positive value with $\sigma>1$.
η	$1/(\sigma\cdot DS_{max})$.
$\mathcal{X}$, $\mathcal{X}'$	the sets of optimal fractional values (i.e., $x^*_{k,t,q}$) that are smaller and no smaller than η.
$Y_{k,t}$, $Y_{k,t,q}$	the event that represents the amount of dataset that is selected for local training in ue_k, and the amount of data whose trained parameters are sent to location Loc_q for aggregation.
$Pr[\cdot]$, ϵ	the probability of an event, and a constant with $\epsilon>0$.
θ, Θ, $\|\Theta\|$	a context available in the system, the set of contexts, and the number of contexts.
Δ and Ξ	a vector of the data updating delays ($\delta_{k,q}$ and $\delta_{q,m}$) and a vector of the data aggregation delays (ξ_q and ξ_m).
$p_{\theta,m}$	the recommendation probability of FCL request r_m under context $\theta\in\Theta$.
$w_{\tau,\theta,m}$	a weight of each context θ on FCL request r_m at each time slot τ.
p_m	the probability of selecting FCL request r_m to adjust dataset selection and aggregator placement.
$\phi_{\tau,\theta,m}$	the penalty of context θ is the total age of FCL request r_m if r_m is selected by context θ.
ε	a constant in the range of $[0,1)$.

the accuracy of FCL is out of the scope of this paper, since the accuracy mainly depends on specific neural network models used to implement the FCL requests. Note that the proposed algorithms in this paper are general enough to be applied to FCL for various neural networks.

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

IV. Solutions to the Age-Aware Data Selection and Aggregator Placement Problem for FCL in an MEC Network With a Single Request

We study the age-aware data selection and aggregator placement problem for FCL in an MEC network with a single request, by proposing an exact solution and an approximation algorithm.

A. An Exact Solution

Recall that the dataset $ds_{k,t}\in\mathcal{DS}_k$ is generated at time slot t by UE ue_k and $\tau-t_{k,t}$ is the age of it. To obtain high quality training, for current time slot τ, we may use the most fresh dataset for local training in ue_k, to obtain the minimum age. However, the current available resource in ue_k may not be enough. Therefore, we need to strategically select datasets for local training in each ue_k and place aggregators of each FCL request, to efficiently obtain high quality training. Let $x_{k,t,q}$ be an indicator variable to show whether dataset $ds_{k,t}$ is selected for local training in ue_k and assigned to location Loc_q to aggregate the trained parameters, where $\max\{1,\tau-|\mathcal{DS}_k|\}\le t\le\tau$. Let $y_{m,q}$ be a binary indicator variable that shows whether there is an aggregator in location Loc_q for FCL request r_m. The objective is to minimize the total age of locally trained datasets in time slot τ, i.e.,

$$\textbf{ILP:}\ \min\sum_{ue_k\in UE;ds_{k,t}\in DS_k}\ \sum_{Loc_q\in\{\mathcal{BS}\cup\mathcal{CL}\}} x_{k,t,q}(\tau-t_{k,t}),\tag{9}$$

subject to

$$\sum_{q=1}^{|\mathcal{BS}\cup\mathcal{CL}|}\ \sum_{t=1}^{\tau-|DS_k|} x_{k,t,q}=1\quad\forall\, ue_k\in UE\tag{10}$$

$$\sum_{Loc_q \in BS \cup CL} y_{m,q} = 1 \tag{11}$$

$$x_{k,t,q} \le y_{m,q} \qquad \forall Loc_q \in \mathcal{CL} \cup \mathcal{BS} \tag{12}$$

$$\sum_{q=1}^{|\mathcal{BS} \cup \mathcal{CL}|} \sum_{t=1}^{\tau - |DS_k|} x_{k,t,q} \cdot \alpha_k \cdot |ds_{k,t}| \le C_k, \forall ue_k \in UE \tag{13}$$

$$\sum_{ue_k \in UE} \sum_{t=1}^{\tau - |DS_k|} x_{k,t,q} \cdot |w^l_{m,k}| \cdot \beta_q \le C_q, \ \forall Loc_q \tag{14}$$

$$\begin{aligned} &\sum_{q=1}^{|\mathcal{BS} \cup \mathcal{CL}|} \sum_{t=1}^{\tau - |DS_k|} \left(x_{k,t,q} \cdot |ds_{k,t}| \cdot \gamma_k \right) + d^{wls}_{m,k} \\ &\quad + \sum_{q=1}^{|\mathcal{BS} \cup \mathcal{CL}|} \Big(x_{k,t,q} (\delta_{k,q} |w^l_{m,k}| + \delta_{q,m} |w_m| \\ &\quad + \xi_q |w^l_{m,k}|) \Big) + \xi_m |w_m| \le T_r, \forall \ ue_k \in UE \end{aligned} \tag{15}$$

$$x_{k,t,q}, y_{m,q} \in \{0, 1\} \tag{16}$$

Constraint (10) guarantees that each UE ue_k chooses one of its datasets in $\mathcal{DS}_k$ and the trained parameters can only be sent to a single location for aggregation. Constraint (11) ensures that there is maximally a single aggregator in location Loc_q for request r_m. Constraint (12) guarantees that if a UE is assigned to Loc_q for aggregation, an aggregator has to be placed into Loc_q. Constraints (13) and (14) indicate that the computing capacities of ue_k and an aggregator $A_{m,o}$ cannot be exceeded by the total resource demand of the requests assigned to them, respectively. Constraint (15) indicates that each request r_m should be processed during the given time T_r, such that its local model can be updated for aggregation timely. Meanwhile, Constraint (16) imposes an integral constraint on the variables $x_{k,t,q}$ and $y_{m,q}$.

B. An Approximation Algorithm

Since the exact solution may not be scalable for large problem size, we here instead propose an efficient approximation algorithm by adopting a novel randomized rounding technique that carefully controls the quality of the obtained solution. To this end, we relax the **ILP** into an Linear Program (**LP**), and obtain a fractional solution to the **LP**. We carefully adjust the rounding procedure to minimize the gap of the rounded solution (integral solution) from the optimal one.

We first relax constraint (16) of **ILP** to

$$0 \le x_{k,t,q}, y_{m,q} \le 1. \tag{17}$$

Let **LP** be the obtained relaxed version of **ILP**. We can then obtain a fractional solution to the **LP** in polynomial time. Let x^* and y^* be the optimal fractional solution to the **LP**. The fractional solution means that each aggregator may be 'split' into multiple locations and a dataset may be partially selected and assigned to a location. This however is not feasible to the age-aware data selection and aggregator placement problem for FCL in an MEC network with a single request. We thus round the fractional solution to an integral solution.

Algorithm 1 `ApproS`

Input: An MEC network $G = (\mathcal{BS} \cup \mathcal{CL}, E)$, a number of UEs, and an FCL request r_m.
Output: The age of implementing FCL request r_m.

Relax Constraint (16) of ILP into Constraint (17) and obtain an LP;
Obtain a fraction solution x^* and y^* by solving the LP;
Filter x^* to x' according to Eq. (18) and Eq. (20);
for each UE **do**
  Select the dataset $ds_{k,t}$ and assign its trained parameters to location Loc_q with probability $x'_{k,t,q}/\sigma$;
  if $ds_{k,t}$ is selected and its trained parameters are sent to Loc_q **then**
    Place an aggregator in Loc_q and let $y_{m,q} = 1$;

To obtain a high quality integral solution in the randomized rounding, we filter out some small fraction values in the solution. The rationale behind is that a small fraction solution means a small probability to select and assign a dataset of each UE, thereby having the least contribution to minimize the age of trained datasets. As a result, we filter out such fractional solutions and amortize the filtered probabilities to the rest fractional solutions. Specifically, we first filter out the value of each $x^*_{k,t,q}$. Let $x'_{k,t,q}$ be the filtered value of $x^*_{k,t,q}$, we then have

$$x'_{k,t,q} = 0, \text{if } x^*_{k,t,q} < \eta; \tag{18}$$

otherwise, we amortize the filtered values of each $x^*_{k,t,q}$ to the variables that are no smaller than η. Specifically,

$$\eta = 1/(\sigma \cdot DS_{max}), \tag{19}$$

where $DS_{max} = \max_k\{|\mathcal{DS}_k|\}$ and σ is a positive value with $\sigma > 1$. For clarity, we use $\mathcal{X}$ and $\mathcal{X}'$ to denote the set of optimal fractional values (i.e., $x^*_{k,t,q}$) that are smaller and no smaller than η, respectively. If $x^*_{k,t,q} \in \mathcal{X}'$, we set it to

$$x'_{k,t,q} = \frac{x^*_{k,t,q} \cdot \sum_{x^*_{k',t',q'} \in \mathcal{X}} x^*_{k',t',q'}}{\sum_{x^*_{k',t',q'} \in \mathcal{X}'} x^*_{k',t',q'}} + x^*_{k,t,q}. \tag{20}$$

Given the filtered fractional solution x', we have each fractional value $x'_{k,t,q}$ greater than η and the filtered value of all $x^*_{k,t,q}$ (smaller than η) is amortized to x'. We then select the dataset $ds_{k,t}$ of UE ue_k and assign its trained parameters to location Loc_q with probability $x'_{k,t,q}/\sigma$. If $ds_{k,t}$ is selected and its trained parameters are sent to Loc_q, we place an aggregator in Loc_q, i.e. $y_{m,q} = 1$. The detailed steps of the algorithm are described in **Algorithm** 1, which is referred to as `ApproS`.

C. Algorithm Analysis

We first show that the obtained fractional solution x' by `ApproS` can be used to obtain a feasible randomized solution.

Lemma 1: Showing that x' is a feasible probability is to show that each variable $x'_{k,t,q} \le 1$.

Proof: According to the filtering and rounding procedure of **Algorithm** `ApproS`, we consider the following cases: (1) $x^*_{k,t,q} < \eta$, i.e., $x^*_{k,t,q} \in \mathcal{X}$; and (2) $x^*_{k,t,q} \ge \eta$, i.e., $x^*_{k,t,q} \in \mathcal{X}'$. For each variable $x^*_{k,t,q} < \eta$, we have $x'_{k,t,q} = 0$. In the following, we thus prove case (2). The sum of all the optimal fractional values of each ue_k and its dataset $ds_{k,t}$ is 1, since only one dataset of a UE can be selected and the trained parameters can be assigned to a single location for aggregation. That is, if the optimal fractional values of all UEs are considered, we have

$$\sum_{x^*_{k,t,q}\in\mathcal{X}} x^*_{k,t,q} + \sum_{x^*_{k,t,q}\in\mathcal{X}'} x^*_{k,t,q} = |UE|. \tag{21}$$

Let DS_{max} be the maximum number of datasets of UE ue_k. According to Eq. (20) and $\eta = \frac{1}{\sigma\cdot DS_{max}}$, where σ is a positive value $\sigma > 1$, we have

$$\begin{aligned} x'_{k,t,q} &= \frac{x^*_{k,t,q}\cdot(|UE| - \sum_{x^*_{k',t',q'}\in\mathcal{X}'} x^*_{k',t',q'})}{\sum_{x^*_{k',t',q'}\in\mathcal{X}'} x^*_{k',t',q'}} + x^*_{k,t,q} \\ &\le \frac{x^*_{k,t,q}\cdot(|UE| - DS_{max}\cdot|UE|\cdot\eta)}{DS_{max}\cdot|UE|\cdot\eta} + x^*_{k,t,q} \\ &\le x^*_{k,t,q}\cdot(\sigma-1) + x^*_{k,t,q} = \sigma\cdot x^*_{k,t,q}. \end{aligned} \tag{22}$$

Since we scale down each variable in $\mathcal{X}'$ by a factor of σ, $x'_{k,t,q}\le 1$. □

We then show that the computing resource capacities are violated with small probabilities.

Lemma 2: Assuming that $C_k \ge 6\sigma\ln|UE|$, the computing resource capacity of each UE ue_k is violated with probability of $1/|UE|^2$, where σ is a small positive constant with $\sigma > 1$.

Proof: In the current time slot τ, a single dataset of each UE ue_k is selected for local training. The computing resource allocated to processing the selected dataset should not exceed its computing capacity C_k. Let $Y_{k,t}$ be the event that represents the amount of dataset that is selected for local training in ue_k, and $Y_{k,t,q}$ be the amount of data whose trained parameters are sent to location Loc_q for aggregation. We have $Y_{k,t} = \sum_{q=1}^{|\mathcal{CL}\cup\mathcal{BS}|} Y_{k,t,q}$. Dataset $ds_{k,t}$ is selected and assigned to location Loc_q with probability $x'_{k,t,q}$ with $\frac{1}{\sigma^2 DS_{max}} \le x'_{k,t,q} \le 1$. We thus have $Y_{k,t,q} = |ds_{k,t}|$ with probability $\frac{x'_{k,t,q}}{\sigma}$ and zero with probability $1-\frac{x'_{k,t,q}}{\sigma}$. Then, we can calculate the expected amount of dataset that is selected for local training in ue_k by

$$\begin{aligned} E(Y_{k,t}) &= E\left(\sum_{q=1}^{|\mathcal{CL}\cup\mathcal{BS}|} Y_{k,t,q}\right) \\ &= \sum_{q=1}^{|\mathcal{CL}\cup\mathcal{BS}|} E(Y_{k,t,q}) = \sum_{q=1}^{|\mathcal{CL}\cup\mathcal{BS}|} \frac{x'_{k,t,q}}{\sigma}\cdot|ds_{k,t}| \quad (23) \\ &= \sum_{q=1}^{|\mathcal{CL}\cup\mathcal{BS}|} \frac{x^*_{k,t,q}}{\sigma}\cdot|ds_{k,t}| \quad (24) \\ &\le C_k/(\alpha_k\sigma), \text{due to Constraint (13).} \quad (25) \end{aligned}$$

Note that the derivation from (23) to (24) is due to the fact that the filtered values of x^* are also amortized to other frac tional solutions.

Denote by $Pr[\cdot]$ the probability of an event. The capacity of each UE ue_k being violated can be calculated by $Pr[Y_{k,t}\ge C_k]$. By a Chernoff bound [23] with $\delta = 1$, we have

$$\begin{aligned} &Pr[\alpha_k Y_{k,t}\ge C_k] = Pr[\alpha_k Y_{k,t}\ge 2\alpha_k E(Y_{k,t})] \\ &\le exp\big(-E(Y_{k,t}\cdot\alpha_k)/3\big) \le exp\big(-C_k/(3\sigma)\big) \\ &\le exp\big(-6\sigma\ln|UE|/(3\sigma)\big) = 1/|UE|^2. \end{aligned} \tag{26}$$

We now show the computing resource capacity of each location Loc_q is violated by a small probability of $\frac{1}{|\mathcal{CL}\cup\mathcal{BS}|^2}$. Clearly, the trained parameter of each dataset will be sent to a single location for aggregation. Recall that in **Algorithm** `ApproS`, the trained parameter of dataset $ds_{k,t}$ is aggregated in location Loq_q with probability $\frac{x'_{k,t,q}}{\sigma}$. Denote by Z_q the amount of trained parameters that are to be aggregated in Loc_q. We have $Z_q = \sum_{ue_k\in UE \&\& ds_{k,t}\in DS_k} Y_{k,t,q}\cdot|ds_{k,t}|$ and $E(Z_q)$ is

$$\begin{aligned} E(Z_q \mid Y_{k',t',q'}=1) &= \sum_{ue_k\in UE, ds_{k,t}\in DS_k} E(Y_{k,t,q})|ds_{k,t}| \\ &\le \sum_{ue_k\in UE, ds_{k,t}\in DS_k} x'_{k,t,q}|ds_{k,t}|/\sigma \\ &\le \sum_{ue_k\in UE, ds_{k,t}\in DS_k} x^*_{k,t,q}|ds_{k,t}|/\sigma \\ &\le C_q/(\beta_q\sigma), \text{ due to Constraint (14).} \end{aligned} \tag{27}$$

Calculating the probability that the capacity of each location is violated is to calculate

$$\begin{aligned} &Pr[\beta_q\cdot Z_q\ge C_q \mid Y_{k',t',q'}=1] \\ &= Pr[Z_q \ge 2\beta_q E(Z_q)\mid Y_{k',t',q'}=1] \\ &\le exp\big(-E(Z_q)/3\big) \le exp\big(-C_q/(3\sigma)\big) \\ &\le exp\big(-(6\sigma\ln|\mathcal{CL}\cup\mathcal{BS}|)/(3\sigma)\big) = 1/|\mathcal{CL}\cup\mathcal{BS}|^2, \end{aligned} \tag{28}$$

by a Chernoff bound [23] with $\delta = 1$. □

Remarks: It is important to note that the theoretical analysis on the resource violation ratio bounds aims to provide an upper bound to capture worst-case scenarios. However, in real-world scenarios, the capacity violation in a UE does not happen often due to multiple safeguards in place, such as rate limiting and buffer management. The proposed algorithm can be easily extended to preserve the resource capacity constraint, by scaling down the amount of the available resource of each UE by the worst-case violation ratio.

We now analyze that each ue_k has a small probability of failing to upload its trained parameters to an aggregator.

Lemma 3: Assuming that $DB \ge 6\sigma\ln T_r$, each $ue_k\in UE$ has a probability $1/(T_r)^2$ of failing to upload its trained parameters to its assigned aggregator, where $DB = \min\Big\{T_r, \gamma_k\frac{C_k}{\alpha_k\sigma} + \max\{\delta_{k,q},\delta_{q,m},\eta_q\}\frac{C_q}{\beta_q\sigma}\frac{|w_m|}{|w^L_{m,k}|} + E_c\Big\}$.

Proof: Showing the lemma is to show that Constraint (15) is violated with a small probability. To this end, we divide Constraint (15) to three parts: **part (1)**: $\sum_{q=1}^{|\mathcal{BS}\cup\mathcal{CL}|}\sum_{t=1}^{\tau-|DS_k|}\big(x_{k,t,q}|ds_{k,t}|\gamma_k\big)$, **part (2)**: $d^{wls}_{m,k}+\xi_m|w_m|$, and **part (3)**: $\sum_{q=1}^{|\mathcal{BS}\cup\mathcal{CL}|}\big(x_{k,t,q}(\delta_{k,q}|w^l_{m,k}| + \delta_{q,m}|w_m| + \xi_q|w^l_{m,k}|)\big)$.

For part (1), we use M_q to denote the delay of local training in ue_k. Following inequality (25), we can get

$$E(M_k \mid Y_{k',t',q'} = 1) \leq \gamma_k C_k / (\alpha_k \sigma).$$

For part (2), we use N_q to denote the amount of trained parameters that is assigned to Loc_q for aggregation. According to inequality (27), we get

$$E(N_q \mid Y_{k',t',q'} = 1) \leq \frac{C_q \max\{\delta_{k,q}, \delta_{q,m}, \eta_q\}}{\beta_q \sigma} \frac{|w_m|}{|w^L_{m,k}|},$$

where $|w_m|/|w^L_{m,k}|$ is the ratio of the maximum size of local model to the minimum size of local model.

For part (3), the delay of aggregating trained local models is not related to the selection of dataset, because the size of a local model depends on the selected neural network. It thus is treated as a constant with an expectation of E_c. Let D_k be the event representing the delay experienced by ue_k.

$$\begin{aligned} E(D_k) &= E(M_k \mid Y_{k',t',q'} = 1) + E(N_q \mid Y_{k',t',q'} = 1) \\ &\leq \min\Big\{T_r, \gamma_k \frac{C_k}{\alpha_k \sigma} + \max\{\delta_{k,q}, \delta_{q,m}, \eta_q\} \frac{C_q}{\beta_q \sigma} \frac{|w_m|}{|w^L_{m,k}|} + E_c\Big\}, \end{aligned}$$

according to Constraint (15). We then get

$$\begin{aligned} &Pr[D_k \geq C_q \mid Y_{k',t',q'} = 1] \\ &= Pr[D_k \geq 2\beta_q E(D_k) \mid Y_{k',t',q'} = 1] \\ &\leq exp\big(-E(D_k)/3\big) \leq exp\big(DB\big) \\ &\leq exp\big(-6\sigma \ln T_r/(3\sigma)\big) = 1/(T_r)^2, \end{aligned} \tag{29}$$

by adopting a Chernoff bound [23] with $\delta = 1$. □

We finally show the approximation ratio of `ApproS`.

Theorem 1: Given an MEC network G and an FCL request, there exists an approximation algorithm `ApproS` for the age-aware data selection and aggregator placement problem for FCL, and its approximation ratio is 2 with a probability of $exp(-\frac{1}{3}(1 - \frac{1}{(T_r)^{2|UE|}}))$, where T_r is the time length of each round of an FCL process.

Proof: We now show the approximation ratio of **Algorithm** `ApproS`. The objective of the age-aware data selection and aggregator placement for FCL in an MEC network with a single request is to minimize the total age of selected datasets in UEs. Let OPT be the optimal age of the problem, i.e., the obtained solution due to solving **ILP**. Let A be the AoD obtained by **Algorithm** `ApproS`. Let $U_{k,t,q}$ be the event that dataset $ds_{k,t}$ of UE ue_k is selected and assigned to location Loc_q. We have $Pr[U_{k,t,q} = 1 \mid U_{k',t',q'} = 1] = x'_{k,t,q}/\sigma$. The obtained solution A may not be a feasible solution if none of the UEs update its trained parameters to aggregators. By lemma 3, the probability that A is feasible is $1 - 1/((T_r)^{2|UE|})$. The expected age obtained by **Algorithm** `ApproS` thus is

$$\begin{aligned} \mathbb{E}(A) &= \sum_{ue_k \in UE; ds_{k,t} \in DS_k} Pr[U_{k,t,q} = 1 \mid U_{k',t',q'} = 1] \\ &\geq \sum_{ue_k \in UE} (1 - 1/(T_r)^2), \text{since} \min\{\tau - t_{k,q}\} = 1, \\ &\geq (1 - 1/(T_r)^{2|UE|}). \end{aligned} \tag{30}$$

To show the approximation ratio of **Algorithm** `ApproS`, we show the following probability $Pr[A \geq (1+\epsilon)OPT]$, where ϵ is a constant with $\epsilon > 0$. Considering that the fractional solution to **LP** is a lower bound of the optimal solution, we have $OPT \geq \mathbb{E}(A)$, meaning that $Pr[A \geq (1+\epsilon)OPT] < Pr[A \geq (1+\epsilon)\mathbb{E}(A)]$. Similar to lemmas (1), (2), and (3), using the Chernoff bound [23] with $\epsilon = 1$, we obtain

$$\begin{aligned} &Pr[A \geq 2OPT] < Pr[A \geq 2\mathbb{E}(A)] \\ &< exp(-\mathbb{E}(A)/3) \\ &< exp(-1/3 + 1/(3(T_r)^{2|UE|}))), \text{due to (30)}, \end{aligned} \tag{31}$$

where T_r is the time length of each round of FCL. □

V. An Online Learning Algorithm for the Age-Aware Data Selection and Aggregator Placement Problem for FCL in an MEC Network With Multiple Requests and Delay Uncertainty

In this section, we propose an online learning algorithm with a bounded regret for the problem for FCL in an MEC network with multiple FCL requests and delay uncertainties.

A. Basic Idea and Definition of Contexts

To minimize the age of data, the delays of transmission and processing play a vital role. Unfortunately, such delays can be uncertain due to the dynamics of both networks and user demands. The delay experienced by each UE ue_k is determined by various factors, such as the congestion level of locations, location and energy status of UEs, and etc. Each of such factors has a significant impact on the probability of successfully updating the trained parameters of each UE. For example, if UEs are busy with its main functionalities [7], [12], it incurs prohibitively long local training delay. With more and more UEs failing to update trained parameters, the accuracy of the global model can be very low. We define a *context* as a combination of factors. Specifically, let θ be a context available in the system. We consider that each context θ consists of model updating and aggregation delays. That is, $\Theta =< \Delta, \Xi >$, where Δ is a vector of the data updating delays ($\delta_{k,q}$ and $\delta_{q,m}$) and Ξ is a vector of the data aggregation delays (ξ_q and ξ_m).

Our basic idea is to find the most influencing context on the delay of processing a dataset. Given the dynamics of networks, the contexts that have the most significant impact on a dataset may vary as the time goes. We thus needs an adaptive and dynamic online learning algorithm that reacts to such network dynamics. A usual method to guarantee such adaptivity is to follow a well-tuned exploit-and-explore method to find the best policy that maps from a context to the selection of a request in R. However, the optimality in such method may not be guaranteed. Considering that the approximate solution can be obtained via algorithm *ApproS* if the context is matched to an FCL request, we thus consider the proposed algorithm `ApproS` as an *oracle* to admit the selected FCL request under the context. In this way, the context selection is adaptively while the optimality of the obtained solution is guaranteed.

Algorithm 2 OL

Input: $G = (\mathcal{CL} \cup \mathcal{DC}, E)$, a set R of FCL requests.
Output: The age of each FCL request in R.

```
for each time slot τ ← 1 . . . T do
    R ← ∅; // the set of selected FCL requests
    for each FCL request r_m ∈ R do
        if request r_m is newly arrived then
            Initialize the weight of request r_m under each context θ as w_{τ,θ,m} = 1;
            Invoke Algorithm ApproS;
        else
            Update p_{θ,m} by Eq. (32);
            Obtain the probability p_m of selecting request r_m by Eq. (33);
            Select request r_m with probability p_m;
            if FCL request r_m is selected then
                Invoke Algorithm ApproS;
    In the end of time slot τ, observe the age information of FCL request r_m, and update its weight by Eq. (34);
```

B. Online Learning Algorithm

We now describe the detailed steps of the contextual online learning algorithm. In each time slot τ, for a newly-arrived FCL request, we invoke **Algorithm** ApproS to implement it. For existing FCL requests, we allow their dataset selection and aggregator location can be dynamically adjusted, which consists of three stages: **stage 1**: request selection based on contextual bandits; **stage 2**: age-aware FCL; and **stage 3**: policy updating.

In **stage 1**, we observe the contexts that influence the age of each FCL request and learn the correlation between contexts and the age of each FCL request. Specifically, each context has a set of probabilities of recommending the FCL requests to adjust the selection of datasets and aggregator placements. Since the MEC network has various uncertain status that influence the contexts and learning accuracy, such probabilities usually are not known in advance and need to be learnt iteratively. Let $p_{\theta,m}$ be the recommendation probability of FCL request r_m under context $\theta \in \Theta$. $p_{\theta,m}$ is dynamically updated as the context changes and more information is revealed to the system. To this end, we maintain a weight of each context θ on FCL request r_m at each time slot τ, which is denoted by $w_{\tau,\theta,m}$. We assume that the recommendation probability of r_m under context θ is proportional to the weight, that is,

$$p_{\theta,m} = (w_{\tau,\theta,m}) \left(\sum_{m'=1}^{|R|} w_{\tau,\theta,m} \right). \tag{32}$$

Let p_m be the probability of selecting FCL request r_m to adjust dataset selection and aggregator placement, then,

$$p_m = \left(\sum_{\theta \in \Theta} p_{\theta,m} \right) / |\Theta|, \tag{33}$$

where $|\Theta|$ is the number of contexts available in the system.

In **stage 2**, we select the FCL requests whose implementation (dataset selection and aggregator assignment) will be adjusted according to probability p_m. That is, if request r_m is selected, its dataset selections and aggregator placements will be adjusted according to the current contexts, by revoking **Algorithm** ApproS. The procedures continue until every request $r_m \in R$ is considered.

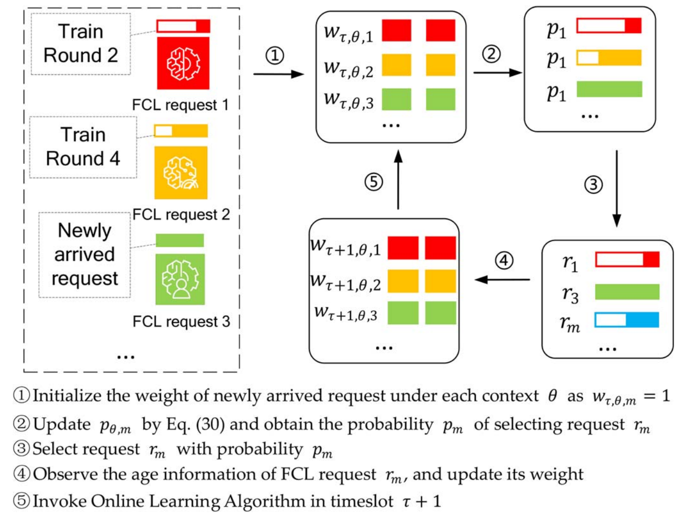


Fig. 2. The framework of online learning algorithm.

In **stage 3**, we obtain the context values of the current time slot τ after adjusting each request r_m in the time round. As such, each request obtains a total age of the locally trained datasets of its UEs. We consider such obtained age as the *loss* of the online learning. According to the updated contexts and obtained loss, we then update the probability of selecting FCL requests. To this end, we update weight $w_{\tau,\theta,m}$ of FCL requests under context θ in the end of each time slot, once the losses of each FCL request are revealed. In other words, we assign penalties to some contexts with higher losses. That is, the penalty of context θ is the total age of FCL request r_m if r_m is selected by context θ, denoted by $\phi_{\tau,\theta,m}$. For each context, we update its weight on request r_m in the next time slot $\tau + 1$ by

$$w_{\tau+1,\theta,m} = w_{\tau,\theta,m} \cdot (1 - \varepsilon)^{\phi_{\tau,\theta,m}}, \tag{34}$$

where ε is a constant in the range of $[0, 1)$. The above-mentioned procedure continues iteratively until all requests are considered. The detailed steps are shown in **Algorithm** 2 and Fig. 2, referred to as OL.

C. Algorithm Analysis

We now analyze the accumulative regret of **Algorithm** OL. We first define the regret within a finite time horizon T as

$$Reg(T) = \Phi(\mathtt{OL}) - E(\Phi_{\mathcal{R}^*}), \tag{35}$$

where $\Phi(\mathtt{OL})$ is the sum of the age of the selected FCL requests by **Algorithm** OL, $\mathcal{R}^*$ is the best subset of requests to adjust their implementation, and $\Phi_{\mathcal{R}^*}$ is the total age if the best subset $\mathcal{R}^*$ of FCL requests are selected for adjustment.

Theorem 2: Given an MEC network G and a set R of FCL requests, there exists an online learning algorithm OL for the age-aware data selection and aggregator placement problem of FCL in an MEC network with multiple requests and delay uncertainty, and its regret is upper bounded by $\frac{\ln T}{\varepsilon} + 2\varepsilon D S_{max}$, where ε is a given constant in the range of $(0, 1/2)$ and DS_{max} is the maximum difference between a maximum age and a minimum age of a request, and $\Phi_{\mathcal{R}^*}$ is the total age if the best subset $\mathcal{R}^*$ of FCL requests are selected.

Proof: In **Algorithm** $\texttt{OL}$, we adjust the weight of each context θ on each FCL request r_m dynamically, according to the AoD of its trained datasets in each time slot. Since the objective of the problem is to minimize the AoD of the datasets that are trained by FCL requests, a context that receives a lower age will be assigned to a higher weight.

Let W_τ and $W_{\tau,\theta}$ denote the total weight of all FCL requests and the requests of context θ, respectively. We then have $W_\tau = \sum_{\theta\in\Theta} W_{\tau,\theta} = \sum_{\theta\in\Theta}\sum_{r_m\in R} w_{\tau,\theta,m} \geq 1$, since the weight of each context θ on an FCL request is initialized to 1 in time slot 1. Let θ^* be the best context with the minimum age, the total weight after T time slots is

$$W_{T+1} > w_{T+1}(\theta^*) = (1-\varepsilon)^{\phi^*} > (1-\varepsilon)^{\Phi_{\mathcal{R}^*}}, \tag{36}$$

due to the definition of costs and the fact that $(1-\varepsilon)^a > (1-\varepsilon)^b$ for any positive values of a and b with $1 \leq a < b$.

We now find a correlation among the total weight of FCL requests and the total age, by showing the relation between $W_{\tau+1}$ and W_τ in the following.

$$\begin{aligned} W_{\tau+1}/W_\tau &= \sum_{\theta\in\Theta}\sum_{r_m\in R} w_{\tau,\theta,m}(1-\varepsilon)^{\phi_{\tau,\theta,m}}/W'_\tau \\ &< \sum_{\theta\in\Theta}\sum_{r_m\in R}(1-\alpha\phi_{\tau,\theta,m}+\beta(\phi_{\tau,\theta,m})^2)\cdot p_{\theta,m} \\ &\quad \text{since } W_\tau \geq 1 \\ &\quad \text{and } \exists\alpha,\beta>0, (1-\varepsilon)^x < 1-\alpha x+\beta x^2 \text{ for all } x>0. \end{aligned} \tag{37}$$

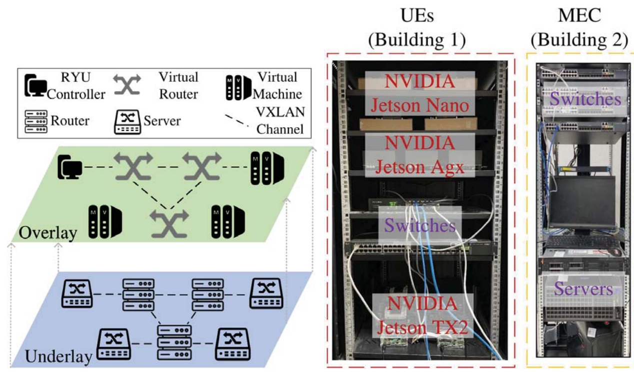
$$\begin{aligned} &\qquad \text{due to the definition of } p_{\theta,k} \\ &= \sum_{\theta\in\Theta}\sum_{r_m\in R} p_{\theta,m} - \alpha\sum_{\theta\in\Theta}\sum_{r_m\in R} p_{\theta,m}\phi_{\tau,\theta,m} \\ &\quad + \beta\sum_{\theta\in\Theta}\sum_{r_m\in R} p_{\theta,m}(\phi_{\tau,\theta,m})^2 \\ &= 1-\alpha\sum_{\theta\in\Theta}\sum_{r_m\in R} p_{\theta,m}\phi_{\tau,\theta,m} \\ &\quad + \beta\sum_{\theta\in\Theta}\sum_{r_m\in R} p_{\theta,m}(\phi_{\tau,\theta,m})^2 \\ &= 1-\alpha\mathbb{E}(\Phi(\texttt{OL},\tau)) + \beta\mathbb{E}(\Phi(\texttt{OL},\tau)^2)), \end{aligned} \tag{38}$$

where $\Phi(\texttt{OL},\tau)$ is the total age of FCL requests at time slot τ. By taking a logarithm on inequality (38), we have

$$\begin{aligned} \ln(W_{\tau+1}/W_\tau) &< \ln(1-\alpha\mathbb{E}(\Phi(\texttt{OL},\tau)) + \beta\lneqq(\texttt{OL},\tau)^2)) \\ &< -\alpha\mathbb{E}(\Phi(\texttt{OL},\tau)) + \beta\mathbb{E}(\Phi(\texttt{OL},\tau)^2), \end{aligned} \tag{39}$$

since $\ln(1-x) < -x$ for any $x\in(0,1)$. In particular, this holds when $(\alpha,\beta)=(\varepsilon,0)$. Considering all time slots within the finite time horizon with $1\leq\tau\leq T$, we have

$$\begin{aligned} &\sum_{\tau=1}^{T}(\alpha\mathbb{E}(\Phi(\texttt{OL},\tau)) - \beta\mathbb{E}(\Phi(\texttt{OL},\tau)^2)) \\ &< -\ln(W_{\tau+1}/W_\tau) < -\ln\Pi_{r=1}^{T}(W_{\tau+1}/W_\tau) \end{aligned}$$



(a) The software architecture (b) The hardware platform

Fig. 3. A prototype MEC network for FCL.

$$\begin{aligned} &= -\ln(W_{\tau+1}/W_\tau) = \ln W_1 - \ln W_{T+1} \\ &< \ln T - \ln(1-\varepsilon)\Phi_{\mathcal{R}^*}. \end{aligned} \tag{40}$$

With $(\alpha,\beta)=(\varepsilon,0)$, we have $\mathbb{E}(\Phi(\texttt{OL})) < \ln T/\varepsilon + (1/\varepsilon)\ln[1/((1-\varepsilon))]\mathbb{E}(\Phi^*_{\mathcal{R}})$. Assuming that $\varepsilon\in(0,1/2)$, we have $\frac{1}{\varepsilon}\ln\frac{1}{(1-\varepsilon)} \leq 1+2\varepsilon$, which means $\mathbb{E}(\Phi(\texttt{OL})) - \Phi_{\mathcal{R}^*}) < \ln T/\varepsilon + 2\varepsilon\mathbb{E}(\phi^*)$. Clearly we have $\phi^* < DS_{max}$. The regret of $\texttt{OL}$ thus is $\mathbb{E}(\Phi(\texttt{OL})) - \Phi_{\mathcal{R}^*}) < \ln T/\varepsilon + 2\varepsilon DS_{max}$. □

VI. Experiments

In this section, we evaluate the performance of the proposed algorithms by implementing a prototype system.

A. Prototype Settings

We built a prototype MEC network with both real hardware devices and virtual network elements, as shown in Fig. 3(a). Specifically, the prototype contains both an underlay with hardware switches, UEs, servers and an overlay with virtual switches. The physical underlay consists of five Huawei S5720-32C-HI-24S-AC switches in two on-campus buildings. Each switch is connected to at least two other switches. There are also five servers with i7-8700 CPU and 16G RAM. Netconf and SNMP protocols are used to manage the switches and the links interconnecting them. As shown in Fig. 3(b), we use heterogeneous edge devices, i.e., 3 NVIDIA Jetson AGX Xavier, 3 NVIDIA Jetson TX2, and 3 NVIDIA Jetson Nano as UEs, to perform local training of FCL requests. These UEs connect to the MEC network via WiFi connections or wired links. The servers and switches in the underlay can be seen as a resource pool with computing and bandwidth resources that can be used to build overlay networks. We use VXLAN to virtualize the resources provided by physical servers and switches in the underlay, to build an overlay that spans the hardware in two different buildings. Specifically, the overlay network has a number of Open vSwitch (OVS) [26] nodes and VMs running in the physical servers.

We implemented a human activity recognition (HAR) application to predict human activities, walking, hopping, phone calls, waving and typing, based on the testbed and a 3-layer fully

TABLE II
EXPERIMENTAL PARAMETERS

Parameter	Value
Switch Model	Huawei S5720-32C-HI-24S-AC
Number of Switches	5
Server CPU	i7-8700
Server RAM	16G
Management Protocols	Netconf, SNMP
UE Devices	3 NVIDIA Jetson AGX Xavier 3 NVIDIA Jetson TX2 3 NVIDIA Jetson Nano
Application	Human Activity Recognition (HAR)
Hidden Channel Shrinkage Ratios	AGX Xavier: 0.5, TX2: 0.25, Nano: 0.125
Data Entry Allocation Strategy	Proportional to computing capability
Dataset Time Slot Distribution	Poisson distribution

connected neural network [16], [27]. We deploy different submodels of the HAR application into UEs in our experiments. Notice that each UE performs the computations in parallel, substantially expediting the training and inference process. The hidden channel shrinkage ratios for Jetson AGX Xavier, Jetson TX2, and Jetson Nano are set to 0.5, 0.25 and 0.125 [5], respectively. The data entries in HARBox are distributed into the 9 UEs in our testbed, and the number of entries allocated to each UE is proportional to its computing capability. We also partition the datasets of each UE into different time slots, where the number and the volume of datasets in each time slot follow a Poisson distribution. We considered that each context is a combination of updating delay from each UE to an aggregation location and aggregation delays of locations. However, to show the applicability of the proposed online learning to other scenarios, we considered other contributing elements that impact the age of the datasets, such as the size of the dataset on each UE, the capacities of computing and bandwidth resources of each potential location, and the accuracy of the models corresponding to the datasets. The experimental parameters are summarized in Table II.

We compared the proposed algorithms `ApproS` and `OL` with the following benchmark algorithms:

- Algorithm `Random` selects the datasets randomly without considering the age of each dataset. The online version of this algorithm is referred to as `OL_Random`.
- Algorithm `Greedy` greedily selects the datasets with the minimum age according to the resource availabilities of base stations and cloudlets, and its online version is denoted by `OL_Greedy`.
- Algorithm `CFL` in [37] divides UEs into a number of clusters, with each cluster having an aggregator to aggregate its local models. Since `CFL` does not consider the AoD and their selections, we use our dataset selection algorithm as the UE selection for `CFL`, for the sake of fairness.

Note that algorithm `CFL` aims to cluster the UEs into different groups and the algorithm does not involve the dataset selection. We thus only compared algorithm `OL` with `CFL` for fairness. In addition, to the best of our knowledge, we are not aware of any algorithm that deals with the dataset selection for a single FCL request, we only compared algorithm `ApproS` with `Random` and `Greedy`, respectively.

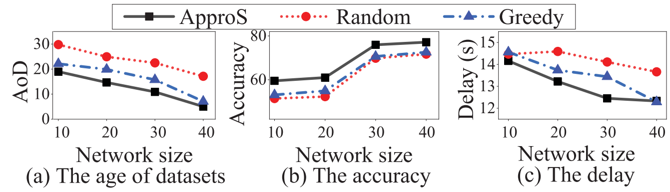


Fig. 4. The impact of the network size on the performance of algorithms `ApproS`, `Greedy` and `Random`.

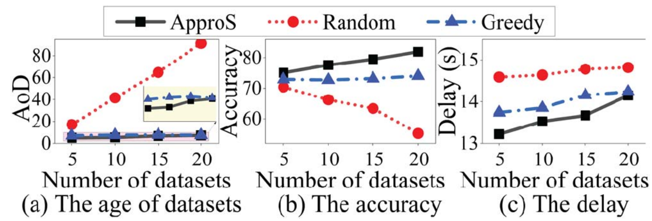


Fig. 5. The impact of the maximum number of datasets in each UE on the performance of algorithms `ApproS`, `Greedy` and `Random`.

B. Performance of Algorithm `ApproS`

We first evaluated the performance of **Algorithm** `ApproS` for the age-aware data selection and aggregator placement problem for FCL in an MEC network with a single request against that of `Random` and `Greedy`, by varying the network size (i.e. the total number of base stations and cloudlets) from 10 to 40. From Fig. 4(a), we can see that `ApproS` achieves the minimum average age of each FCL request. It also achieves the highest accuracy among the three algorithms, as shown in Fig. 4(b).

Specifically, the average age of data required by each implemented FCL request is 47% and 24% lower than that of `Random` and `Greedy`, respectively; while the accuracy by `ApproS` is 12% and 9% higher than that of `Random` and `Greedy`. The reason is that `ApproS` finds a fine trade-off between the age of data and the obtained accuracy of the trained parameters of each UE. Further, we can see that `ApproS` achieves 8% and 3% lower delay than its counterparts, because `ApproS` carefully selects datasets of each UE such that the transmission and processing of the selected dataset meet the delay requirement of FCL requests.

We then dealt with the performance of `ApproS` against `Random` and `Greedy`, by varying the maximum number of datasets in each UE from 5 to 20 and fixing the network size to 40. As shown in Fig. 5(a), we can see that the age achieves by the algorithms is increasing with the growth of the number of datasets in each UE. The rationale behind is that when each UE has more datasets, the size of each dataset has a higher probability of being larger. This leads to higher data transmission and processing delays, which can be reflected in Fig. 5(c). In addition, the accuracy is also increasing, since better datasets can be selected in each UE. Also, the accuracy gap between `ApproS` and its counterparts is enlarging, because `Random` and `Greedy` either randomly selects datasets or only greedily selects the datasets with the minimum age.

We finally investigated the impact of the size of trained parameters on the performance of algorithms `ApproS`, `Random`,

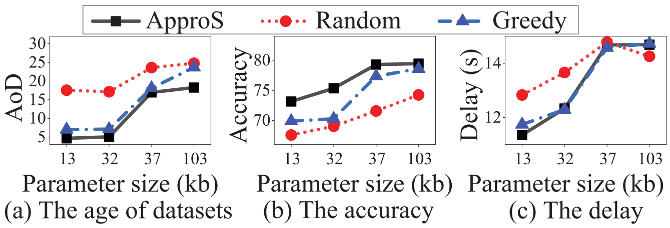


Fig. 6. The impact of the trained parameter size on the performance of algorithms `ApproS`, `Greedy` and `Random`.

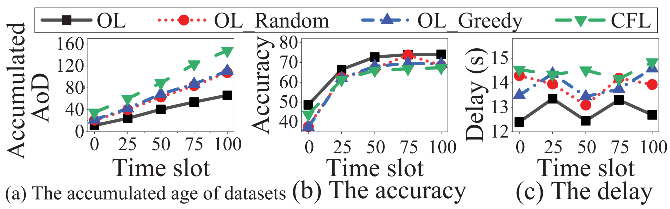


Fig. 7. The performance of algorithms `OL`, `OL_Random`, `OL_Greedy` and `CFL` in online environment.

and `Greedy`, by changing the number of hidden layers and the number of neurons in each layer of the global model and fixing the network size to 40. As shown in Fig. 6(a), we can see that the age of data of the three algorithms increases with the growth of the size of trained parameters. The reasons are two fold: (1) a larger trained parameter incurs a longer transmission delay that is an important part of the age; (2) a longer transmission delay pushes the algorithm to select smaller datasets, and the age is pushed up when smaller datasets become stale. However, the accuracy of the algorithms increases, because a larger trained parameter means more hidden layers and neurons, as shown in Fig. 6(b).

C. Performance of Algorithm `OL`

In the following, we studied the performance of algorithms `OL`, `OL_Random`, `OL_Greedy`, and `CFL`, by fixing the network size to 40 in a finite time horizon T. As shown in Fig. 7(a) and (c), the accumulated age of `OL` is much lower than that of `OL_Random`, `OL_Greedy`, and `CFL`. In particular, the age gap between `OL` and `CFL` is enlarging, as `CFL` does not consider the age of data. Also, Fig. 7(b) depicts that the accuracy of `OL` is higher than that of its counterparts, since it leverages contexts including accuracy of trained parameters and network status to optimize accuracy and AoD.

We then evaluated performance of algorithms `OL`, `OL_Random`, `OL_Greedy`, and `CFL`, by varying the maximum number of datasets in each UE from 5 to 20 while fixing the network size at 40. Experimental results in Fig. 8 showed that algorithm `OL` delivers lower age and higher accuracy than that of the rest algorithms. The performance gap is much larger than that in Fig. 4, since `OL` dynamically learns various contexts and optimizes the age and accuracy.

We finally investigate the impact of the size of trained parameters on the performance of the algorithms, as shown in Fig. 9. We can see the age and accuracy of the four algorithms increase with the growth of the size of trained parameters.

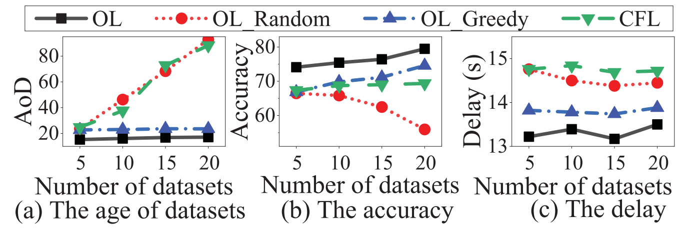


Fig. 8. The impact of the maximum number of datasets on the performance of algorithms `OL`, `OL_Random`, and `CFL`.

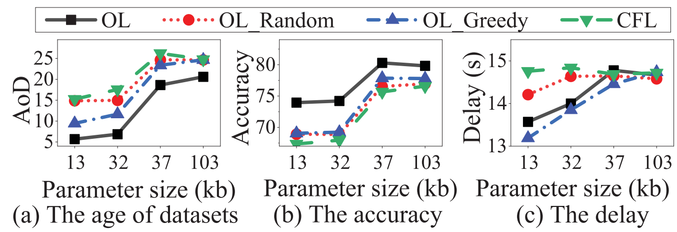


Fig. 9. The impact of the trained parameter size on the performance of algorithms `OL`, `OL_Random`, `OL_Greedy`, and `CFL`.

The delays of the four algorithms gradually approach the length T_r of each round of training. This is because the larger trained parameter will bring more hidden layers, more neurons per hidden layer, and higher transmission delays, ultimately resulting in higher transmission and aggregation delays. UEs may fail to upload their trained parameters in some training rounds, making them require more training rounds to converge to a high accuracy and a high age.

VII. Conclusion and Future Work

In this paper, we investigated the optimization problems of age-aware data selection and aggregator placement for FCL in an MEC network. We first devised an approximation algorithm with a provable approximation ratio for the age-aware data selection and aggregator placement problem for a single federated learning request in an MEC network. We then investigated the problem of age-aware data selection and aggregator placement for FCL with multiple FCL requests and uncertain delays, by devising an online learning algorithm with a bounded regret based on contextual bandits. We also evaluated the performance of the proposed approximation and online algorithms in a real system with various heterogeneous equipment with different computing capabilities. Experiments show that the performance of the proposed algorithms outperform existing studies by achieving 47% lower age of data and 12% higher model accuracy.

The future work stemming from this study includes (1) the neural network architecture of an FCL request that is important roles in the training efficiency. In particular, the neural network architecture may need to fit to edge devices with different computing capacities. We thus will investigate the impact of different neural network architectures on the age of FCL requests; and (2) the implemented test bed can be extended to different types of applications, by adopting the serverless computing architecture. However, we still need to design efficient architectures, software and interface, such that FCL developers could leverage serverless functions to implement the training processes.

REFERENCES

[1] A. Asheralieva and D. Niyato, “Optimizing age of information and security of the next-generation internet of everything systems,” *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20331–20351, Oct. 2022, doi: 10.1109/JIOT.2022.3173912.

[2] K. Bonawitz et al., “Towards federated learning at scale: System design,” in *Proc. Mach. Learn. Syst. (MLSys)*, 2019, pp. 374–388.

[3] A. K. Chathoth, C. P. Necciai, A. Jagannatha, and S. Lee, “Differentially private federated continual learning with heterogeneous cohort privacy,” in *Proc. IEEE Int. Conf. Big Data*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 5682–5691.

[4] D. Chen et al., “Matching-theory-based low-latency scheme for multitask federated learning in MEC networks,” *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11415–11426, Jul. 2021.

[5] E. Diao, J. Ding, and V. Tarokh, “HeteroFL: Computation and communication efficient federated learning for heterogeneous clients,” in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2021, pp. 1–24.

[6] N. Ding, Z. Fang, and J. Huang, “Optimal contract design for efficient federated learning with multi-dimensional private information,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 186–200, Jan. 2021.

[7] A. Drucker, F. Kuhn, and R. Oshman, “On the power of the congested clique model,” in *Proc. ACM Symp. Princ. Distrib. Comput. (PODC)*, New York, NY, USA: ACM, 2014, pp. 367–376.

[8] 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.

[9] C. He, M. Annavaram, and S. Avestimehr, “Towards non-I.I.D. and invisible data with FedNAS: Federated deep learning via neural architecture search,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 5–10.

[10] X. He, S. Wang, X. Wang, S. Xu, and J. Ren, “Age-based scheduling for monitoring and control applications in mobile edge computing systems,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 1009–1018.

[11] J. He, S. Guo, D. Qiao, and L. Yi, “HeteFL: Network-aware federated learning optimization in heterogeneous MEC-enabled internet of things,” *IEEE Internet Things J.*, vol. 9, no. 15, pp. 14073–14086, Aug. 2022.

[12] J. W. Hegeman, G. Pandurangan, S. V. Pemmaraju, V. B. Sardeshmukh, and M. Scquizzato, “Toward optimal bounds in the congested clique: Graph connectivity and MST,” in *Proc. ACM Symp. Princ. Distrib. Comput. (PODC)*, New York, NY, USA: ACM, 2015, pp. 91–100.

[13] “Product specifications: Intel® neural compute stick 2.” Intel ARK. Accessed: Jun. 2022. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/140109/intel-neural-compute-stick-2.html

[14] H. Jiang, M. Liu, B. Yang, Q. Liu, J. Li, and X. Guo, “Customized federated learning for accelerated edge computing with heterogeneous task targets,” *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107569.

[15] G. Lee, M. Jeong, Y. Shin, S. Bae, and S. Yun, “Preservation of the global knowledge by not-true distillation in federated learning,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023, pp. 38461–38474.

[16] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen, “FedMask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking,” in *Proc. 19th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, USA: ACM, 2021, pp. 42–55.

[17] Y. Li et al., “Energy-aware, device-to-device assisted federated learning in edge computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 34. no. 7, pp. 2138–2154, Jul. 2023.

[18] W. Lim et al., “When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 1, pp. 457–466, Jan. 2022.

[19] Y. Liu, Z. Chang, G. Min, and S. Mao, “Average age of information in wireless powered mobile edge computing system,” *IEEE Wireless Commun. Lett.*, vol. 11, no. 8, pp. 1585–1589, Aug. 2022.

[20] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, “HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.

[21] J. Mills, J. Hu, and G. Min, “Communication-efficient federated learning for wireless edge intelligence in IoT,” *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020.

[22] J. Mills, J. Hu, G. Min, R. Jin, S. Zheng, and J. Wang, “Accelerating federated learning with a global biased optimiser,” *IEEE Trans. Comput.*, vol. 72, no. 6, pp. 1804–1814, Jun. 2023, doi: 10.1109/TC.2022.3212631.

[23] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[24] I. Mohammed et al., “Budgeted online selection of candidate IoT clients to participate in federated learning,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5938–5952, Apr. 2021.

[25] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, “Fast-convergent federated learning,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[26] “Production quality, multilayer open virtual switch.” Open vSwitch. Accessed: Sep. 2023. [Online]. Available: https://www.openvswitch.org

[27] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G.Xing, “ClusterFL: A similarity-aware federated learning system for human activity recognition,” in *Proc. 19th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, New York, NY, USA: ACM, 2021, pp. 54–66.

[28] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, “A crowdsourcing framework for on-device federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241–3256, May 2020.

[29] S. Prakash et al., “Coded computing for low-latency federated learning over wireless edge networks,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 233–250, Jan. 2021.

[30] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, “Scheduling for cellular federated edge learning with importance and channel awareness,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.

[31] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, “Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12382–12395, Nov. 2020.

[32] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, “Joint device scheduling and resource allocation for latency constrained wireless federated learning,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.

[33] 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. (INFOCOM)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 1387–1395.

[34] C. Wang, T. Yao, T. Fan, S. Peng, C. Xu, and S. Yu, “Modeling on resource allocation for age-sensitive mobile edge computing using federated multi-agent reinforcement learning,” *IEEE Internet Things J.*, early access, doi: 10.1109/JIOT.2023.3294535.

[35] 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. (INFOCOM)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 1698–1707.

[36] 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.

[37] 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. (INFOCOM)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 1–10.

[38] W. Wu, L. He, W. Lin, and R. Mao, “Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1539–1551, Jul. 2021.

[39] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, “Multi-armed bandit-based client scheduling for federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, Nov. 2020.

[40] 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.

[41] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. V. Poor, “Age-based scheduling policy for federated learning in mobile edge networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 8743–8747.

[42] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.
[43] 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.
[44] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, PMLR, 2021, pp. 12073–12086.
[45] L. Yuan, Y. Ma, L. Su, and Z. Wang, "Peer-to-peer federated continual learning for naturalistic driving action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, 2023, pp. 5250–5259.

Zichuan Xu (Member, IEEE) received the B.Sc. in computer science and M.E. degrees in computer science from Dalian University of Technology, China, in 2011 and 2008, and the Ph.D. degree in computer science from 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 the Ph.D. Advisor with the School of Software at Dalian University of Technology. He is also a "Xinghai Scholar" with Dalian University of Technology. His research interests include mobile edge computing, serverless computing, network function virtualization, Internet of Things, and algorithm design.

Lin Wang received the B.S. degree from Taiyuan University of Technology, Taiyuan, China, in 2016, and the M.S. degree from Inner Mongolia University, Hohhot, China, in 2019. He is currently working toward the Ph.D. degree with the School of Software, Dalian University of Technology. His main research interests include mobile edge computing, task offloading, federated learning, deep learning, and optimization problems.

Weifa Liang (Senior Member, IEEE) received the B.Sc. degree in computer science from Wuhan University, China, in 1984, the M.E. degree in computer science from the University of Science and Technology of China, in 1989, and the Ph.D. degree in computer science from Australian National University, in 1998. He is currently a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong, prior to that position, he was a Full Professor with Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, the Internet of Things and digital twins, mobile edge computing (MEC), network function virtualization (NFV), software-defined networking (SDN), design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.

Qiufen Xia (Member, IEEE) received the B.Sc. and M.E. degrees in computer science from Dalian University of Technology, China, in 2012 and 2009, and the Ph.D. degree in computer science from Australian National University, in 2017. She is currently an Associate Professor with 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.

Wenzheng Xu (Member, IEEE) received the B.Sc., M.E., and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, P. R. China, in 2008, 2010, and 2015, respectively. He is currently a Special Associate Professor with Sichuan University and was a Visitor with Australian National University. His research interests include wireless ad hoc and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

Pan Zhou (Senior Member, IEEE) received the B.S. degree in the advanced class from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006, and the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2011. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST, Wuhan, China. He was a Senior Technical Member with Oracle, Inc., American, from 2011 to 2013, Boston, MA, USA. His current research interests include security and privacy, machine learning and big data analytics, and information networks.

Omer F. Rana (Senior Member, IEEE) received the B.S. degree in information systems engineering from Imperial College of Science, Technology and Medicine, London, U.K., the M.S. degree in microelectronics systems design from the University of Southampton, Southampton, U.K., and the Ph.D. 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.