Sustainable Federated Learning with Mobile Crowdsensing: A DRL Approach for Learning Efficiency Maximization

Jiahui Zhao*†, Ming Chen*, Mai Le‖, Mengyan Huang†‡, Zhaohui Yang§¶, Quoc-Viet Pham†
*National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
†School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, D02 PN40, Ireland
‖Insight Centre for Data Analytics, University College Cork, Cork, Ireland.
‡State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China
§College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China
¶Zhejiang Provincial Key Laboratory of Info. Proc., Commun. & Netw. (IPCAN), Hangzhou, China
E-mails: {zhaojh, chenming}@seu.edu.cn, maile2108@gmail.com, my_huang@stu.xidian.edu.cn, yang_zhaohui@zju.edu.cn, viet.pham@tcd.ie

***Abstract*—In this paper, we propose a Sustainable Sensing Federated Learning (S2FL) system where Internet-of-Things (IoT) devices and mobile users harvest energy wirelessly to perform data sensing, local Federated Learning (FL) training, and model update transmissions. We formulate a joint optimization problem that considers transmission power, CPU frequency, bandwidth allocation, and time allocation to maximize long-term learning efficiency. To solve this complex and dynamic problem, we develop an algorithm that integrates Deep Reinforcement Learning (DRL) with optimization techniques, leveraging the Deep Deterministic Policy Gradient (DDPG) algorithm for time allocation and a Lagrangian-Based Block Coordinate Descent (BCD) Method for per-slot resource optimization. Simulation results demonstrate that our proposed DDPG-based DRL-S2FL algorithm significantly outperforms benchmark schemes, achieving up to 15% higher average reward compared to Deep Q-Networks (DQN) and 40% greater performance than Random strategies. This work highlights the effectiveness of combining advanced optimization techniques with deep reinforcement learning to enhance federated learning performance in dynamic wireless environments.**

***Index Terms*—Federated Learning, Mobile Crowdsensing, Energy Harvesting, Reinforcement Learning.**

I. Introduction

The surge in Internet-of-Things (IoT) services has led to massive data production, fueling advancements in deep learning but raising concerns over data privacy and network congestion due to centralized data processing. Federated Learning (FL) offers a solution by enabling collaborative training on local devices without data sharing, utilizing Mobile Edge Computing (MEC) to boost processing power and support applications like cybersecurity and smart cities. However, FL's deployment faces challenges related to device battery limits and data availability [1].

Advancements in hardware have equipped smart devices with enhanced sensing capabilities [2]. Mobile Crowdsensing (MCS) offers scalability, diverse data collection, and low deployment costs, making it essential for smart healthcare, entertainment, and smart vehicles. Conventional MCS systems require transmitting raw data to cloud or edge servers for AI processing, which again poses privacy and resource management challenges. Studies have begun integrating MCS with FL to develop privacy-preserving and scalable solutions [3], [4], yet high energy consumption and limited battery life of devices remain critical concerns. The work in [3] employs FL in Unmanned Aerial Vehicle (UAV)-enabled MCS systems with blockchain for secure model sharing. Crowd FL in [4] enhances privacy through secure aggregation and incentivizes user participation.

Wireless-Powered Communication (WPC) has become an essential technology for enhancing the sustainability and energy efficiency of FL and MCS systems [5]. By allowing devices to harvest energy wirelessly, WPC supports prolonged operations and enhances the feasibility of FL and MCS in resource-constrained environments. For example, [6] adopts wireless power transfer to power data transmission and local training of FL. In [7], energy harvesting sensors with rechargeable batteries are being deployed to promise cost-efficient and self-sustainable IoT networks. Despite these advancements, existing research often fails to address the complex interplay between sensing, computing, and communication capabilities. Specifically, the joint optimization of these components is essential to fully realize the benefits of WPC-enabled systems. Additionally, many studies employ one-shot optimization methods, which are unsuitable for dynamic network environments with time-varying channel conditions, energy availability, and user behavior. This necessitates the development of long-term optimization strategies that can adapt to changing network conditions and ensure sustained system performance.

To address these limitations, we introduce a Sustainable Sensing Federated Learning (S2FL) system, where users harvest energy to perform data sensing, local FL training, and data transmission. We formulate a joint optimization problem encompassing transmission power, CPU frequency, bandwidth allocation, and time allocation to maximize long-term learning efficiency. To solve this problem, we propose an algorithm that integrates Deep Reinforcement Learning (DRL) with optimization techniques to address the long-term learning efficiency maximization in the S2FL system. Our experiments demonstrate that our proposed Deep Deterministic Policy Gradient (DDPG)-based algorithm outperforms the benchmark schemes, validating the effectiveness of our S2FL approach.

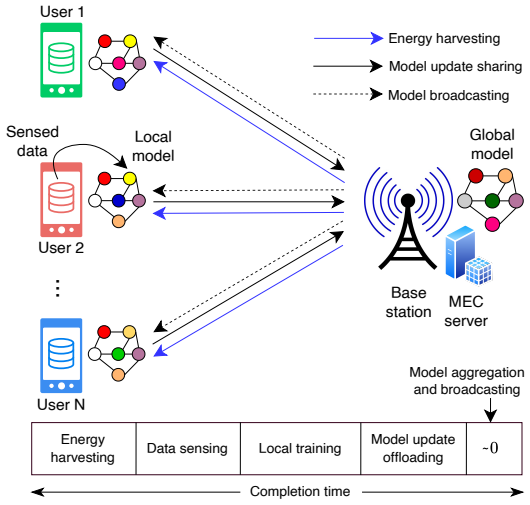

Fig. 1: Illustration of the S2FL.

II. System Model

We consider a wireless network comprising an edge server and N users, as shown in Fig. 1. The edge server performs model aggregation and provides wireless power transfer to the users. The set of users is denoted by $\mathcal{N} = \{1, \ldots, N\}$. In this work, we deploy the *harvesting-sensing-training-transmit* protocol with four phases, as illustrated in Fig. 1. In particular, in Phase I, the users need to harness external energy from the energy source. The harvested energy is used for sensing data from the surrounding environment in Phase II, training the local model in Phase III, and transmitting model updates in Phase IV.

To model S2FL when the channel conditions change dynamically, we discrete the S2FL system into T time slots with equal length, indexed as $\mathcal{T} = \{1, \ldots, t, \ldots, T\}$, with each slot further divided into four phases for each user n: energy harvesting ($\tau_n^h(t)$), data sensing ($\tau_n^s(t)$), local training ($\tau_n^l(t)$), and model transmission ($\tau_n^c(t)$), satisfying:

$$\tau_n^h(t) + \tau_n^s(t) + \tau_n^l(t) + \tau_n^c(t) \leq \tau_0, \forall n \tag{1}$$

which indicates that the completion time of a global communication round is limited to a time duration τ_0. Note that the time for model aggregation and model broadcasting is relatively short and treated as zero without loss of generality [8].

A. Energy Harvesting Model

A major concern in S2FL is the energy limitation of the users that may not have enough energy to complete the FL process, including data sensing from the surrounding environment, local training, and model update sharing with the edge server. During the first phase in S2FL, the edge server collocated with the radio tower can broadcast radio signals to deliver power to users in the downlink with transmit power P. This work leverages the linear energy harvesting model, and thus the amount of energy harvested by user n at time slot t is given by

$$E_n^h(t) = \varphi \tau_n^h(t) P h_n(t), \tag{2}$$

where φ represents the energy conversion efficiency. The term $h_n(t)$ denotes the channel gain between the user n and the edge server at time slot t, formulated as $h_n(t) = |h_n^s(t)|^2 h_n^l$, where h_n^l is the large-scale fading gain, and $h_n^s(t)$ represents the small-scale fading component. According to the Jakes' model [9], the small-scale fading $h_n^s(t)$ can be modelled as a first-order complex Gauss-Markov process $h_n^s(t+1) = \rho h_n^s(t) + \sigma_n(t+1)$, where $\rho \in [0, 1]$ is the correlation coefficient and $n_{ij}(t) \sim \mathcal{N}(0, 1-\rho^2)$. The harvested energy $E_n^h(t)$ is then stored in a finite battery or an energy buffer. Denote $E_n^r(t)$ as the energy of user n at the beginning of time slot t, which evolves as

$$E_n^r(t+1) = \max\left\{\min\left\{E_n^{\max}, E_n^r(t) + E_n^h(t)\right\} - E_n^{tot}(t), 0\right\} \tag{3}$$

where $E_n^{\max}$ is the maximum battery capacity. $E_n^{tot}(t)$ is calculated as the sum of energy consumed for data sensing, local training, and communication, i.e., $E_n^{tot}(t) = E_n^s(t) + E_n^l(t) + E_n^c(t)$, which will be detailed in the rest of this section.

B. Data Sensing

In the second phase, the users sense data from the surrounding environment. Suppose that at each time slot t user has a sensing rate of $r_n(t)$, the amount of sensing data $d_n(t)$ that each user can collect at time slot t is given by $d_n(t) = r_n(t)\tau_n^s(t)$. The corresponding energy consumption of user n at time slot t due to data sensing is given by

$$E_n^s(t) = q_n d_n(t) = q_n r_n(t) \tau_n^s(t), \tag{4}$$

where q_n is the energy consumption per unit of sensing data. Sharing sensing data with a central server compromises privacy and communication efficiency. FL addresses this by allowing users to keep data locally and perform training on their devices. The users then share only their model updates with the server for aggregation, thus enhancing data privacy while still utilizing the data in the learning process.

C. Local Computing Model

The sensing data collected by users are used for local training. Denote $I = (f_n, C_n, d_n)$ as the computing profile of user n, where f_n, C_n, d_n denote the local computing capability, the computation workload, and the sensing data, respectively. In S2FL, the users locally process all the sensing data and only share the model updates with the edge server for model aggregation, thus enhancing data privacy. Suppose that each user n trains its local learning model K_n times, the local training time of user n is

$$\tau_n^l(t) = K_n C_n d_n(t) / f_n(t) = K_n C_n r_n(t) \tau_n^s(t) / f_n(t). \tag{5}$$

The corresponding energy consumption is given as

$$E_n^l(t) = K_n \zeta_n C_n r_n(t) \tau_n^s(t) f_n^2(t), \tag{6}$$

where ζ_n is a coefficient dependent on the CPU architecture.

Similar to [10], users collaboratively build a learning model of size s (in bits), determined by parameters like network width and depth. Assume that each user should sense an amount

of D_0 in each communication round, which would mean the constraint $d_n(t) = r_n(t)\tau^s(t) \geq D_0$ should hold for all users.

D. Communication Model

In our considered S2FL network, we adopt frequency division multiple access (FDMA) to enable asynchronous transmissions [10]. Denote B as the system bandwidth and $b_n(t)$ as the bandwidth assigned to user n at time slot t. The achievable data rate of user n is calculated as

$$R_n(t) = b_n(t)\log_2\left(1 + \frac{h_n(t)p_n(t)}{n_0 b_n(t)}\right), \tag{7}$$

where $p_n(t)$ is the transmit power of user n at time slot t and n_0 is the noise power density. Given the transmission time $\tau^c(t)$ of n at time slot t, the constraint $R_n(t)\tau_n^c(t) \geq s$ should hold for all users to ensure that its model update is successfully transmitted to the server. The transmission energy consumption of user n for data transmission is $E_n^c(t) = p_n(t)\tau_n^c(t)$.

E. Problem Formulation

To investigate the sustainable design of S2FL, we formulate a problem to maximize learning efficiency while considering various constraints on resource allocation and network settings. From the convergence analysis of FL [11], we define learning efficiency of S2FL at time slot t as:

$$\nabla F(t) = \frac{\xi\sqrt{\sum_{n=1}^{N} r_n(t)\tau^s(t)}}{\nabla t(t)}, \tag{8}$$

where ξ is a coefficient that depends on specific settings of the deep model used for local training and $\nabla t(t) = \max_n \left\{\tau_n^h(t) + \tau_n^s(t) + \tau_n^l(t) + \tau_n^c(t)\right\}$ is the time duration of a global communication round.

This work aims to optimize learning efficiency by increasing the sensing data per round. Allowing more time for sensing can reduce the duration available for energy harvesting, potentially leading to insufficient energy for local training and model updates. Conversely, users can store harvested energy for future use when data collection is low, trading off immediate learning efficiency for long-term gains. Therefore, a long-term strategy is essential for maximizing efficiency in S2FL-enabled networks. The long-term learning efficiency optimization problem can be mathematically formulated as:

$$\max_{\boldsymbol{p},\boldsymbol{f},\boldsymbol{b},\boldsymbol{\tau}} \quad \frac{1}{T}\sum\nolimits_{t=1}^{T} \nabla F(t), \tag{9a}$$

$$\text{s.t.} \quad (1), (3) \tag{9b}$$

$$\tau_n^l(t) \geq K_n C_n r_n(t)\tau_n^s(t)/f_n(t), \forall n, t, \tag{9c}$$

$$r_n(t)\tau_n^s(t) \geq D_0, \forall n, t, \tag{9d}$$

$$\tau_n^c(t)R_n(t) \geq s, \forall n, t, \tag{9e}$$

$$0 \leq p_n(t) \leq p_n^{\max}, \forall n, t, \tag{9f}$$

$$\sum_{n\in\mathcal{N}} b_n(t) \leq B, b_n(t) \geq 0, \forall n, t, \tag{9g}$$

$$f_n^{\min} \leq f_n(t) \leq f_n^{\max}, \forall n, t, \tag{9h}$$

where $\boldsymbol{p} \triangleq \{p_n(t)\}_{\forall n}$, $\boldsymbol{f} \triangleq \{f_n(t)\}_{\forall n}$, $\boldsymbol{b} \triangleq \{b_n(t)\}_{\forall n}$, and $\boldsymbol{\tau} \triangleq \{\tau_n^h(t), \tau_n^s(t), \tau_n^l(t), \tau_n^c(t)\}_{\forall n}$ respectively, B is the system bandwidth, $p_n^{\max}$ is the maximum transmit power, and $f_n^{\min}$ ($f_n^{\max}$) is the minimum (maximum) CPU frequency.

Solving the optimization problem (9) faces several challenges. The problem is highly non-convex. Additionally, traditional one-shot optimization methods are not suitable for addressing the dynamic nature of constraints, such as users' fluctuating energy availability. To address these issues, we propose an efficient algorithm that integrates DRL with optimization in the next section.

III. Proposed Algorithm

We propose an algorithm that integrates DRL with optimization techniques to maximize long-term learning efficiency in the S2FL system. The algorithm operates in two main stages:

- **DRL for Time Allocation:** A DRL agent learns the optimal policy for allocating energy harvesting time ($\tau_n^h(t)$) across the users and time slots, influencing the available time for sensing, training, and communication.
- **Per-Slot Optimization for Resource Allocation:** Given the energy harvesting time from the DRL agent, an optimization subroutine determines the optimal CPU frequency ($f_n(t)$), transmission power ($p_n(t)$), and bandwidth ($b_n(t)$) for each user.

A. Problem Reformulation

We simplify the energy evolution constraint (3) into:

$$E_n^{\text{tot}}(t) \leq E_n^r(t) + E_n^h(t) \leq E_n^{\max}, \tag{10}$$

$$E_n^r(t+1) = E_n^r(t) + E_n^h(t) - E_n^{\text{tot}}(t). \tag{11}$$

We also convert constraints (1), (9c), and (9e) into equalities:

$$\tau_n^h(t) + \tau_n^s(t) + \tau_n^l(t) + \tau_n^c(t) = \tau_0, \tag{12}$$

$$\tau_n^l(t) = K_n C_n r_n(t)\tau_n^s(t)/f_n(t), \tag{13}$$

$$\tau_n^c(t) = s/R_n(t), \tag{14}$$

ensuring efficient utilization of the available time τ_0. With these simplifications, the optimization problem becomes:

$$\max_{\boldsymbol{p},\boldsymbol{f},\boldsymbol{b},\boldsymbol{\tau}} \quad \frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{n=1}^{N} r_n(t)\tau_n^s(t)} \tag{15a}$$

$$\text{s.t.} \quad (9\text{d}), (9\text{f}) - (9\text{h}), (10) - (14), \tag{15b}$$

B. Markov Decision Process (MDP) Formulation

To effectively apply DRL, we model the long-term problem as an MDP.

1) State Space: The state at each time slot t, $\boldsymbol{s}_t$, includes:

$$\boldsymbol{s}_t = \{\boldsymbol{h}(t), \boldsymbol{E}^r(t), \boldsymbol{r}(t)\}, \tag{16}$$

where $\boldsymbol{h}(t)$ is the channel gain vector, $\boldsymbol{E}^r(t)$ is the current energy vector, and $\boldsymbol{r}(t)$ is the sensing rate vector.

2) Action Space: The action at each time slot t, $\boldsymbol{a}_t$, consists of the energy harvesting time allocations for all users as $\boldsymbol{a}_t = \{\tau_n^h(t), \forall n\}$. According to constraint (10), we obtain

$$0 \leq \tau_n^h(t) \leq \min\{\frac{E_n^{\max} - E_n^r(t)}{\varphi P h_n(t)}, \tau_0\} \tag{17}$$

3) Reward Function: The immediate reward is defined as:

$$r_t = \begin{cases} S_f\sqrt{\sum_{n=1}^N w_n(t)}, & \text{if } \sum_{n=1}^N w_n(t) \geq 0, \\ -S_f\sqrt{-\sum_{n=1}^N w_n(t)}, & \text{otherwise,} \end{cases} \tag{18}$$

where $w_n(t) = r_n(t)\tau_n^s(t) - P_f$, and P_f is a penalty for not meeting the minimum data requirement, which is defined as

$$P_f = \begin{cases} 0, & \text{if } \tau_n^s(t) \geq \frac{D_0}{r_n(t)}, \\ P_f, & \text{otherwise.} \end{cases} \tag{19}$$

This reward structure incentivizes maximizing the sensed data while penalizing insufficient data collection.

C. Resource Allocation Optimization

Given the energy harvesting time from the DRL agent, we optimize CPU frequency ($f_n(t)$), transmission power ($p_n(t)$), and bandwidth ($b_n(t)$) via an optimization subroutine.

1) Formulate the Per-Slot Optimization Problem: According to (12)- (14), we have

$$\tau_n^s(t) = \frac{\tau_0 - \tau_n^h(t) - \frac{s}{R_n(t)}}{\frac{K_n C_n r_n(t)}{f_n(t)} + 1}. \tag{20}$$

Then, the left-side constraint of (10) can be rewritten as

$$G_n(f_n(t))\left(\tau_0 - \tau_n^h(t) - \frac{s}{R_n(t)}\right) + p_n(t)\frac{s}{R_n(t)} \leq E_n^r(t) + E_n^h(t), \tag{21}$$

where

$$G_n(f_n(t)) = \frac{\left(q_n r_n(t) + K_n\zeta_n C_n r_n(t) f_n^2(t)\right)}{\frac{K_n C_n r_n(t)}{f_n(t)} + 1}. \tag{22}$$

The per-slot optimization problem is formulated as

$$\max_{\boldsymbol{p},\boldsymbol{f},\boldsymbol{b}} \quad \sum_{n=1}^N r_n(t)\frac{\tau_0 - \tau_n^h(t) - \frac{s}{R_n(t)}}{\frac{K_n C_n r_n(t)}{f_n(t)} + 1}, \tag{23a}$$

$$\text{s.t.} \quad (9f), (9g), (9h), (21), \tag{23b}$$

$$r_n(t)\frac{\tau_0 - \tau_n^h(t) - \frac{s}{R_n(t)}}{\frac{K_n C_n r_n(t)}{f_n(t)} + 1} \geq D_0, \forall n. \tag{23c}$$

2) Solve the Per-Slot Optimization Problem: Problem (23) is non-convex due to variable coupling. We employ the Lagrangian-Based Block Coordinate Descent (BCD) method to decouple and solve it iteratively. We first form the partial Lagrangian with multiplier $\alpha \geq 0$ as follows:

$$\max_{\boldsymbol{p},\boldsymbol{f},\boldsymbol{b}} \quad \sum_{n=1}^N r_n(t)\frac{\tau_0 - \tau_n^h(t) - \frac{s}{R_n(t)}}{\frac{K_n C_n r_n(t)}{f_n(t)} + 1} - \alpha\left(\sum_{n=1}^N b_n(t) - B\right), \tag{24a}$$

$$\text{s.t.} \quad (9f), (9h), (21), (23c), \tag{24b}$$

$$b_n(t) \geq 0, \forall n, \tag{24c}$$

This decouples the problem for each user n:

$$\max_{\substack{f_n(t), p_n(t), \\ b_n(t)}} \quad r_n(t)\frac{\tau_0 - \tau_n^h(t) - \frac{s}{R_n(t)}}{\frac{K_n C_n r_n(t)}{f_n(t)} + 1} - \alpha b_n(t), \tag{25a}$$

$$\text{s.t.} \quad (9f), (9h), (21), (23c), \tag{25b}$$

$$b_n(t) \geq 0. \tag{25c}$$

We solve the following subproblems iteratively:

a. Optimizing the CPU frequency $f_n(t)$:

$$\max_{f_n(t)} \quad f_n(t) \tag{26a}$$

$$\text{s.t.} \quad K_n\zeta_n C_n r_n(t) f_n^2(t) - \frac{F_n(t) K_n C_n r_n(t)}{f_n(t)} \leq F_n(t) - q_n r_n(t), \tag{26b}$$

$$f_n(t) \geq \frac{K_n C_n r_n(t)}{\frac{\tau_0 - \tau_n^h(t) - \frac{s}{R_n(t)}}{\frac{D_0}{r_n(t)}} - 1}, \tag{26c}$$

$$f_n^{\min} \leq f_n(t) \leq f_n^{\max}, \tag{26d}$$

where

$$F_n(t) = \frac{E_n^r(t) + E_n^h(t) - p_n(t)\frac{s}{R_n(t)}}{\tau_0 - \tau_n^h(t) - \frac{s}{R_n(t)}}. \tag{27}$$

The constraint (26b) is monotonically increasing with $f_n(t)$. We perform a bisection search to find the root $f_n^{(0)}(t)$ satisfying equality. Then, the closed-form solution of the problem (26) is $f_n^*(t) = \min\{f_n^{(0)}(t), f_n^{\max}\}$.

b. Optimizing the transmit power $p_n(t)$:

$$\max_{p_n(t)} \quad R_n(t) \tag{28a}$$

$$\text{s.t.} \quad \frac{p_n(t)s}{R_n(t)} - \frac{G_n(f_n(t))s}{R_n(t)} \leq E_n^r(t) + E_n^h(t) - G_n(f_n(t))\left(\tau_0 - \tau_n^h(t)\right), \tag{28b}$$

$$R_n(t) \geq \frac{s}{\tau_0 - \tau_n^h(t) - \frac{D_0}{r_n(t)}\left(\frac{K_n C_n r_n(t)}{f_n(t)} + 1\right)}, \tag{28c}$$

$$0 \leq p_n(t) \leq p_n^{\max}. \tag{28d}$$

We analyze the function $f(x) = \frac{x}{\log_2(1+x)}$ for $x \geq 0$. Its first derivative is $f'(x) = \frac{\log_2(1+x) - \frac{x}{\ln 2\cdot(1+x)}}{[\log_2(1+x)]^2}$. Let the numerator be $g(x) = \log_2(1+x) - \frac{x}{\ln(2)(1+x)}$. The derivative of $g(x)$ is $g'(x) = \frac{x}{\ln(2)\cdot(1+x)^2} \geq 0$. Since $g'(x) \geq 0$, $g(x)$ is monotonically increasing. Therefore, $g(x) \geq g(0) = 0$, implying that $f'(x) \geq 0$. Consequently, $f(x)$ is monotonically increasing with respect to x. Hence, the left hand side of (28b) is monotonically increasing with $p_n(t)$, allowing the use of a bisection search to find $p_n^{(0)}(t)$. The optimal transmission power is then determined by $p_n^*(t) = \min\{p_n^{(0)}(t), p_n^{\max}\}$.

Algorithm 1 Per-Slot Optimization for S2FL

Input: Current state $\boldsymbol{s}_t$, action $\boldsymbol{a}_t$,
Initialize: α_{lower}, α_{upper}, tolerance `tol`
while True **do**
  $\alpha \leftarrow (\alpha_{\text{lower}} + \alpha_{\text{upper}})/2$
  for each user n **do**
    Iteratively solve (26), (28), (29) until the objective of (25) converges to obtain $f_n(t)$, $p_n(t)$, $b_n(t)$
  end for
  Adjust α**:** $\alpha_{\text{update}} \leftarrow \sum b_n(t) \leq B \ ? \ \alpha_{\text{upper}} : \alpha_{\text{lower}}$
  Converge: **break** if $|\sum b_n(t) - B| <$ `tol`
end while
Output: $\{f_n^*(t), p_n^*(t), b_n^*(t)\}_{\forall n}$

c. Optimizing the bandwidth $b_n(t)$:

$$\min_{b_n(t)} \quad \frac{r_n(t)s}{\frac{K_n C_n r_n(t)}{f_n(t)} + 1} \frac{1}{R_n(t)} + \alpha b_n(t) \tag{29a}$$

$$\text{s.t.} \quad R_n(t) \geq \frac{p_n(t)s - G_n(f_n(t))s}{E_n^r(t) + E_n^h(t) - G_n(f_n(t))\left(\tau_0 - \tau_n^h(t)\right)}, \tag{29b}$$

$$R_n(t) \geq \frac{s}{\tau_0 - \tau_n^h(t) - \frac{D_0\left(\frac{K_n C_n r_n(t)}{f_n(t)} + 1\right)}{r_n(t)}}, \tag{29c}$$

$$b_n(t) \geq 0, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}, \tag{29d}$$

Lemma 1: Problem (29) is convex.

Proof: Define the function $h(x) = \log_2(1 + x)$ for $x \geq 0$, which is concave. The perspective function $h(t) = t \log_2\left(1 + \frac{1}{t}\right)$ for $t > 0$ is also concave. Since $h(t) \geq 0$, the reciprocal function $\frac{1}{h(t)}$ is convex. Therefore, the objective function in (29) is convex. Additionally, the constraints are linear or convex in $b_n(t)$. Hence, the entire optimization problem (29) is convex.

Given the convexity of problem (29) as established in Lemma 1, we can efficiently solve it using standard convex optimization techniques.

The steps involved in solving the per-slot optimization problem are detailed in Algorithm 1. The algorithm employs a Lagrange multiplier α to enforce the bandwidth constraint $\sum_{n=1}^{N} b_n(t) \leq B$. It begins by initializing α within predefined bounds and utilizes a bisection search to iteratively adjust its value to satisfy the bandwidth constraint. In each iteration, for every user n, the algorithm iteratively optimizes $f_n(t)$, $p_n(t)$ and $b_n(t)$ until convergence based on the BCD method. After updating the resource allocations for all users, the total allocated bandwidth is calculated. If the total allocated bandwidth falls within a specified tolerance `tol` of the available bandwidth B, the algorithm converges and terminates. Otherwise, it adjusts the bounds of α and repeats the optimization process.

D. Training the DRL Agent

To effectively learn the optimal policy for energy harvesting time, we employ the DDPG algorithm. Algorithm 2 presents the pseudocode of the proposed DDPG-based learning approach for the S2FL system. The algorithm initializes the actor and critic networks along with their corresponding target networks and a replay buffer to store experience tuples. During each training episode, the agent observes the current state, selects an action, and executes resource allocation using Algorithm 1. It then receives a reward, observes the next state, and stores the transition in the replay buffer. Periodically, the agent samples mini-batches from the replay buffer to update the critic by minimizing the loss between predicted and target Q-values and to update the actor policy by following the gradient of the expected return. The target networks are softly updated to ensure stable learning. Through this iterative process, the DDPG agent progressively refines its policy, enabling efficient and adaptive energy harvesting decisions that enhance the overall learning efficiency of the S2FL system.

Algorithm 2 DDPG-Based Learning for S2FL

Initialize Actor $\mu(\boldsymbol{s}|\theta^\mu)$, Critic $Q(\boldsymbol{s}, \boldsymbol{a}|\theta^Q)$, Targets $\mu' \leftarrow \mu$, $Q' \leftarrow Q$, Replay buffer $\mathcal{D}$, Noise $\mathcal{N}$.
for each episode **do**
  Reset environment, get initial state $\boldsymbol{s}_1$
  for each time slot t **do**
    Select $\boldsymbol{a}_t = \mu(\boldsymbol{s}_t|\theta^\mu) + \mathcal{N}$, execute Algorithm 1, get $\boldsymbol{s}_{t+1}$, r_t.
    Store $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ in the replay buffer.
    Sample m from $\mathcal{D}$, compute targets for each $(\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_{i+1})$: $y_i \leftarrow r_i + \gamma Q'(\boldsymbol{s}_{i+1}, \mu'(\boldsymbol{s}_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update Critic: $L = \frac{1}{m}\sum(Q(\boldsymbol{s}_i, \boldsymbol{a}_i|\theta^Q) - y_i)^2$
    Update Actor: $\nabla_{\theta^\mu} J \approx \frac{1}{m}\sum \nabla_{\boldsymbol{a}} Q(\boldsymbol{s}_i, \boldsymbol{a}_i = \mu(\boldsymbol{s}_i|\theta^\mu)|\theta^Q) \nabla_{\theta^\mu} \mu(\boldsymbol{s}_i|\theta^\mu)$
    Update Targets: $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$, $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
  end for
end for

IV. Simulation Results

In this section, we provide simulation results to evaluate our proposed DRL-S2FL algorithm and compare it with the benchmarks. We simulate an S2FL network and set the number of users to be $N = 3$. We set the transmit power of the edge server to be $P = 25$ W, the maximum transmit power of users to be $p_n^{\max} = 10$ dBm, the system bandwidth to be $B = 5$ MHz, the noise power density to be $n_0 = -174$ dBm, and the maximum and minimum CPU frequency to be $f_n^{\max} = 1.0$ GHz and $f_n^{\min} = 0.1$ GHz, respectively. We set $\varphi = 0.9$, $E_n^{\max} = 10$ mJ for all users. The coefficient of the CPU is $\xi = 10^{-28}$ for all users, and the model size is $s = 28.1$ Kb. In DDPG, the experience replay memory can store up to 10^4 samples, and the mini-batch size is $M = 128$. The learning rate is set to be $\alpha = 0.001$ for both the actor and critic networks. We set the discount factor to be $\gamma = 0.9$. For comparison, we choose two benchmarks including 1) **DQN:** the continuous action is discretized into 10 levels, each of which is considered as a potential action in DQN; 2) **Random:** the agent randomly selects an action every time.

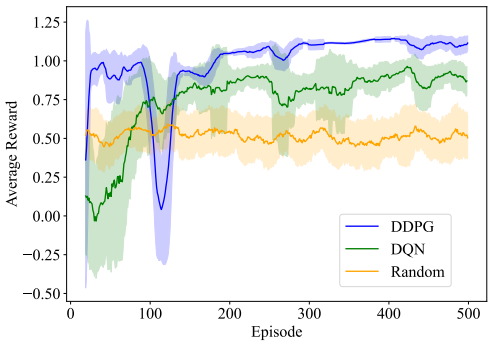

Fig. 2: Convergence of the proposed DRL-S2FL algorithm.

Fig. 2 shows the convergence of the DRL-S2FL algorithm based on DDPG, DQN and Random. The curves are smoothed using a moving average with a 20-slot window. Both DDPG and DQN exhibit dynamic convergence, with DDPG outperforming DQN because DQN's requirement for action space discretization leads to a loss of precision and less effective exploration, while DDPG's continuous action handling and actor-critic architecture allow for more refined decision-making and adaptive learning. The Random line remains relatively flat, as random actions do not leverage past learnings to enhance performance.

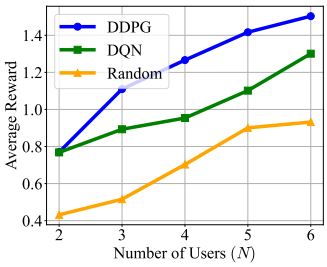

Fig. 3: Impact of N

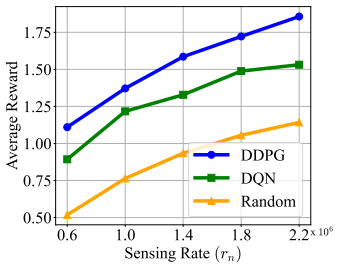

Fig. 4: Impact of sensing rate

Fig. 3 illustrates the impact of the number of users. As N increases, the average reward consistently rises, which can be attributed to more data sensed and subsequently utilized for federated learning. Both DDPG and DQN consistently outperform the Random strategy across all user scenarios. Notably, at $N = 2$, the performance of DQN closely matches that of DDPG. This similarity is expected in simpler, low-dimensional settings where the action space can be effectively discretized and managed by DQN's value-based approach. However, as the number of users increases, DDPG exhibits superior performance compared to DQN. This divergence can be attributed to DDPG's inherent ability to handle high-dimensional continuous action spaces more efficiently.

Fig. 4 illustrates the impact of the sensing rate. Here, we assume an equal sensing rate for all users at all time. As the sensing rate increases, the average reward also increases across all strategies. This positive correlation is intuitive, as higher sensing rates result in more data being collected and utilized for federated learning, thereby improving the overall learning efficiency. Among the evaluated strategies, DDPG achieves about 15% higher average reward than DQN and 40% greater than the Random strategy. The enhanced performance of DDPG demonstrates its robustness and efficiency in dynamically adjusting to varying sensing rates.

V. Conclusion

In this paper, we presented a novel approach to optimize long-term learning efficiency in S2FL systems. By integrating DRL with optimization techniques, we addressed the challenges posed by dynamic wireless network environments and resource constraints. Our proposed algorithm leverages the DDPG method to learn optimal energy harvesting time allocations while employing a Lagrangian-based BCD method for efficient per-slot resource allocation. Simulation results show that the DDPG-based algorithm consistently outperforms DQN and random action selection, validating the effectiveness of our approach.

Acknowledgments

The work is supported in part by the key research and development plan projects of Jiangsu Province under BE2022316. The work of Quoc-Viet Pham is supported in part by the European Union under the ENSURE-6G project (Grant Agreement No. 101182933).

References

[1] M. Le, T. Huynh-The, T. Do-Duy, T.-H. Vu, W.-J. Hwang, and Q.-V. Pham, "Applications of distributed machine learning for the internet-of-things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, Jul. 2024.

[2] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, "Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 9–39, Jan. 2023.

[3] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for UAV-assisted crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1055–1069, Apr.-Jun. 2021.

[4] B. Zhao, X. Liu, W.-N. Chen, and R. Deng, "CrowdFL: Privacy-preserving mobile crowdsensing system via federated learning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4607–4619, Aug. 2023.

[5] J. Wang, H. Zhou, L. Zhao, D. Meng, and X. Shouzhi, "Joint optimization of charging time and resource allocation in wireless power transfer assisted federated learning," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2024, pp. 1–6.

[6] Y. Wu, Y. Song, T. Wang, L. Qian, and T. Q. S. Quek, "Non-orthogonal multiple access assisted federated learning via wireless power transfer: A cost-efficient approach," *IEEE Transactions on Communications*, vol. 70, no. 4, pp. 2853–2869, Apr. 2022.

[7] E. Delfani and N. Pappas, "Version age-optimal cached status updates in a gossiping network with energy harvesting sensor," *IEEE Transactions on Communications*, pp. 1–1, 2024.

[8] Q.-V. Pham, M. Le, T. Huynh-The, Z. Han, and W.-J. Hwang, "Energy-efficient federated learning over UAV-enabled wireless powered communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4977–4990, May 2022.

[9] J. Zhao, M. Chen, Y. Pan, H. Sun, Y. Cang, and J. Wang, "Energy minimization of the cell-free mec networks with two-timescale resource allocation," *IEEE Transactions on Wireless Communications*, vol. 23, no. 12, pp. 18 623–18 636, Dec. 2024.

[10] 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, Mar. 2021.

[11] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, p. 661–670.