Privacy is not Free: Energy-Aware Federated Learning for Mobile and Edge Intelligence

Wenqi Yang[1], Yang Zhang[1,*], Wei Yang Bryan Lim[2,3], Zehui Xiong[3], Yutao Jiao[4], Jiangming Jin[5]
[1]School of Computer Science and Technology, Wuhan University of Technology, China
[2]Alibaba Group
[3]Alibaba-NTU Joint Research Institute, Nanyang Technological University, Singapore
[4]Army Engineering University of PLA, China
[5]TuSimple
[*]Email: yangzhang@whut.edu.cn

***Abstract*—In mobile and edge intelligence systems, federated learning (FL) enables local data training and learning model sharing without obtaining actual data from mobile and edge users, which are data owners. Data training processes are performed at the user side with only trained gradients passed to an aggregator, i.e., learning server. The learning server continually trains and updates corresponding learning models by collecting gradients. The updated learning models are delivered back to the mobile and edge users for improved data training results. Despite the advantages of federated learning in preserving privacy, the local data training process will consume an adequate amount of energy from the perspective of mobile and edge users. In mobile and edge intelligence systems, mobile users may not always prefer to apply federated learning to reduce energy consumption. There is a trade-off between applying federated learning to reserve data privacy and updating actual data for learning servers to train. In this work, a Markov decision process (MDP) based system model is proposed for mobile and edge users to make federated learning decisions to optimize long-term performance in terms of utility function consists of data training reward and data processing delay. A deep reinforcement learning approach is proposed to solve the MDP problem in highly dynamic systems with a large state space.**

***Index Terms*—Federated learning, edge intelligence, Markov decision process, reinforcement learning**

I. Introduction

Federated learning (FL) is a technique to deploy machine learning models to solve the information isolated island problem among various distributed users with privacy-preserving features [1]. In a conventional FL model, *FL workers* are data owners who need to utilize the data, e.g., for making predictions and optimal decisions, by training the data and applying machine learning models on the data. Instead of uploading the data to a centralized server for training, an FL worker requests learning models from a corresponding *FL server*, and trains the data set locally with the requested learning model. After the local training is performed, the FL worker only uploads the trained gradient of the data set to the FL server to avoid data leakage. The FL server aggregates all the collected gradients and calculates the averaged gradients to update the learning model without accessing actual data on the user side.

Existing studies on federated learning mainly focus on overcoming statistical and privacy challenges [2], improving safety [3] and mitigating the pressure of data communication [4], which are mainly caused by traditional and centralized machine learning. A privacy-preserving asynchronous FL mechanism (PAFLM) has been proposed for edge network computing in [5]. An enhanced FL model has been proposed [6] by proposing an asynchronous learning strategy on the clients and a temporally weighted aggregation of the local models on the server. A conceptual application employing FL on malware classification has been proposed in [7]. An alliance privacy protection method has been studied [8] based on joint learning to promote collaborative machine learning among multiple model owners in mobile crowd perception. However, conventional federated learning frameworks strictly prohibit actual data sharing, which limits the knowledge of other system participants. As a result, part of the model accuracy has been sacrificed [4], [9].

In the context of mobile and edge intelligence systems, energy could become a bottleneck of system performance, especially when the system participants have heavy task load. Energy saving has become a concern [10]. Some open research questions which achieve energy efficiency through topology control have been identified in [11]. Energy efficiency of federated learning has been improved by reducing the CPU-cycle frequency of faster mobile devices in the training group [12]. In [13], a control algorithm has been proposed to determine the trade-off between local update and global parameter aggregation to minimize the loss function given a resource budget constraint. Trade-offs between communication and computational costs in federated learning over wireless networks have been optimized in [14].

In this work, we mainly model and analyze the trade-off between privacy-preserving federated learning and energy-aware resource allocation in a mobile and edge intelligence scenario, given constrained energy conditions. A market-oriented utility model and data trading strategies between data owner (i.e., FL worker) side and model owner (i.e., FL server) side are proposed.

978-1-7281-7236-1/20/$31.00 ©2020 IEEE

II. System Model

In this section, we propose a general system model for optimizing workload-aware federated learning in mobile edge intelligence systems, as shown in Fig. 1.

Suppose there are K different mobile and edge devices as FL workers in the system. FL workers are connected to an FL server, which intends to obtain data from the FL workers to train machine learning or deep learning models with the collected data. For the purpose of well trained models, the FL server prefers to obtain actual data. However, with privacy concerns, FL workers prefer to perform local training and upload the encrypted gradients to the FL server, instead of uploading actual data sets. Moreover, in mobile and edge systems, transferring a large amount of actual data requires reliable communication channels between each FL worker to the FL server.

Each FL worker, as a mobile and edge device, is equipped with a battery with the maximum capacity of E energy units. Both data training and other local computing tasks will consume a certain amount of energy units, e.g., data sensing. Charging facilities are provided to replenish the battery of the FL worker, e.g., by wireless energy charging. The FL worker also has a processing queue with the size Q for both training and local computing tasks. Only the tasks pushed into the queue can be processed because of the constrained computational capacity of the FL worker. As aforementioned, although with privacy concerns, local training is not always preferable by FL workers when local resource constraints exist. It is possible that there are not enough energy and computational capacity for local training the data.

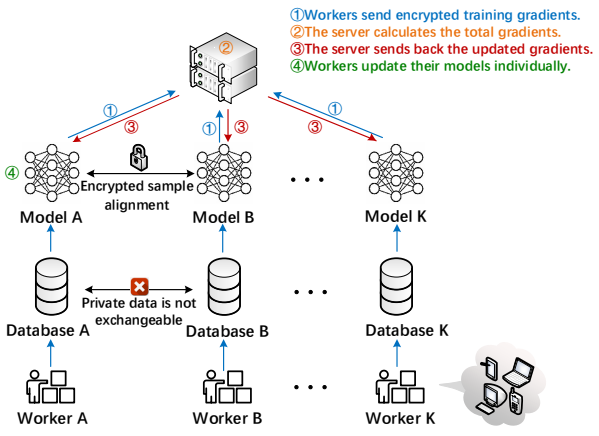


Fig. 1: General architecture for federated learning systems.

III. A Markov Decision Process based Optimization Problem Formulation

We optimize the decision of balancing between uploading actual data and performing federated learning to process data locally by modeling the decision making process of each FL worker as a Markov decision process.

A. State Space and Action Space

The state space of the FL worker is defined as follows:

$$\mathbb{S} = \{(\mathcal{Q}, \mathcal{E}, \mathcal{C}) | \mathcal{Q} \in \mathbb{Q}, \mathcal{E} \in \mathbb{E}, \mathcal{C} \in \mathbb{C}\} \quad (1)$$

where the state $\mathcal{S} \in \mathbb{S}$ is a compound state including three state components of the FL worker, i.e., a local data state $\mathcal{Q}$, an energy state $\mathcal{E}$, and a communication channel quality state $\mathcal{C}$, which are formally defined as follows:

- The local data state $\mathcal{Q} \in \mathbb{Q} = \{0, 1, \ldots, Q\}$ is denoted as the current amount of data stored in the FL worker. Q is the maximum data capacity of each FL worker. The FL worker regularly senses and generates data to be processed. We assume the data generating probability follows a Poisson distribution.
- The energy state $\mathcal{E} \in \mathbb{E} = \{0, 1, \ldots, E\}$ is the current battery energy level of the FL worker, where E is the maximum battery energy capacity. Without loss of generality, the external energy charging facilities will charge r energy units to the FL worker by the end of each time slot for decision making. If the current energy storage has already reached the maximum capacity, the charged energy will lost, and the energy level will not increase in this case.
- The communication channel quality state $\mathbb{C} = \{0, 1\}$ indicates the current channel state of the FL worker, regardless of the data state $\mathcal{Q}$ and energy state $\mathcal{E}$ of the FL worker. $\mathcal{C} = 0$ and 1 denote the poor and good qualities of the transmission channel between the FL worker and server, respectively. When the FL worker transmits actual data to the FL server, failure will occur if $\mathcal{C} = 0$.

The action space of the FL worker is defined as $\mathbb{A} = \{0, 1\}$, where the action $\mathcal{A} = 0$ denotes that the FL worker refuses to upload actual data. Local training will be performed, and encrypted gradient of the trained data will be uploaded to the FL server, i.e., federated learning. The FL worker consumes its own computing power in this case. Action $\mathcal{A} = 1$ denotes the situation that the FL worker prefers to upload actual data because of the potentially excessive amount of computational resources consumed, e.g., high data processing delay and energy outage.

B. State Transitions

The state of the FL worker changes from the current state $\mathcal{S}$ to the next state $\mathcal{S}'$ in each time slot when a decision $\mathcal{A}$ is made. The transition matrices of all the state components as well as the overall transition matrix are derived as follows:

1) Data State Transition: The process of data generating and processing can be demonstrated as a two-step process, as shown in Fig. 2. The FL worker firstly generates data, e.g., by sensing, and then decides the action of local training or data uploading. The probability that the FL worker generates k unit of data is a Poisson distribution with parameter λ, i.e., $\mathrm{P}_\lambda(k)$. However, the amount of data cannot exceed the capacity of the data capacity.

For ease of notation, we define a $Q \times 1$ matrix $\mathscr{P}_j$ as follows, $\forall j \in \{0, 1, \ldots, Q-1\}$:

$$\mathscr{P}_j = \left[\begin{array}{cccc} \mathbf{0}_{1\times j} & \mathsf{P}_\lambda(0) & \cdots & \mathsf{P}_\lambda(Q-j-1) \quad 1 - \sum_{k=0}^{Q-j-1} \mathsf{P}_\lambda(k) \end{array} \right] \tag{2}$$

and $\mathscr{P}_Q = [\ \mathbf{0}_{1\times(Q-1)} \quad 1\ ]$

When the local data training action $\mathcal{A} = 0$ is taken, the data state transition pattern can be shown in Fig. 2(a). After the training process, the amount of data successfully processed follows a Uniform distribution, i.e., $\mathsf{U}(0, i)$. The transition matrix of $\mathcal{Q}(\mathcal{A} = 0)$ in this case is denoted as follows:

$$\mathbf{Q}(\mathcal{A} = 0) = \left[\begin{array}{c} \boldsymbol{\phi}_0 \\ \vdots \\ \boldsymbol{\phi}_Q \end{array} \right] \tag{3}$$

where each line of the transition matrix $\boldsymbol{\phi}_i$ is defined as

$$\boldsymbol{\phi}_i = \frac{1}{i+1} \sum_{j=0}^{i} \mathscr{P}_j \tag{4}$$

Similarly, when the FL worker surrenders privacy by uploading actual data to the FL server, as shown in Fig. 2(b). As the FL worker has different local tasks to execute, the probability that the data uploading process is successfully scheduled by the FL worker processing unit is denoted as σ. The transition matrix $\mathcal{Q}(\mathcal{A} = 1)$ in this case is expressed as follows:

$$\mathbf{Q}(\mathcal{A} = 1) = \left[\begin{array}{c} \boldsymbol{\psi}_0 \\ \vdots \\ \boldsymbol{\psi}_Q \end{array} \right] \tag{5}$$

where each line $\boldsymbol{\psi}_j = \boldsymbol{\psi}_i = \sigma\mathscr{P}_0 + (1-\sigma)\mathscr{P}_i$

2) Energy State Transition: Mobile and edge devices are assumed to be supplied with energy charging facilities. In each time slot, the energy level will increase r units until fully charged. As both of the local training and actual data uploading processes consume energy, the transition matrix of the energy state $\mathcal{E}$ can be derived as follows:

$\mathbf{E}(\mathcal{A} = 0)$ is defined as the energy state transition matrix when the local federated learning is adopted by the FL worker, i.e., $\mathcal{A} = 0$. An amount of m units of energy will be consumed for each local training process. The training process fails if the current energy stored by the FL worker does not meet the minimum energy requirement of m energy units. No energy will be consumed in this case. The transition matrix is expressed as follows:

$$\mathbf{E}(\mathcal{A} = 0) = \left[\begin{array}{cc} \mathbf{0}_{(E-r+1)\times r} & \mathbf{I}_{(E-r+1)\times(E-r+1)} \\ \mathbf{0}_{(m+r-E-1)\times E} & \mathbf{1}_{(m+r-E-1)\times 1} \\ \mathbf{0}_{(E-r+1)\times r} & \mathbf{I}_{(E-r+1)\times(E-r+1)} \\ \mathbf{0}_{(r-m)\times E} & \mathbf{1}_{(r-m)\times 1} \end{array} \right] \tag{6}$$

where $\mathbf{I}$ is an identical matrix, and $\mathbf{1}$ is a matrix of ones.

When action $\mathcal{A} = 1$ is taken, i.e., the FL worker uploads actual data directly to the server for obtaining extra potential reward. An amount of n units of energy will be consumed for each uploading process. The transition matrix is expressed as follows:

$$\mathbf{E}(\mathcal{A} = 1) = \left[\begin{array}{cc} \mathbf{0}_{(E-r+1)\times r} & \mathbf{I}_{(E-r+1)\times(E-r+1)} \\ \mathbf{0}_{(n+r-E-1)\times E} & \mathbf{1}_{(n+r-E-1)\times 1} \\ \mathbf{0}_{(E-r+1)\times r} & \mathbf{\Gamma}_{(E-r+1)\times(E-r+1)} \\ \mathbf{0}_{(r-n)\times E} & \mathbf{1}_{(r-n)\times 1} \end{array} \right] \tag{7}$$

where

$$\mathbf{\Gamma} = \left[\begin{array}{ccc} \sigma & \cdots & 1-\sigma \\ & \ddots & \vdots \\ & & 1 \end{array} \right]$$

3) Channel State Transition: Communication channel quality affects the transmission of data from the FL worker to the corresponding FL server. As uploading actual data involves a relatively large amount of data, compared with uploading gradient values only, actual data could be lost during the uploading if the channel quality is not good enough. The transition matrix of the channel quality state $\mathcal{C}$ can be derived as follows:

When the FL worker uploads trained gradients, it is assumed that the FL server can be guaranteed to receive the gradients, as each gradient is merely a value. The transition matrix can be expressed as

$$\mathbf{C}(\mathcal{A} = 0) = \left[\begin{array}{cc} 0 & 1 \\ 0 & 1 \end{array} \right] \tag{8}$$

where $\mathcal{C} = 0$ and 1 indicates low and high channel quality situations, respectively.

When the FL worker directly uploads actual data to the FL server, the transition matrix can be expressed as follows:

$$\mathbf{C}(\mathcal{A} = 1) = \left[\begin{array}{cc} 1-\delta & \delta \\ 1-\delta & \delta \end{array} \right] \tag{9}$$

where $\delta \in [0, 1]$ indicating the probability that the uploading channel has high quality.

4) Overall State Transition: The overall state transition matrix of the FL worker is shown as follows:

$$\mathbf{W}(\mathcal{S}, \mathcal{S}'|\mathcal{A}) = \mathbf{Q}(\mathcal{A}) \bigotimes \mathbf{E}(\mathcal{A}) \bigotimes \mathbf{C}(\mathcal{A}) \tag{10}$$

where $\bigotimes$ is Kronecker product. With the overall state transition matrix $\mathbf{W}$ derived, the probability of all the possible transitions of the FL worker is obtained to formulate and solve the MDP.

C. Utility-based Model and Immediate Reward Function

After each action is taken in each decision making time slot, an immediate reward $R(\mathcal{Q}, \mathcal{E}, \mathcal{C}|\mathcal{A})$ will be incurred to the FL worker, which only represents the instant reward[1] generated in the current time slot, without considering any potential future rewards. We split the immediate reward $R(\mathcal{Q}, \mathcal{E}, \mathcal{C}|\mathcal{A})$ into three reward components, denoted as

$$R(\mathcal{Q}, \mathcal{E}, \mathcal{C}|\mathcal{A}) = \rho_M R_M(\mathcal{Q}|\mathcal{A}) + \rho_D R_D(\mathcal{Q}|\mathcal{A}) + \rho_P R_P(\mathcal{Q}|\mathcal{A}) \tag{11}$$

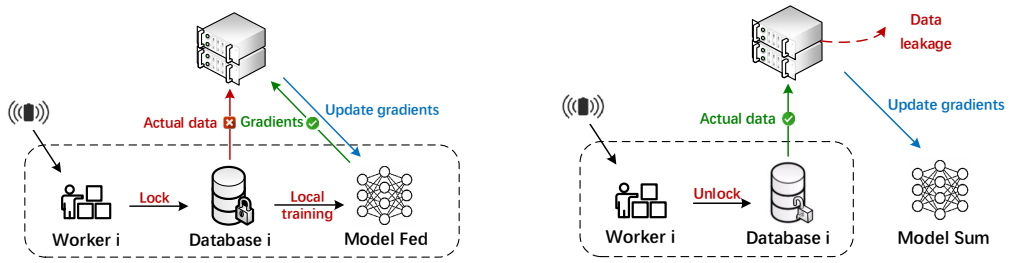
where ρ_M, ρ_D and ρ_P are weight factors.

In (11), $R_M(\mathcal{Q}|\mathcal{A})$ denotes the direct reward from data training or uploading by the FL worker, defined as follows:

$$R_M(\mathcal{Q}|\mathcal{A}) = \begin{cases} M(\triangle\mathcal{Q}), & (\mathcal{A} = 0 \text{ and } \mathcal{E} \geq m) \\ & \text{or} \\ & (\mathcal{A} = 1,\ \mathcal{E} \geq n \text{ and } \mathcal{C} = 1) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $M(\triangle\mathcal{Q}) \geq 0$ refers to the reward function of the final effect of the training model which is generally positively related to the amount of data used by the trained model, i.e.,

[1]We use reward and utility interchangeably in this work if there is no ambiguity. And reward can be negative, i.e., cost.



(a) (b)

Fig. 2: (a) Transition diagram of data states when $\mathcal{A}=0$ (i.e., local training), and (b) for the action $\mathcal{A}=1$ (i.e., actual data uploading).

$\triangle\mathcal{Q}$. Otherwise, if there is insufficient energy or low channel transmission quality, the worker cannot update information to the server. In this case, the reward will be 0.

$R_D(\mathcal{Q}|\mathcal{A})$ indicates delay caused by data packets stored in the FL worker, i.e., in the data cache queue, defined as follows:

$$R_D(\mathcal{Q}|\mathcal{A}) = D(\mathcal{Q}') \tag{13}$$

where $D(\mathcal{Q}') \leq 0$ is the cost function of data delay, which is negatively correlated with the data state at the end of the current time slot, i.e., $\mathcal{Q}'$.

$R_P(\mathcal{Q}|\mathcal{A})$ denotes the cost of privacy leakage, defined as follows:

$$R_P(\mathcal{Q}|\mathcal{A}) = \begin{cases} P(\triangle\mathcal{Q}), & (\mathcal{A}=1 \text{ and } \mathcal{E}\geq n \text{ and } \mathcal{C}=1) \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $P(\triangle\mathcal{Q}) \leq 0$ is the cost function of privacy leakage when the FL worker decides to upload actual data, which decreases as the amount of data used by the training model increases. As the amount of data increases, the amount of newly obtained information gradually decreases, so the amount of privacy leaked gradually tends to be flat and eventually $P(\triangle\mathcal{Q})$ converges to a certain value. The privacy cost is 0 by only uploading encrypted gradients when federated learning is employed.

IV. Deep Reinforcement Learning for Optimized Federated Learning

The goal of system optimization is to find an optimal decision strategy for each FL worker in response to any current observed system states, which maximizes the long-term expected reward of the FL worker. Conventionally, value iteration with Bellman equation [15] is employed to solve the MDP for obtaining optimal decisions, defined as follows:

$$V(\mathcal{S}) = \max_{\pi(\mathcal{A}|\mathcal{S})} Q(\mathcal{S}|\mathcal{A}) \tag{15}$$

$$Q(\mathcal{S}|\mathcal{A}) = R(\mathcal{S}|\mathcal{A}) + \gamma \sum_{\mathcal{S}'\in\mathbb{S}} P(\mathcal{S},\mathcal{S}'|\mathcal{A})Q(\mathcal{S}') \tag{16}$$

$$\pi^*(\mathcal{A}|\mathcal{S}) = \arg\max_{\pi(\mathcal{A}|\mathcal{S})} Q(\mathcal{S}|\mathcal{A}) \tag{17}$$

In (15)-(17), policy $\pi(\mathcal{A}|\mathcal{S})$ is defined as the action $\mathcal{A}$ taken in response to the current state $\mathcal{S}$ by the FL worker. Function $Q(\mathcal{S}|\mathcal{A})$ is the long-term reward considering both the immediate reward $R(\mathcal{S}|\mathcal{A})$ as well as all the discounted possible future reward, where $\gamma \in (0,1)$ is a discount factor.

Given the complete system and environmental information, the value iteration algorithm can calculate the optimal strategy of the FL worker. However, in general cases and application scenarios, system state space could be too large for the value iteration algorithm to solve. In this case, deep reinforcement learning based scheme can be applied. Deep reinforcement learning (DRL) is a combination of reinforcement learning (RL) and deep learning (DL) for solving optimal and intelligent decision making processes with large amount of system states and actions [16]. Compared with the value iteration algorithm, DRL significantly reduces the complexity of the model by predicting system state information learned from encountered and recorded historical experience.

Deep Q Network (DQN) is a typical implementation of DRL approaches, where a deep neural network (NN) is employed to approximate the long term expected reward, namely, Q-function. DQN takes the current observed system state as an input of the NN [17]. DQN outputs a set of Q values (or Q-action pairs) representing the corresponding values of all possible actions in the current state. With the NN approximator structure, the complexity of generating optimal decisions of the FL worker does not increase as the system scales up. As a result, the proposed MDP based system model can be solved by employing DQN, as shown in [18].

V. Numerical Results

A. System Parameters

The following parameters will be applied unless otherwise stated. The maximum data capacity of each FL worker is $Q=10$. The maximum energy capacity of each FL worker is $E=10$. Each local training decision consumes $m=3$ units of energy. Uploading actual data to the FL server side consumes $n=1$ unit of energy. The FL worker can replenish with $r=1$ unit energy during a time slot. The probability that the FL worker successfully schedules actual data uploading process is $\sigma=0.9$. The probability of high-quality channel transmission is $\delta=0.8$.

For immediate reward function in Section III-C, the training model reward is set as $M(\triangle\mathcal{Q})=\triangle\mathcal{Q}$, the data delay cost is set as $D(\mathcal{Q}')=-Q'$, and the privacy leakage cost is set as $P(\triangle\mathcal{Q})=-\log_2(\triangle\mathcal{Q}+1)$.

To evaluate the proposed MDP based scheme, four baseline schemes are implemented in the numerical results, as follows:

- RND: A random scheme that randomly makes decisions.
- GRE: A greedy scheme that only makes short-sighted decisions to maximize the immediate utility $R(\mathcal{Q}, \mathcal{E}, \mathcal{C}|\mathcal{A})$ of the current state.
- ALL-0: A local federated learning scheme that always only chooses to train data locally and upload the corresponding encrypted gradients to the server.
- ALL-1: A scheme that only uploads the actual data to the FL server.

B. Impacts of Maximum Data Capacity in FL Workers

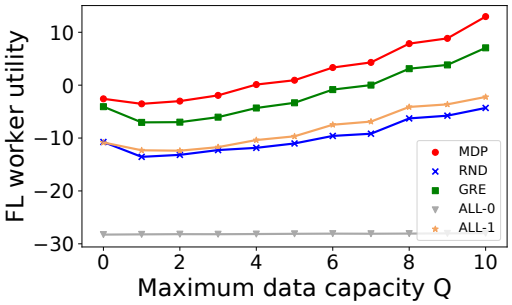


Fig. 3: Impacts of maximum data capacity Q to long-term expected utility.

We vary the maximum data capacity Q of the FL worker from 0 to 10. As shown in Fig. 3, the utilities of all schemes tend to increase as Q increases, since the increased data capacity enables FL workers to store more data for future training and rewards, despite possible delay caused by stored data. The MDP based scheme outperforms baseline schemes by considering long-term future utility and making far-sighted decision accordingly.

C. Impacts of Maximum Energy Capacity in FL Workers

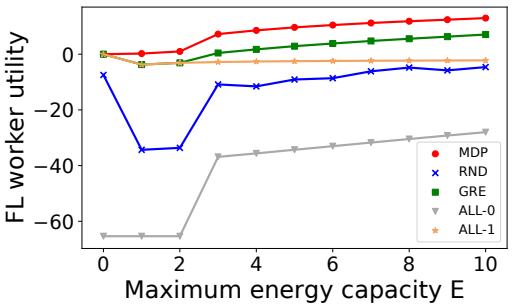
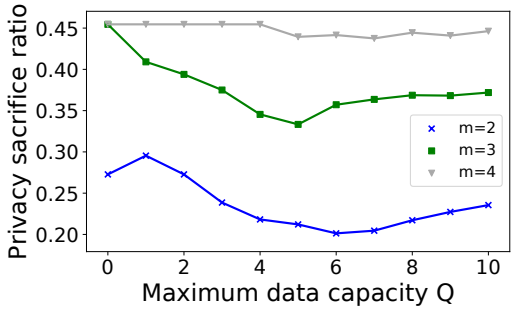


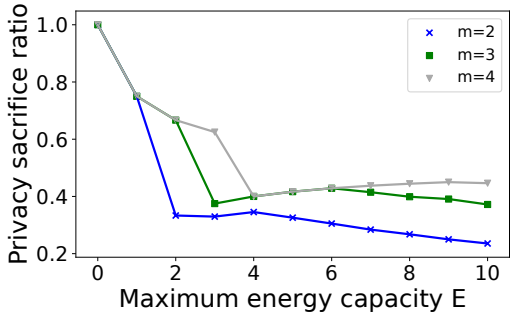
Fig. 4: Impacts of maximum energy capacity E to long-term expected utility.

We vary the maximum energy capacity E of the FL worker from 0 to 10. As shown in Fig. 4, when E is greater than 3, i.e., the required amount of energy units for local data training by the FL worker, the expected utility of the proposed MDP scheme tends to increase slightly, as the increase of energy capacity enables the FL worker to store more energy for future use, thereby reducing the probability of energy shortage. When the energy capacity far exceeds the energy required for local training, the marginal return of rewards caused by energy supply decreases.

D. Privacy Measurements



(a)



(b)

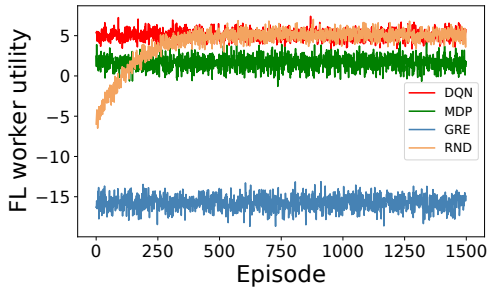
Fig. 5: Impacts of (a) maximum data capacity Q and (b) maximum energy capacity E to privacy sacrifice ratio.

Privacy sacrifice ratio measures the number of decisions to upload actual data out of all the decisions made by each FL worker in average. As shown in Fig. 5(a), as the maximum data capacity Q in each FL worker increases, privacy sacrifice ratio first decreases and then increases. The reason is that, when Q starts to increase, the FL worker tends to train the increased data locally, since FL workers are privacy leakage averse. However, as Q continues to increase, a significant impact of delay will be incurred to reduce the long term possible utility of the FL worker. In this case, the FL worker starts to upload data directly to the corresponding FL server to alleviate the delay. It is also demonstrated that when the number of data units contained in a batch of local training m increases, the FL worker also prefers to surrender data privacy to reduce extra cost caused.

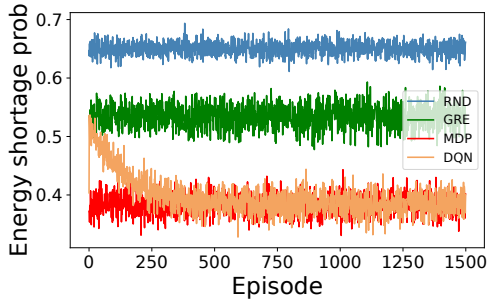
As shown in Fig. 5(b), when the maximum energy capacity E of the FL worker does not meet the requirement of energy consumption for local data training, i.e., $E < m$, privacy sacrifice ratio is extremely high, as the FL worker has to accumulate energy for several time slots first, and then train local data with the accumulated energy. At the boarder line cases when E increases to the level around m, the FL worker may still slightly prefers to drop privacy by uploading actual data since energy outage situations still occur in this case, which is more likely when the energy consumption of each

local training m is large, e.g., the $m = 4$ case. Afterwards, when E is much larger than m, the privacy sacrifice ratio decreases as the FL worker has abundant energy for local training to preserve data privacy.

E. Results for Deep Reinforcement Learning Solutions



(a)



(b)

Fig. 6: Simulation results of DRL and baseline schemes in terms of (a) expected utility and (b) energy shortage prob

Figure 6(a) compares the utilities of the FL worker solved by simulations including a DQN approach. After about 300 training episodes, the average utility obtained by DQN scheme converges and remains stable. As shown in Fig. 6(a), the simulation result of DQN converges to that of the MDP scheme solved by value iteration, which is a theoretically upper bound of the proposed long-term expected utility as the MDP value iteration solution has the information of the states and state transitions.

Energy shortage probability is demonstrated as in Fig. 6(b), which is defined as the probability that the battery stored energy units do not meet the energy consumption requirements when the FL worker makes decisions. The DQN scheme can well balance between energy consumption and model training, achieving an energy-aware federated learning strategy. The energy shortage probability solved by DQN gradually converges to the MDP scheme. The DQN scheme makes decision based on both long-term energy consumption and long-term utility by learning from historical experience replay, and quickly adjusts the decision strategy to avoid utility loss caused by energy outage.

VI. Conclusion

To tackle the energy consumption issues in federated learning for mobile and edge users, a Markov decision process based optimal decision model has been proposed in this work for mobile and edge users to balance between deploying federated learning to preserve privacy and surrender actual data to reduce energy consumption. Both conventional value iteration and deep reinforcement learning approaches have been applied to solve the proposed optimal decision making model. Numerical results have analyzed and compared the system performance metrics, and also demonstrated the attitude of each mobile and edge user towards data privacy in the system.

References

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, Jan. 2019.

[2] S. Han, W. K. Ng, L. Wan, and V. C. S. Lee, "Privacy-preserving gradient-descent methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 6, pp. 884–899, 2010.

[3] S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," in *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pp. 198–198, 2019.

[4] H. Li and T. Han, "An end-to-end encrypted neural network for gradient updates transmission in federated learning," in *2019 Data Compression Conference (DCC)*, pp. 589–589, 2019.

[5] X. Lu, Y. Liao, P. Lio, and P. Hui, "Privacy-preserving asynchronous federated learning mechanism for edge network computing," *IEEE Access*, vol. 8, pp. 48970–48981, 2020.

[6] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2019.

[7] K. Lin and W. Huang, "Using federated learning on malware classification," in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, pp. 585–589, 2020.

[8] W. Y. B. Lim, Z. Xiong, C. Miao, D. Niyato, Q. Yang, C. Leung, and H. V. Poor, "Hierarchical incentive mechanism design for federated machine learning in mobile networks," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[9] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, Z. Han, and C. S. Hong, "Incentivize to build: A crowdsourcing framework for federated learning," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019.

[10] M. Kaddari, M. El Mouden, A. Hajjaji, and A. Semlali, "Reducing energy consumption by energy management and energy audits in the pumping stations," in *2018 Renewable Energies, Power Systems Green Inclusive Economy (REPS-GIE)*, pp. 1–6, 2018.

[11] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich, "A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 121–144, 2013.

[12] Y. Zhan, P. Li, and S. Guo, "Experience-driven computational resource allocation of federated learning by deep reinforcement learning," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 234–243, 2020.

[13] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 63–71, 2018.

[14] 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 *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1387–1395, 2019.

[15] R. Bellman, "A markovian decision process," *Indiana University Mathematics Journal*, vol. 6, p. 15, 04 1957.

[16] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, *An Introduction to Deep Reinforcement Learning*. 2018.

[17] L.-J. Lin, "Reinforcement learning for robots using neural networks /," 01 1993.

[18] M. Roderick, J. MacGlashan, and S. Tellex, "Implementing the deep q-network," *CoRR*, vol. abs/1711.07478, 2017.