Federated Learning for Online Resource Allocation in Mobile Edge Computing
A Deep Reinforcement Learning Approach

Zheng, Jingjing ; Li, Kai; Mhaisen, Naram; Ni, Wei; Tovar, Eduardo ; Guizani, Mohsen

DOI
10.1109/WCNC55385.2023.10118940

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 IEEE Wireless Communications and Networking Conference (WCNC)

2023 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-9122-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/WCNC55385.2023.10118940

Federated Learning for Online Resource Allocation in Mobile Edge Computing: A Deep Reinforcement Learning Approach

Jingjing Zheng
CISTER Research Centre
Porto, Portugal
zheng@isep.ipp.pt

Kai Li*
CISTER Research Centre
Porto, Portugal
kaili@ieee.org

Naram Mhaisen
Delft University of Technology
Delft, Netherlands
n.mhaisen@tudelft.nl

Wei Ni
CSIRO
Sydney, Australia
wei.ni@data61.csiro.au

Eduardo Tovar
CISTER Research Centre
Porto, Portugal
emt@isep.ipp.pt

Mohsen Guizani
MBZUAI
Abu Dhabi, United Arab Emirates
mguizani@ieee.org

***Abstract*—Federated learning (FL) is increasingly considered to circumvent the disclosure of private data in mobile edge computing (MEC) systems. Training with large data can enhance FL learning accuracy, which is associated with non-negligible energy use. Scheduled edge devices with small data save energy but decrease FL learning accuracy due to a reduction in energy consumption. A trade-off between the energy consumption of edge devices and the learning accuracy of FL is formulated in this proposed work. The FL-enabled twin-delayed deep deterministic policy gradient (FL-TD3) framework is proposed as a solution to the formulated problem because its state and action spaces are large in a continuous domain. This framework provides the maximum accuracy ratio of FL divided by the device's energy consumption. A comparison of the numerical results with the state-of-the-art demonstrates that the ratio has been improved significantly.**

***Index Terms*—Federated learning, mobile edge computing, online resource allocation, deep reinforcement learning.**

I. Introduction

Edge servers with powerful computing capabilities can handle compute-intensive tasks offloaded by mobile edge computing (MEC) devices [1]. This task offloading is vulnerable to wireless communication attacks, such as eavesdropping [2], denial of service attack [3], or blackhole attacks [4]. To avoid divulging edge devices' private data, the authors [5] firstly developed federated learning (FL) to train a global shared model on the edge server, which aggregates local model updates instead of original training data of the edge devices.

Fig. 1 depicts that the selected edge devices concurrently compute updates of the local models based on their private source data, e.g., pulse rate, body temperature, and blood pressure. The server collects the local models instead of the source data from the edge devices and updates a global model that comprehensively combines all the local models. The global model is eventually sent to all devices. The above process is called one communication round. In particular, scheduling edge devices with large training data is capable of ameliorating the learning accuracy of FL. This results in non-negligible energy consumption. But, on the flip side, scheduling edge devices with small data for FL can reduce energy consumption, which leads to a decrease in the learning accuracy of FL. In this work, we propose a new online resource allocation optimization, which is a trade-off between the learning accuracy of FL and the energy consumption of edge devices in each communication round.

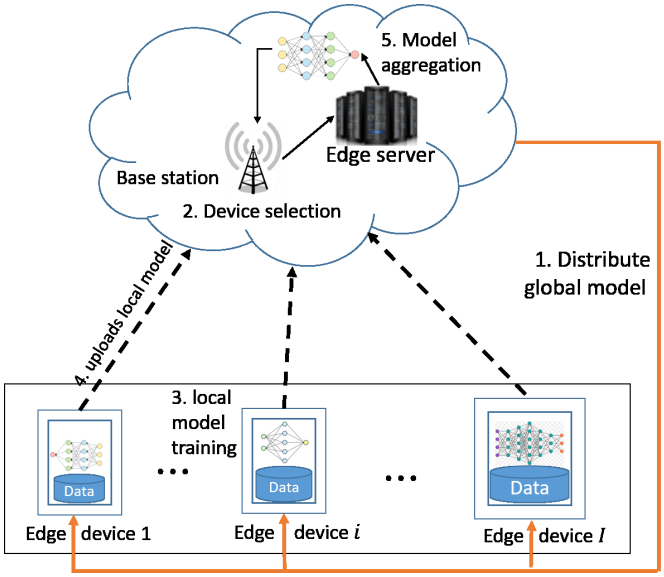

Fig. 1. Framework of FL-enabled edge devices. Each selected edge device downloads the global model from the edge server, then uses its source data to update a local model. The edge server aggregates the local models to update the global model, which is distributed to the edge devices.

The local training data and bandwidth have a dynamic

* Corresponding author.

978-1-6654-9122-8/23/$31.00 ©2023 IEEE

behavior and change instantaneously. The edge server is blind to the remaining energy of the respective edge devices and time-varying channel conditions amidst the base station and the edge device. Therefore, the online resource allocation problem is formulated as a partially observable Markov decision process (POMDP). The network state, in detail, is composed of the data size, bandwidth and remaining battery energy of the edge devices, and channel gains amid the base station and the edge devices. FL-TD3 is proposed for learning the dynamic behavior of the network state in the context of its state and action space, maximizing the ratio, i.e., FL's learning accuracy divided by the energy consumption of the edge device of choice in every communication round (a ratio discussed in this paper). As part of FL-TD3, the edge devices are scheduled and transmitted to maximize their power efficiency.

The rest of this paper is organized as follows. In Section II, we introduce the literature on FL-based resource allocation in MEC. Section III presents the system model. Section IV details the proposed FL-TD3 framework. The evaluations of FL-TD3 are presented in Section V. Finally, Section VI concludes this work.

II. RELATED WORK

Assume that all the edge devices have the same amount of data and computing resources. Federated Averaging (FedAvg) algorithm [6] randomly selects the edge devices in one training iteration to participate in local training and synchronously aggregates local models. Considering inconsistent data quantity and computing resources at the edge devices, FedCS, an FL protocol, was developed [7] to improve the FL accuracy. FedCS enables the server to collect a maximum number of local models, which is allowed by the bandwidth capability.

To decrease the training latency of FL systems, the authors in [8] develop a multi-armed bandit-based algorithm to schedule the devices' local model offloading given unknown channels and computing power of the edge devices. The authors of [9] study three scheduling policies, namely, random scheduling (RS), round robin (RR), and proportional fair (PF), to figure out the convergence rate for FL in the presence of limited bandwidth and wireless inter-cell interference. According to the analysis, FL with PF is faster than RS or RR under high SINR thresholds, while RR outperforms RS and PF under low SINR thresholds. To decrease the energy consumption of edge devices and the training time of the FL, a deep Q-learning-based resource concerning data, energy, and CPU cycle allocation for the edge devices is developed in [10].

Since the optimization problem is NP-hard, our previous work [11] presented a heuristic algorithm, federated learning for energy-accuracy-based client selection (FedAECS), to approximate the optimal client scheduling policy offline, subject to the constraints of the energy consumption and the FL accuracy. Distinctively, this work focuses on an online edge device scheduling policy in an actual case, where the edge server is blind to the remaining energy of the respective edge devices and time-varying channel gains amidst the base station and the heterogeneous edge device. Taking into account the large state and action space in a continuous domain, a new online FL-TD3 framework is proposed to trade-off between the FL accuracy against energy consumption of the selected edge devices. We, in addition, compare the performance of the proposed FL-TD3 with the state-of-the-art FedAECS [11], FedCS [7] and FedAvg [6].

III. SYSTEM MODEL

A. Energy Model

Each chosen edge device's energy consumption consists of two parts, one is local model training, and the other is local model transmission [12], [13]. Assuming that there are I number of edge devices, each edge device $i \in [1, I]$ requires c_i CPU cycles per bit to train a data sample. The local model training for $D_{t,i}$ data samples require $c_i D_{t,i}$ number of CPU cycles. In the t-th communication round, the time consumed by the edge device i to train the local model is

$$\tau_{t,i}^{train} = \frac{L c_i D_{t,i}}{f_i}, \tag{1}$$

where L is the number of epochs of FL local model training at the edge device i, f_i is the computation capacity of edge device i and gauged in CPU cycles per second. Furthermore, the energy consumption of local model training is

$$E_{t,i}^{cmp} = L \zeta_i c_i D_{t,i} f_i^2, \tag{2}$$

where ζ_i represents the effective capacitance coefficient of the computing chipset of the edge device i.

Let $b_{t,i}$, $P_{t,i}$, and $G_{t,i}$ denote the bandwidth, transmit power assigned by the edge server to the edge device i, and the uplink channel gain of edge device i, respectively. The achievable uplink transmit rate, $r_{t,i}^{up}$, can be written as

$$r_{t,i}^{up} = b_{t,i} \log_2(1 + \frac{P_{t,i} G_{t,i}}{N_0 b_{t,i}}), \tag{3}$$

where N_0 stands for the Gaussian noise, $P_{t,i}$ is a continuous variable within the range of P_i^{min} and P_i^{max}.

Let F_t^s and $H_{t,i}$ denote the edge server's transmit power and downlink channel gain, respectively. The downlink transmit rate of device i can be represented by

$$r_{t,i}^{down} = b_{t,i} \log_2(1 + \frac{F_t^s H_{t,i}}{N_0 b_{t,i}}). \tag{4}$$

The download time of the FL global model is

$$\tau_{t,i}^{down} = \frac{\mathfrak{S}_g}{r_{t,i}^{down}}, \tag{5}$$

where $\mathfrak{S}_g$ represents the global model size. Similarly, given the size of total local parameters, $\mathfrak{S}$, the required transmission time of the local parameters can be given by

$$\tau_{t,i}^{up} = \frac{\mathfrak{S}}{r_{t,i}^{up}}. \tag{6}$$

By replacing (3) with (6), the energy consumption of transmitting the local parameters can be further written as

$$E_{t,i}^{up} = P_{t,i} \tau_{t,i}^{up} = \frac{P_{t,i} \mathfrak{S}}{b_{t,i} \log_2(1 + \frac{P_{t,i} G_{t,i}}{N_0 b_{t,i}})}. \tag{7}$$

The chosen edge device's total energy consumption in the t-th communication round is given by

$$E_{t,i}^{c} = E_{t,i}^{cmp} + E_{t,i}^{up}. \tag{8}$$

Moreover, we denote $\Delta E_{t,i}$ as the amount of energy harvesting of each edge device i. In t-th communication round, the remaining energy of the selected edge device is

$$E_{t,i} = E_{t-1,i} - E_{t-1,i}^{c} + \Delta E_{t,i}. \tag{9}$$

It is noted that the total energy consumption of the selected edge device in each communication round must be less than or equal to its remaining energy, namely, $E_{t,i}^{c} \leq E_{t,i}$.

B. Accuracy of FL

Clearly, if $a_{t,i} = 1$, namely device i is scheduled in t-th communication round, otherwise, $a_{t,i} = 0$. According to [14], the accuracy of FL, $\Gamma(a_{t,i})$, can be denoted by

$$\Gamma(a_{t,i}) = \log(1 + \sum_{i=1}^{I} \mu_i a_{t,i} D_{t,i}) \quad \forall t \in \mathcal{T}, \tag{10}$$

where $\mu_i > 0$ is a system parameter [14].

C. Evenness of Edge Devices Scheduling

To measure the data size evenness of the chosen edge devices, according to literature [15], we calculate the expectation of the subtraction value amidst the whole data volume of all edge devices and the data volume of the chosen edge devices, the evenness of edge device scheduling can be defined as the normalized expectation, i.e.,

$$\Delta_t = \frac{\mathbb{E}\Big[\nu \sum_{i=1}^{I} D_{t,i} - a_{t,i} D_{t,i}\Big]}{\sum_{i=1}^{I} D_{t,i}} \quad \forall t \in \mathcal{T}, \tag{11}$$

where $\nu \in (0, 1]$ is a weighted parameter, and $D_{t,i}$ follows normal or uniform distribution.

IV. DRL-based Edge Devices Scheduling and Resource Allocation

The local training data and bandwidth have a dynamic behavior and change instantaneously. The edge server is blind to the remaining energy of the respective edge devices and time-varying channel gains amidst the base station and the edge devices. Therefore, a POMDP is formulated for resource allocation.

Action and State Space: The action space of the POMDP includes two optimization variables, that is, the selection of the edge devices and the transmission power of the selected edge devices, which is given by

$$A \in \{(a_{\phi,i}, P_{\phi,i}), i = 1, \cdots, I\}, \tag{12}$$

where $a_{\phi,i} \in \{0, 1\}$ and $P_{\phi,i} \in [P_i^{min}, P_i^{max}]$.

The state space, S_ϕ, is composed of data size, remaining battery energy and bandwidth of the edge devices, and channel gains of both the edge devices and the edge server, i.e.,

$$S_\phi = \{D_{\phi,i}, E_{\phi,i}, B_{\phi,i}, G_{\phi,i}, H_{\phi,i}\}. \tag{13}$$

Observation Space: The edge server partially observes the network state, which means the state of the unselected edge devices is unable to be observed. The state observation S_ϕ^o is packed in the local model and uploaded to the edge server. Particularly, $S_\phi^o \in S_\phi$ is presented as

$$S_\phi^o = \{(D_{\phi,i}, E_{\phi,i}, B_{\phi,i}, G_{\phi,i}, H_{\phi,i})_o, i = 1, \cdots, I\}. \tag{14}$$

Reward: The instant reward as the objective function of optimization, also called AE (Accuracy to Energy) gain, consists of two terms, the first term is the ratio, the second term is the penalty factor caused by the unevenness of edge devices scheduling, i.e.,

$$R_\phi = \frac{\Gamma(a_{\phi,i})}{\sum_{i=1}^{I} a_{\phi,i} E_{\phi,i}^{c}} - \Delta_\phi. \tag{15}$$

To assess the action chosen by a policy π_ω with parameters ω, we denote the optimal action-value function as

$$Q_{\pi_\omega}(S_\phi^o, A_\phi) = \max_{\pi \in \Pi} \mathbb{E}_{S_\phi^o}^{\pi_\omega}\Big\{ \sum_{n=0}^{\infty} \gamma^n R_\phi \Big\}, \tag{16}$$

where Π and $\gamma \in [0, 1]$ are the set of all policies and the discount factor, respectively. $\mathbb{E}_{S_\phi^o}^{\pi_\omega}\{\cdot\}$ refers to take the expectation with respect to π_ω and S_ϕ^o. Following the Bellman equation [16], we can rewrite (16) as

$$Q_{\pi_\omega}(S_\phi^o, A_\phi) = \max_{\pi_\omega \in \Pi} \mathbb{E}_{S_\phi^o}^{\pi_\omega}\Big\{ R_\phi + \gamma Q_{\pi_\omega}(S_{\phi'}^o, A_{\phi'}) \Big\}, \tag{17}$$

where $S_{\phi'}^o$ and $A_{\phi'}$ are the next state observation and next action, respectively. The optimal action guarantees the maximized AE gain, which is given by

$$A_\phi^* = \arg \max_{\pi_\omega \in \Pi} \mathbb{E}_{S_\phi^o}^{\pi_\omega}\Big\{ R_\phi + \gamma Q_{\pi_\omega}(S_{\phi'}^o, A_{\phi'}) \Big\}. \tag{18}$$

A. TD3 on the Edge Server

In view of the large continuous action space, an actor-critic method is used to solve the POMDP problem, where the edge server simultaneously learns a policy function and a value function. The actor, also called policy, will take action in a continuous domain. The critic is taken to evaluate how well the actor takes action. The policy π_ω utilizes the deterministic policy gradient algorithm [17] and takes the gradient of the expected return concerning ω to optimally update,

$$\nabla_\omega J(\omega) = \mathbb{E}_{\pi_\omega}\big[\nabla_{A_\phi} Q_{\pi_\omega}(S_\phi^o, A_\phi)|_{a_\phi = \pi(S_\phi^o)} \nabla_\omega \pi_\omega(S_\phi^o)\big], \tag{19}$$

where $Q_{\pi_\omega}(S_\phi^o, A_\phi)$ is estimated by the critic neural network [18] of a differentiable function $Q_\beta(S_\phi^o, \beta_\phi)$ with parameters β. To update $Q_\beta(S_\phi^o, A_\phi)$, the critic neural network by manipulating the parameter β to minimize the loss between the target value and $Q_\beta(S_\phi^o, A_\phi)$, i.e.,

$$\min_\beta \mathbb{E}\Big[\Big(R_\phi + \gamma Q_{\beta'}\big(S_{\phi'}^o, \pi_\omega(S_{\phi'}^o)\big) - Q_\beta(S_\phi^o, A_\phi)\Big)^2\Big], \tag{20}$$

where $\beta^{'}$ is the periodical update parameter of the target critic network, $\pi_{\omega}(S^{o}_{\phi^{'}})$ stands for the action taken by the target policy network in the next state observation $S^{o}_{\phi^{'}}$.

However, to address the overestimation bias issue caused by updating the critic networks with the value function, TD3 adopted two approximately independent target critic networks $\{Q_{\beta_1}, Q_{\beta_2}\}$ to evaluate the value function, i.e.,

$$\begin{aligned} z^{tar}_{\phi,1} &= R_{\phi} + \gamma Q_{\beta^{'}_1}(S^{o}_{\phi^{'}}, \pi_{\omega}(S^{o}_{\phi^{'}})), \\ z^{tar}_{\phi,2} &= R_{\phi} + \gamma Q_{\beta^{'}_2}(S^{o}_{\phi^{'}}, \pi_{\omega}(S^{o}_{\phi^{'}})), \end{aligned} \tag{21}$$

where the smaller target critic network value is selected for updating the value function, $z^{tar}_{\phi} = \min_{m=1,2}\{z^{tar}_{\phi,m}\}$.

B. The Proposed Framework of FL Resource Allocation

The TD3-based edge devices scheduling and resource allocation in a continuous action space constitute the proposed FL-TD3 framework, as depicted in Fig. 2. The edge server (a.k.a. agent) randomly selects the edge devices and allocates the transmission power to them in the first K steps to participate in FL training. The environment returns the reward R_{ϕ} and transforms new state $S^{o}_{\phi^{'}}$. Meanwhile, the edge server takes policies randomly in the first K steps and stores the transition, *previous action of the agent, current state observation, current action of the agent, reward, next state observation*, namely, $\{(A_{\phi^-}, S^{o}_{\phi}, A_{\phi}, R_{\phi}, S^{o}_{\phi^{'}})\}$, into the replay buffer $\mathcal{M}$. After K steps, the server samples random mini-batch of transitions $\{(A_{\phi^-}, S^{o}_{\phi}, A_{\phi}, R_{\phi}, S^{o}_{\phi^{'}})\}$ from $\mathcal{M}$ to train the actor and critic networks. The actor network with exploration noise can ascertain the action.

Moreover, FL-TD3 also follows another trick of TD3 that renews both actor and critic target network every l intervals. Algorithm 1 details the proposed FL-TD3 edge devices scheduling and transmit allocation policy. Given the I quantity of edge devices, each state observation has five elements, and T iterations are required before FL-TD3 terminates. In sum, the complexity of FL-TD3 is $\mathcal{O}\big(T[(N_{pc1}-1)n^2_{pc1} + (N_{pc2}-1)n^2_{pc2} + (N_{pa}-1)n^2_{pa} + 7I \times (n_{pa}+n_{pc1}+n_{pc2}) + 12 \times ((7I)^2 + 7I)]\big)$, where N_{pa}, N_{pc1} and N_{pc2} are the quantity of the hidden layers of the respective actor network, critic network 1 and critic network 2; n_{pa}, n_{pc1} and n_{pc2} are the quantity of neurons in the hidden layer of the respective actor network, critic network 1 and critic network 2.

Algorithm 1: The proposed FL-TD3 edge devices scheduling and transmit power allocation

Initialize $\{Q_{\beta_1}(S^{o}_{\phi}, A_{\phi}, A_{\phi^-}), Q_{\beta_2}(S^{o}_{\phi}, A_{\phi}, A_{\phi^-})\}$ and $\pi_{\omega}(S^{o}_{\phi}, A_{\phi^-})$ with random parameters β_1, β_2, ω.
Initialize $\beta^{'}_1 \leftarrow \beta_1, \beta^{'}_2 \leftarrow \beta_2, \omega^{'} \leftarrow \omega$.
Initialize S^{o}_{ϕ}, $A_{\phi^-} \leftarrow 0$, $\mathcal{M}$.
for $\phi = 1, \cdots, T$ **do**
  if $\phi \leq K$ **then**
    Explore K steps haphazardly, get R_{ϕ} and $S^{o}_{\phi^{'}}$, and store $(A_{\phi^-}, S^{o}_{\phi}, A_{\phi}, R_{\phi}, S^{o}_{\phi^{'}})$ of the K steps into $\mathcal{M}$.
  else
    Take $A_{\phi} \sim \pi_{\omega}(S^{o}_{\phi}, A_{\phi^-}) + \jmath, \jmath \sim \mathcal{N}(0, \sigma)$.
    Assigns $P_{\phi,k}$ to the selected devices.
    Server observes $S^{o}_{\phi^{'}}$, calculates R_{ϕ}, and stores $(A_{\phi^-}, S^{o}_{\phi}, A_{\phi}, R_{\phi}, S^{o}_{\phi^{'}})$ into $\mathcal{M}$.
    Sample $\{(A_{\phi^-}, S^{o}_{\phi}, A_{\phi}, R_{\phi}, S^{o}_{\phi^{'}})_k\}^{K}_{k=1}$ from $\mathcal{M}$.
    Obtain $\widehat{A}_{\phi^{'}} \leftarrow \pi_{\omega^{'}}(S^{o}_{\phi^{'}}, A_{\phi}) + \widehat{\jmath}, \widehat{\jmath} \sim$ $\mathrm{clip}(\mathcal{N}(0, \widehat{\sigma}), -\rho, \rho)$.
    $z^{tar}_{\phi} \leftarrow \min_{m=1,2}\{z^{tar}_{\phi,m}\}$.
    Update critic value $\beta_m \leftarrow \arg\min_{\beta_m} \mathbb{E}[z^{tar}_{\phi} - Q_{\beta_m}(S^{o}_{\phi}, A_{\phi^-})]^2$.
    if ϕ *mod* l = *0* **then**
      Renew ω according to (19).
      Renew target networks:
      $\beta_{m^{'}} \leftarrow \theta\beta_m + (1-\theta)\beta_{m^{'}}$.
      $\omega^{'} \leftarrow \theta\omega + (1-\theta)\omega^{'}$.
    end
  end
end

V. Numerical Results

In this section, the proposed FL-TD3 is implemented with Python 3.9. All experiments are conducted using the PyTorch framework, which is installed on an open-source Linux kernel working environment with Ubuntu 16.04 system. All experiments are done on 2 Nvidia GPUs (graphics processing units), i.e., GeForce GTX 1060 and GeForce RTX 2060 with 3 GB memory and 6 GB memory, respectively.

A. Simulation parameters

In our simulation, let $T = 1000$, $L = 4$ and $c_i = 20$ cycles/bit, respectively. I increases from 10 to 80 in intervals of 10. f_i of the edge device follows uniform distribution in [2, 4] GHz. $P_{t,i}$ and F^{s}_{t} follow uniform distribution in [0.1, 60] W and [100,1000] W, respectively. $G_{t,i}$ and $H_{t,i}$ follow uniform distribution in $[10^{-3}, 10^{-1}]$ dB and $[10^{-1}, 10]$ dB, respectively. The amount of harvested energy $\Delta E_{t,i}$ follows uniform distribution in [50, 200] J. The global model's size and local parameters' size $\mathfrak{S}_g = 1 \times 10^4$ bits and $\mathfrak{S} = 5 \times 10^4$ bits, respectively. System parameter $\mu_i = 4.2 \times 10^{-9}$, $\epsilon_i = 1.2 \times 10^{-28}$, $N_0 = 1.0 \times 10^{-8}$, $\nu = 1.0$. Both critic networks' and actor networks' learning rates are 3×10^{-4}. The rest of the parameters, $\theta = 5 \times 10^{-3}$, $l = 10$, $\gamma = 0.99$, $K = 45$, $|\mathcal{M}| = 5 \times 10^5$, $\sigma = 0.5$, and $\rho = 0.5$.

B. Performance Analysis

For the evaluation of FL-TD3, three latest edge devices scheduling approaches of FL as benchmarks are compared; namely, FedAECS [11], FedCS [7] and FedAvg [6].

- **FedAECS:** The server schedules the edge devices to meet the preset ratio, as well as the requirements of FL accuracy and limited bandwidth in each communication round.

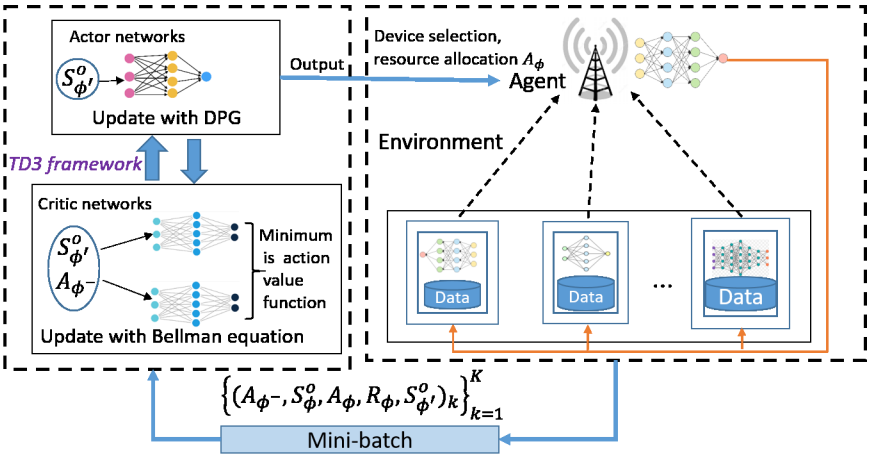

Fig. 2. The proposed FL-TD3 framework.

- **FedCS:** Considering the limited bandwidth, the server aggregates as many edge devices as possible for FL in each communication round.
- **FedAvg:** Considering the limited bandwidth, the server determines the quantity of the edge devices, and indiscriminately schedules the devices.

Fig. 3 depicts the AE gains vary with the t-th communication round, which ranges from 1 to 1000 with 40 edge devices. The data size $D_{t,i}$ varies in [2, 10] MB, the bandwidth $b_{t,i}$ varies in [10, 50] KHz. The biggest AE gain of FL-TD3 improved, on average, by 19.87%, 70.73% and 75.15% as compared with FedAECS, FedCS and FedAvg, respectively. This is because FL-TD3 can leverage experience replay to select the edge device. Nevertheless, FedAECS, FedCS, and FedAvg advantage of historical information.

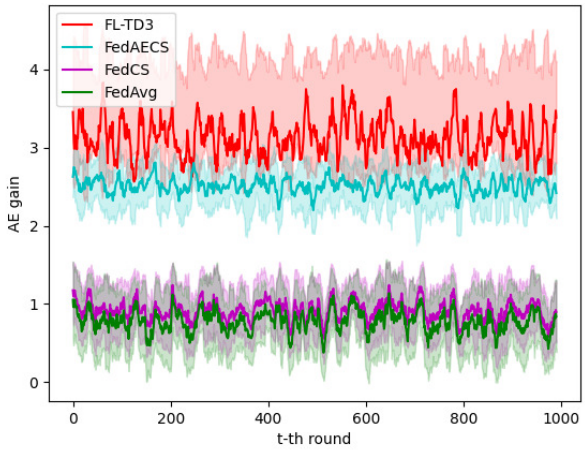

Fig. 3. Comparison of AE gains obtained by FL-TD3 with those obtained by three latest methods

Fig. 4 shows the AE gains obtained by FL-TD3, where I increases from 10 to 80. Generally, AE gains decrease as the quantity of edge devices increases, and FL-TD3 obtains higher AE gain than others. Specifically, while the number of devices is 40, 60 and 80, the AE gains of FL-TD3 increase mainly because accuracy dominates it.

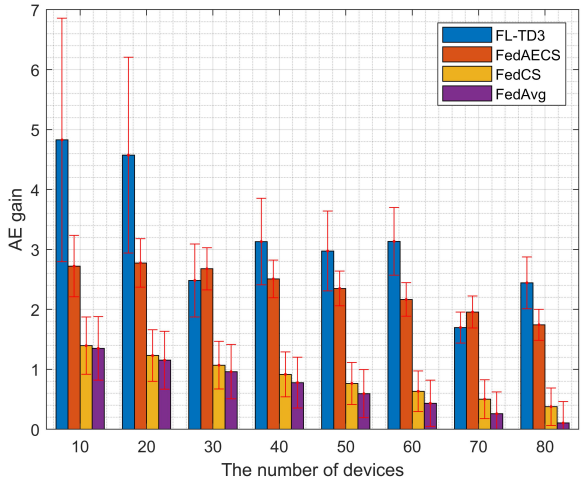

Fig. 4. AE gains comparison with different number of edge devices using FL-TD3, FedAECS, FedCs and FedAvg

Fig. 5 describes AE gains that vary with the average data size while maintaining the invariant variance. On the whole, the AE gains of benchmarks rise with the increment of the data size. However, the AE gains of FL-TD3 decrease while the average data size is 6, 8 and 10 MB. This is because energy consumption dominates. In addition, the AE gains achieved by FL-TD3 are about twice the AE gains obtained by FedAECS, while the average data size is 1, 2 and 5 MB. The reason is that the FL-TD3 deliberates transmission power allocation.

Fig. 6 depicts the AE gains vary with the average bandwidth while maintaining the invariant variance. Generally, the AE gain of benchmarks increases as average bandwidth increases. A decrease in AE gain occurs when the average bandwidth

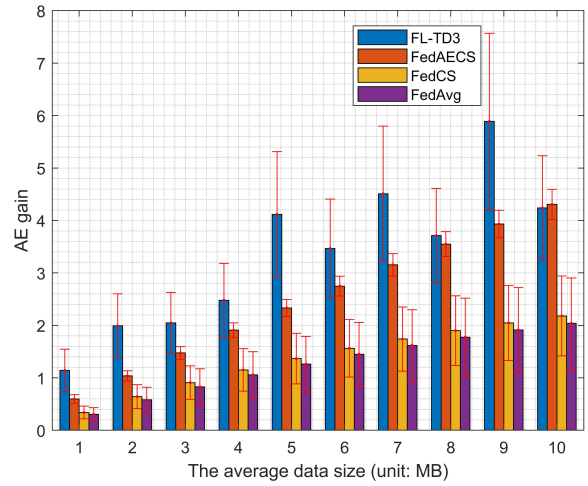

Fig. 5. Comparison of AE gains, where the data size is uniformly distributed and the constant variance is 0.2 MB

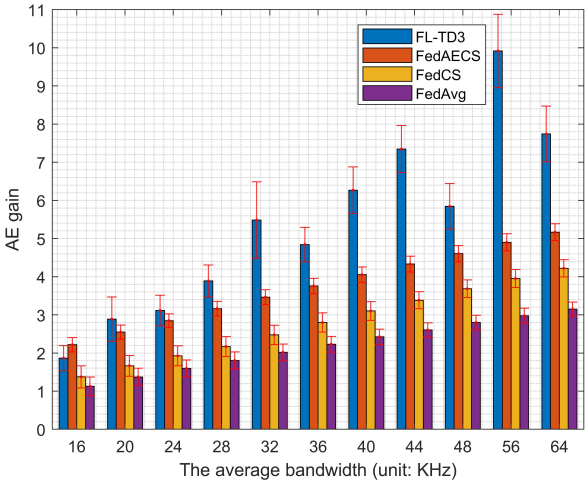

Fig. 6. AE gains change with bandwidth following a normal distribution while maintaining the invariant variance (4 kHz)

drops to 36, 44, and 56 kHz. This is due to the fact that the increase in energy consumption is more significant than the increase in FL accuracy. Moreover, FL-TD3 obtains the highest AE gain whatever the bandwidth is.

VI. Conclusion

In this work, we have proposed FL-TD3, which is a new DRL-based online resource allocation for FL in MEC. The optimization of the edge devices scheduling and the transmit power allocation has been formulated as a POMDP, where the network state is composed of data size, the remaining battery energy and bandwidth of the edge devices, and channel gains of both the edge devices and the edge server. The proposed FL-TD3 optimally selects the edge devices in each communication round of FL training at the edge server while allocating the transmit power to the selected edge devices. Compared with the state-of-the-art works, numerical results have demonstrated that the ratio has been significantly improved.

Acknowledgment

This work was supported by the CISTER Research Unit (UIDP/UIDB/04234/2020) and by project ADANET (PTDC/EEI-COM/3362/2021), financed by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology).

References

[1] K. Li, Y. Cui, W. Li, T. Lv, X. Yuan, S. Li, W. Ni, M. Simsek, and F. Dressler, "When internet of things meets metaverse: Convergence of physical and cyber worlds," *IEEE Internet of Things Journal*, 2022.

[2] X. Zhou, B. Maham, and A. Hjorungnes, "Pilot contamination for active eavesdropping," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 903–907, 2012.

[3] T. Jamal, P. Amaral, A. Khan, A. Zameer, K. Ullah, and S. A. Butt, "Denial of service attack in wireless lan," *ICDS 2018*, vol. 51, 2018.

[4] H. Kalkha, H. Satori, and K. Satori, "Preventing black hole attack in wireless sensor network using hmm," *Procedia computer science*, vol. 148, pp. 552–561, 2019.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[6] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016.

[7] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC*, pp. 1–7, IEEE, 2019.

[8] B. Xu, W. Xia, J. Zhang, T. Q. Quek, and H. Zhu, "Online client scheduling for fast federated learning," *IEEE Wireless Communications Letters*, 2021.

[9] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, 2020.

[10] T. T. Anh, N. C. Luong, D. Niyato, D. I. Kim, and L. Wang, "Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 5, pp. 1345–1348, 2019.

[11] J. Zheng, K. Li, E. Tovar, and M. Guizani, "Federated learning for energy-balanced client selection in mobile edge computing," in *International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1942–1947, IEEE, 2021.

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

[13] J. Zheng, K. Li, N. Mhaisen, W. Ni, E. Tovar, and M. Guizani, "Exploring deep reinforcement learning-assisted federated learning for online resource allocation in privacy-preserving edgeiot," *IEEE Internet of Things Journal*, 2022.

[14] W. Y. B. Lim, J. Huang, Z. Xiong, J. Kang, D. Niyato, X.-S. Hua, C. Leung, and C. Miao, "Towards federated learning in uav-enabled internet of vehicles: A multi-dimensional contract-matching approach," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[15] X. Lyu, C. Ren, W. Ni, H. Tian, R. P. Liu, and E. Dutkiewicz, "Optimal online data partitioning for geo-distributed machine learning in edge of wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2393–2406, 2019.

[16] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[17] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, pp. 387–395, PMLR, 2014.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.