Sumedh Jitendra Dongare, Andrea Ortiz, and Anja Klein, ”Federated Deep Reinforcement Learning for Task Participation in Mobile Crowdsensing”, in *Proceedings of the IEEE Global Communications Conference - (IEEE GLOBECOM 2023)*, December 2023.

# Federated Deep Reinforcement Learning for Task Participation in Mobile Crowdsensing

Sumedh Dongare, Andrea Ortiz, Anja Klein
Communications Engineering Lab, Technical University of Darmstadt, Germany.
{s.dongare, a.ortiz, a.klein}@nt.tu-darmstadt.de

***Abstract*—Mobile Crowdsensing (MCS) is a promising distributed sensing architecture that harnesses the power of sensors on mobile units (MUs) to perform sensing tasks. The MCS is a dynamic system in which the requirements of the sensing tasks, the MUs' conditions and the available resources change over time. The performance of an MCS system depends on the selection of the MUs participating in each sensing task. However, this is not a trivial problem. An optimal task participation strategy requires non-causal knowledge about the dynamic MCS system, a requirement that cannot be fulfilled in real implementations. Moreover, centralized optimization-based approaches do not scale with increasing number of participating MUs and often ignore the MUs' preferences. To overcome these challenges, in this paper we propose a novel multi-agent federated deep reinforcement learning algorithm (FDRL-PPO) which does not need this perfect non-causal knowledge, but instead, enables the MUs to learn their own task participation strategies based on their own conditions, available resources, and preferences. Through federated learning, the MUs share their learned strategies without disclosing sensitive information, enabling a robust and scalable task participation scheme. Numerical evaluations validate the effectiveness and efficiency of FDRL-PPO in comparison with reference schemes.**

## I. Introduction

In recent years, a novel sensing architecture called Mobile Crowdsensing (MCS) has emerged in the field of distributed sensing. MCS employs sensors installed on smart devices (e.g. smartphones, wearables, and smart vehicles) to perform sensing tasks and utilizes the 'wisdom of the crowd' [1]. In comparison with traditional wireless sensor networks (WSNs), MCS offers many advantages such as higher coverage due to the mobility of mobile units (MUs), lower infrastructural costs and larger availability of MUs in a given area. As a result, MCS has become a topic of interest in the research community [2], [3]. Many applications such as traffic monitoring [4], [5], environmental monitoring [6], spectrum sensing [7], mHealth [8] and crowd-sourcing [9] use MCS for distributed sensing.

An MCS scenario commonly consists of three entities, namely, data requesters, an MCS platform (MCSP) and MUs. Data requesters require sensing data and communicate this request to an MCSP. The request includes different requirements of the sensing result (e.g., maximum size of the sensing result in bits) along with a predefined monetary budget for this request. The monetary budget is usually directly proportional to the strictness of the task requirements. The MCSP reformulates the request as a sensing task and sequentially broadcasts such tasks along with their requirements to its associated MUs. Based on their individual preferences, the MUs decide whether they want to convey their willingness to perform the task or not (i.e., task proposal). A task proposal consists of the MU's desired payment as a reward for their task efforts in terms of energy spent if they perform the current sensing task successfully. Multiple MUs may send their proposals to perform one sensing task. The MUs are typically battery operated and require a charging mechanism to replenish their batteries. In this work, we assume energy harvesting (EH) MUs that form a sustainable MCS architecture [10], [11]. Due to the budgeted nature of the problem, the MCSP selects the cheapest MUs within its task budget, and signals them this decision such that they can perform the complete sensing task independently. In an ideal case, the MCSP needs only one MU to perform the sensing task. However, to ensure task completion, the MCSP may select multiple MUs within the task budget for redundancy. Only those MUs which complete the task successfully according to the requirements are rewarded with their desired payment.

In a budgeted MCS scenario, the MUs' task participation strategy greatly affects the performance of the system and the MUs' individual payments. For example, an MU participating in a task whose requirements it cannot meet, causes unnecessary energy expenditure without any payment. There could also be an MU which can potentially complete the task but is not selected by the MCSP due to budget restrictions. Thus, to make an optimal decision, the MUs theoretically require non-causal knowledge about the tasks, as well as the battery and channel conditions of all MUs in the MCS system.

In the literature, many works assume the availability of such knowledge at the MCSP and optimize the task allocation decision [12]–[14]. However, having such non-causal knowledge is impractical and, due to the complexity of the problem, these solutions are non-scalable. Nevertheless, such approaches provide an upper bound on the system performance. To achieve scalability, some works propose game theoretic approaches that also consider user preferences for the task allocation problems [15], [16]. However, these works assume static problem formulations and cannot adapt to dynamic MCS scenarios.

This work has been funded by the German Research Foundation (DFG) as a part of the projects C1 and T2 within the Collaborative Research Center (CRC) 1053 - MAKI (Nr. 210487104) and has been supported by the BMBF project Open6GHub (Nr. 16KISK014) and the LOEWE Center EmergenCITY. This work is also supported by DAAD with funds from the Federal Ministry of Education and Research (BMBF).

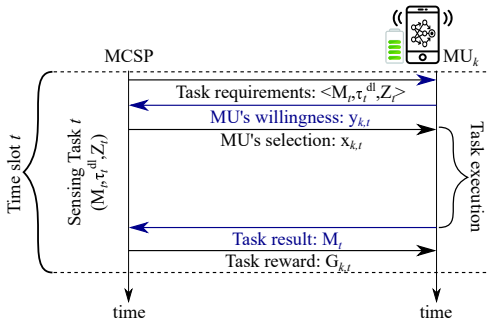


Fig. 1. System model with task execution steps

Moreover, they require private MU information at the MCSP to make allocation decisions. To overcome this requirement, some works employ deep reinforcement learning to improve the task allocation decisions over time [10], [17]. However, the scalability issue and lack of user preferences remains in these works. To overcome this issue, [18] considers a multi-agent reinforcement learning solution to the task selection problem. However, the solution requires exchange of MU's sensitive information as well as MU task participation preferences in the training phase which raises a privacy concern.

In this work, we model a decentralized task participation problem in MCS, where the MUs propose to participate in a task and decide their task execution efforts in terms of the energy they are willing to spend. The MUs aim to maximize the payments received from the MCSP (individual goal) while maximizing the number of completed tasks (global goal) in a limited time horizon, and under the constraints of deadline, energy and budget. In such a complex scenario, the MUs must learn when to propose for a task and when to back off such that other MUs can potentially perform it.

To solve this problem, we propose a novel and scalable federated deep reinforcement learning (FDRL-PPO) algorithm based on a multi-agent reinforcement learning formulation. Our proposed approach does not require perfect non-causal knowledge regarding the amount of harvested energy, the communication channel conditions, or the future tasks. The proposed solution enables MUs to learn their own task participation strategy based on their own information. Leveraging our federated DRL approach, the MUs share their learned task participation strategy with other MUs without sharing any sensitive information. This helps the MUs to propose only to suitable tasks and to maximize their individual as well as the global goals. Moreover, the proposed approach is robust and adapts to MU dropouts and new MU connections.

The rest of the paper is organized as follows: Section II introduces the system model. In Section III we formulate the problem to maximize the number of completed tasks in a given time horizon. Section IV introduces the FDRL-PPO approach. In Section V, we discuss the simulation parameters and numerical evaluation and Section VI concludes the paper.

## II. System Model

The considered MCS system consists of an MCSP that sequentially publishes sensing tasks, and multiple EH MUs. We consider a time-slotted model in which time is divided into discrete time steps with index $t \in \{0, 1, \dots, T-1\}$ of duration $\tau^{\text{int}}$ each, where $T$ is the number of considered time steps. We assume the MCSP publishes one task per time step $t$. Consequently, the variable $t$ can be used as time step index as well as task index. The set $\mathcal{K} = \{0, 1, \dots, K-1\}$ contains the indices of the $K = |\mathcal{K}|$ available MUs.

### A. MCSP

As shown in Fig.1, at the beginning of each time step $t$, the MCSP publishes a new sensing task. A task $t$ is characterized by its requirements: $\langle M_t, \tau_t^{\text{dl}}, Z_t \rangle$, where $M_t$ is the task result size which the MUs have to send back to the MCSP after task execution in bit, $\tau_t^{\text{dl}}$ denotes the task deadline in s, and $Z_t$ is the task specific budget in monetary units. The requirement tuple characterizes the difficulty of task $t$. We define a task difficulty weight as

$$V_t = \xi M_t' + \omega \tau_t^{\text{dl'}}, \forall t, \tag{1}$$

such that $M_t'$ is the normalized task size given by $\frac{M_t}{M_{\max}}$ and $\tau_t^{\text{dl'}}$ is normalized task deadline given by $1 - (\frac{\tau_t^{\text{dl}}}{\tau^{\text{int}}})$. The variables $\eta, \omega \in [0, 1]$ are importance factors whose values decide on the importance of the task size and deadline in deciding the overall task difficulty. For a large task size $M_t$, the MUs would need to spend more resources to complete the task successfully. Therefore $V_t$ increases as $M_t$ increases. In contrast to this, as the task deadline $\tau_t^{\text{dl}}$ gets shorter, the task completion becomes more difficult, and hence, $V_t$ increases. The task budget $Z_t$ is defined by the data requesters such that $Z_t = \eta V_t$, where $\eta$ is the budget coefficient.

Based on their task selection strategy, some MUs convey their willingness to perform the task $t$ along with their desired payments $G_{k,t}$ to the MCSP. The MCSP processes all these proposals and selects the cheapest MUs for task execution within the task budget $Z_t$. The platform's acceptance decision $x_{k,t} \in \{0, 1\}$ is sent back to the MUs. $x_{k,t} = 1$ denotes that MU $k$'s participation request has been accepted by the MCSP, and $x_{k,t} = 0$ indicates that it has not been accepted. If MU $k$ transmits the sensing result fulfilling all of the task requirements, then the MCSP awards it with the respective desired payment $G_{k,t}$, otherwise not.

### B. Mobile Units

Each MU $k$ makes two consecutive decisions in each time step $t$, namely, a task participation decision and a selection of its transmit power. The task participation decision is denoted by $y_{k,t} \in \{0, 1\}$ such that $y_{k,t} = 1$ indicates MU $k$ is willing to perform the task $t$, and $y_{k,t} = 0$ indicates that it is not. The transmit power $p_{k,t}^{\text{tx}}$ is used by MU $k$ for the transmission of the task proposal message and the sensing result of task $t$. $p_{k,t}^{\text{tx}}$ can take any value in the range $[0, p_{\max}^{\text{tx}}]$ where $p_{\max}^{\text{tx}}$ is the maximum transmit power. By deciding the power $p_{k,t}^{\text{tx}}$, MUs have control over the transmission energy $E_{k,t}^{\text{tx}} = \tau_{k,t}^{\text{tx}} p_{k,t}^{\text{tx}}$,

where $\tau^{\text{tx}}_{k,t}$ is the transmission time required to communicate $M_t$ from MU $k$ back to MCSP. $\tau^{\text{tx}}_{k,t}$ is defined by,

$$\tau^{\text{tx}}_{k,t} = \frac{M_t}{W \log_2\left(1 + \frac{p^{\text{tx}}_{k,t}|h_{k,t}|^2}{\sigma^2}\right)}, \tag{2}$$

where $W$ is the channel bandwidth in MHz, $h_{k,t}$ is the channel coefficient for the link between the MCSP and MU $k$ in time step $t$ and $\sigma^2$ is the noise power. Additionally, the MUs also spend energy while sensing. This sensing energy is denoted by $E^{\text{s}}_{k,t} = \tau^{\text{s}}_{k,t} p^{\text{s}}_{k,t}$, where $\tau^{\text{s}}_{k,t}$ is the sensing time required to generate the sensing data from MU $k$ at time step $t$ and $p^{\text{s}}_{k,t}$ is the power required for sensing. The sensing time $\tau^{\text{s}}$ is a random variable with mean value $\bar{\tau}^{\text{s}}$. The actual sensing time $\tau^{\text{s}}_{k,t}$ for each MU $k$ in time step $t$ depends on the task size $M_t$, the characteristics of the MU's sensor and its conditions in time step $t$. Similarly, the sensing power $p^{\text{s}}$ is a random variable with mean value of $\bar{p^{\text{s}}}$. The actual sensing power $p^{\text{s}}_{k,t}$ for MU $k$ in time step $t$ depends on the task size $M_t$, the characteristics of the MU's sensor and its conditions in time step $t$. The total effort in terms of $E^{\text{exec}}_{k,t}$ put in by MU $k$ in time step $t$ is measured as, $E^{\text{exec}}_{k,t} = E^{\text{s}}_{k,t} + E^{\text{p}}_{k,t} + E^{\text{tx}}_{k,t}$. Similarly, the total execution time required for MU $k$ in $t$ is denoted by $\tau^{\text{exec}}_{k,t} = \tau^{\text{s}}_{k,t} + \tau^{\text{tx}}_{k,t}$. Since the task execution time starts after getting accepted from the MCSP, $\tau^{\text{p}}_{k,t}$ is not considered in the task execution time. A payment request $G_{k,t}$ is made proportional to the effort $E^{\text{exec}}_{k,t}$ by each MU $k$ who is willing to perform the task $t$ such that, $G_{k,t} = \kappa E^{\text{exec}}_{k,t}$, where $\kappa$ is a factor in monetary units per Joule.

Each MU $k$ harvests $E^{\text{harv}}_{k,t}$ energy in Joules in every time step $t$. This energy is stored in a battery with capacity $B_{\max}$ without any losses. The MUs update their battery status $b_{k,t}$ at the end of time step $t$ as,

$$b_{k,t} = \min\{B_{\max}, b_{k,t-1} - E^{\text{exec}}_{k,t} + E^{\text{harv}}_{k,t}\}, \tag{3}$$

since the MUs' battery capacity is $B_{max}$. To ensure energy causality, each MU $k$ may only use the amount of energy available in the battery at the beginning of each time step $t$.

## III. Problem Formulation

### A. Centralized task allocation problem

To obtain an upper bound on the performance of the MCS system in terms of number of completed tasks in a finite number of time steps, we initially consider the problem from the MCSP's perspective. In this case, we assume that the MCSP has perfect non-causal knowledge about the battery statuses, amounts of harvested energy, communication channel conditions and future tasks. With this knowledge, the MCSP optimally decides which MU to choose for each task and how much transmit power this MU should select to maximize the average weighted number of completed tasks. Thus, for this problem formulation, MUs do not have any preference, on the contrary, the MCSP decides for each MU $k$. This problem is NP-hard and grows exponentially as the number of MUs increase [19]. The two decisions, task allocation and transmit power selection, are stored in the matrices $\mathbf{Y}$ and $\mathbf{P}^{\text{tx}}$ respectively, such that $\mathbf{Y} = (y_1, y_2, \ldots, y_T)$, where $y_t = (y_{1,t}, y_{2,t}, \ldots, y_{K,t})^T$. Similarly, $\mathbf{P}^{\text{tx}} = (p^{\text{tx}}_1, p^{\text{tx}}_2, \ldots, p^{\text{tx}}_T)$, where $p^{\text{tx}}_t = (p^{\text{tx}}_{1,t}, p^{\text{tx}}_{2,t}, \ldots, p^{\text{tx}}_{K,t})^T$.

A deadline constraint for each task $t$ has to be fulfilled by every MU $k$ which is selected for task execution, i.e.,

$$\tau^{\text{exec}}_{k,t} y_{k,t} \leq \tau^{\text{dl}}_t, \forall k \in \mathcal{K}_t, \forall t. \tag{4}$$

For this, the MCSP decides the transmit power $p^{\text{tx}}_{k,t}$ which follows (4). Moreover, due to the optimality, one MU is sufficient to complete the task, i.e.,

$$\sum_{k=0}^{K-1} y_{k,t} \leq 1, \quad \forall k, \forall t. \tag{5}$$

Note that the MCSP may choose not to perform a sensing task by allocating no MUs to it , and thus save the MUs' resources for more difficult future tasks. The budget constraint is given by the equation,

$$\sum_{k=0}^{K-1} G_{k,t} y_{k,t} \leq Z^{\text{task}}_t, \quad \forall t. \tag{6}$$

The energy causality constraint given by,

$$\sum_{j=1}^{J} E^{\text{exec}}_{k,j} y_{k,t} \leq \sum_{j=0}^{J-1} E^{\text{harv}}_{k,j}, \forall k, J = 1, \ldots, T, \tag{7}$$

guarantees that MU $k$ does not spend more energy than $b_{k,t}$ in time step $t$. Moreover, the MUs cannot spend $E^{\text{harv}}_{k,t}$ in time step $t$ itself. The overflow constraint given by,

$$\sum_{j=0}^{J-1} E^{\text{harv}}_{k,j} - \sum_{j=1}^{J} E^{\text{exec}}_{k,j} y_{k,t} \leq B_{\max}, \forall k, \forall t, \forall J. \tag{8}$$

ensures that the maximum value of energy that can be stored in the battery is $B_{\max}$. The optimization problem from the perspective of MCSP to maximize the average weighted sum of completed tasks is given as follows,

$$\begin{aligned} &\underset{\{y_{k,t}, p^{\text{tx}}_{k,t}\}}{\arg\max} \quad \sum_{t=0}^{T-1} V_t \sum_{k=0}^{K-1} y_{k,t} \\ &\text{subject to} \quad (4), (5), (6), (7), (8). \end{aligned} \tag{9}$$

Note that these constraints are inter-dependent and non-convex. To fairly handle different tasks with different requirements, we maximize the average weighted sum of completed tasks denoted by (9) instead of simply maximizing the number of completed tasks. Our proposed FDRL-PPO algorithm overcomes the requirement of non-causal knowledge and learns better task participation decisions over time.

### B. Reformulation as Markov Game

The optimization problem in (9) is formulated from the perspective of the MCSP to fulfil the assumption that one entity in the MCS scenario has the perfect non-causal knowledge about all the MUs and future tasks. This assumption is unrealistic to fulfill, and the MU task preferences are ignored in this approach. Additionally, this approach is not scalable.

To overcome these drawbacks, we formulate the task selection problem from the perspective of the MUs.

Each MU $k$ makes decisions about the task participation in time step $t$, i.e., $y_{k,t}$, and the transmit power $p^{\text{tx}}_{k,t}$ to determine the energy efforts to put in executing the task, if selected. Such decision making problems where multiple entities, in our case, the MUs, decide about their actions based on their observations can be formulated using a Markov Game (MG). An MG is characterized by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$. $\mathcal{S}$ is the set of states each MU can observe. The state $S_t \in \mathcal{S}$ helps MU $k$ to take action $A_t \in \mathcal{A}_k$ in time step $t$. $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K$ is collection of action sets of all MUs from set $\mathcal{K}$. The set $\mathcal{P}$ contains transition probabilities $P(S_{t+1}|S_t, A_{k,t})$, i.e., the probability of visiting state $S_{t+1} \in \mathcal{S}$ given the current state of the agent is $S_t \in \mathcal{S}$ and it takes action $A_{k,t} \in \mathcal{A}_k$. The reward set $\mathcal{R}$ contains all the possible rewards that the learning agent can receive after taking some action $A_{k,t} \in \mathcal{A}_k$ in state $S_t \in \mathcal{S}$.

In our scenario, the state $S_{k,t} \in \mathcal{S}$ of MU $k$ in time step $t$ is $S_{k,t} = \langle b_{k,t}, \bar{h}_{k,t}, M_t, \tau^{\text{dl}}_t, Z_t \rangle$, where $\bar{h}_{k,t}$ is the average channel coefficient calculated from the past observed coefficients since causal knowledge of $h_{k,t}$ is difficult to obtain. Similarly, each MU $k$ takes action $A_{k,t} = \{y_{k,t}, p^{\text{tx}}_{k,t}\}$, $A_{k,t} \in \mathcal{A}$, in each time step $t$. Since the transmit power $p^{\text{tx}}_{k,t}$ is continuous in range $[0, p^{\text{tx}}_{\max}]$, the set $\mathcal{A}$ has infinitely many possible actions. We assume that $\mathcal{P}$ is unknown since the MUs do not have a perfect non-causal knowledge about the other MUs, their actions, and also the future tasks. Finally, each MU $k$ receives its desired payment $G_{k,t}$ as a reward $R_{k,t} \in \mathcal{R}$ for taking action $A_{k,t}$ in state $S_{k,t}$ for task $t$. $R_{k,t} = G_{k,t}$ if the constraints (4)-(8) are fulfilled, else $R_{k,t} = 0$.

Each MU $k$ aims to maximize long-term discounted reward $R = \sum_{t=0}^{\infty} \gamma^t R_{k,t}$, where $\gamma \in [0, 1]$ is the discount factor, by finding a task selection policy $\pi^k$ which maps the state $A_{k,t}$ to the action $S_{k,t}$. Given the infinite number of states in our problem, the policy is modeled using an artificial neural network, termed policy network, with parameters $\theta^k$, such that $A_{k,t} = \pi^k(S_{k,t}; \theta^k)$. To evaluate how good or bad is a policy $\pi^k$, a value function $V^{\pi^k}$ is defined. Similar to the policy, $V^{\pi^k}$ is modeled using an artificial neural network, termed value network, with parameters $\phi^k$, such that $V^{\pi^k}(S_{k,t}; \phi^k)$. All the MUs optimize their policies $\pi^k$ over time with an aim to converge to the optimal policy $\pi^*$ that maximizes the payment $G_k$ for each MU $k$.

## IV. Reinforcement Learning Solution

### A. PPO-based task selection strategy

In our proposed solution, each MU $k$ implements a deep reinforcement learning approach called Proximal Policy Optimization (PPO) [20]. PPO is an actor-critic policy gradient method consisting of two networks, the actor, or policy network, and the critic, or value network. The actor decides which action $A_{k,t}$ should be taken in a given state $S_{k,t}$ based on MU $k$'s policy $\pi^k(A_{k,t}|S_{k,t}; \theta^k)$. The critic informs the actor how good or bad was this action $A_{k,t}$ in reality and how it should be adjusted by computing the value function $V^{\pi^k}(S_{k,t}; \phi^k)$.

**Algorithm 1** FDRL-PPO

**Initialization:**
Initialize a global model with weights $\Omega(0)$ at the MCS platform.
Each MU $k$ initializes a local model $\Omega_k(0), \forall k \in \mathcal{K}$ and sets it with: $\Omega_k(0) = \Omega(0)$.
**for** each round $r = 0, 1, \ldots, r_{\max} - 1$ **do**
  **for** each MU $k = 0, 1, \ldots, K - 1$ **do**
    Download global model $\Omega(r)$ from the platform and set $\Omega_k(r) = \Omega(r)$.
    Train the model locally using weights $\Omega(r)$. ▷ Section IV-A
    Upload weights after training to the MCS platform.
  **end for**
  **At the MCS platform:**
  Collect all weight updates from all MUs.
  Compute federated averaging to obtain $\Omega(r + 1)$
  Distribute updated weights to all MUs
**end for**

To train our proposed approach, multiple training episodes are considered. A training episode $i$ consists of $T$ time steps in which each MU observes its own states, takes its individual actions and observes the individual rewards. The observed states, selected actions and rewards within one training episode $i$ are termed trajectory $\mathcal{D}^k_i$. At the beginning of the first training episode $i = 1$, each MU $k$ initializes its policy parameters $\theta^k$ and value function parameters $\phi^k$. These parameters are updated in each subsequent training episode using the observed trajectories $\mathcal{D}^k_i$. Specifically, at the end of each training episode, the policy network (i.e. the actor) updates the parameters $\theta^k$ in the direction that maximizes the average long term rewards via stochastic gradient ascent algorithm. Similarly, the value network (i.e. the critic) updates $\phi^k$ by minimizing the loss function which represents the gap between the expected long term rewards and the actual rewards using stochastic gradient descent algorithm. The same procedure is repeated until convergence is achieved.

### B. Federated Deep Reinforcement Learning using PPO

In the previous section, we explained the PPO algorithm. This algorithm is implemented on each MU $k$ and is trained based on the observed trajectories $\mathcal{D}^k_i$. In the considered MCS scenario, each MU should ideally learn to which tasks it should propose and to which tasks it should not propose and let other, potentially capable MUs, participate. However, with no communication between the MUs, this is difficult to learn. Since the considered MCS scenario is a mixed cooperative and competitive, there has to be some cooperation between the MUs. By cooperating with each other, MUs can collectively learn to maximize their own payments and also maximize the average weighted sum of completed tasks in a finite time horizon. This cooperation is achieved by sharing their learnt models, i.e., $\theta_k$ and $\phi_k$. This knowledge sharing also helps the newly associated MUs in the MCS system because they can take advantage of the already learned models of other MUs. Note that by sharing the learning parameters, the MU's private and sensitive data is preserved and not shared.

Our proposed approach enables this sharing by exploiting a federated learning algorithm to distributively train the MUs

using PPO. In FDRL-PPO, an aggregator node (in our case the MCSP) maintains a global model $\Omega(r)$. This model consists the policy and critic network parameters $\theta$ and $\phi$, respectively. We divide each training episode $i$ into $T^{\text{fed}}$ federation rounds indexed by $r$, such that each federation round $r$ is formed by $T/T^{\text{fed}}$ time steps. At the beginning of federation round $r$, each MU $k$ downloads the global model and trains this model locally based on its own decisions and experience. At the end of round $r$, each MU $k$ transmits this trained local model $\Omega_k(r)$ to the MCSP. The MCSP combines the received parameters based on the individual rewards of each agent as,

$$\Omega(r+1) = \sum_{k=0}^{K-1} \frac{R_{k,r}}{\sum_k R_{k,r}} \Omega_k(r) \tag{10}$$

This way, the aggregated model $\Omega(r+1)$ is more similar to the best performing MU. After this, the MUs download the updated global model $\Omega(r+1)$ and again train this model based on their local data. The process repeats until convergence.

## V. Numerical evaluation

In this section, we present and discuss the simulation results to evaluate the performance of the proposed FDRL-PPO algorithm in comparison with the reference schemes. These results are averaged over $I = 1000$ independent realizations. In each realization, we consider $T = 100$ time steps and as a result, same number of tasks. Each task size $M_t$ of task $t$ can take any value in the range $[1, 7]$ Mbit. The MUs are in an area where the maximum distance between an MU and the MCSP is 1 km. The communication channel between MU $k$ and the MCSP is modelled as a Rayleigh fading channel with path loss exponent of three. The MUs can move freely in the area with a maximum average speed of $10\,\text{km/h}$. We set $\xi$ and $\omega$ to one to promote fair task completion. Table I provides the list of simulation parameters.

For the proposed FDRL-PPO approach, we use a discount factor $\gamma = 0.99$. For the policy and value network, we use an artificial neural network with 2 hidden layers each with 64 nodes. Moreover, we use a learning rate $l_r = 1e^{-6}$. A training batch size of 1024 is used with a mini-batch size 128 to compute the stochastic gradient descent in the learning process. We consider the following reference schemes for the performance comparison.

**Optimal task allocation (OTA):** As described in Section III-A, this is a centralized task allocation scheme which assumes perfect non-causal knowledge about the amount of harvested energy, communication channel conditions and the future tasks and their requirements. Although unrealistic in practice, this scheme provides the performance upper bound.

**Myopically optimal task participation (MOTP):** This approach assumes that each MU has the perfect causal knowledge about the communication channel coefficients. Based on this, each MU $k$ makes an informed decision on task proposal and transmit power selection. Thus, this scheme makes myopically optimal decisions for the respective time step without any concern about future consequences.

**Always participating scheme (APS):** In this scheme, each MU $k$ proposes to all the tasks. The MUs only focus on maximizing the short term payments without focusing on the long term consequences.

**Reinforcement Learning with Independent Agents (RLIA):** This is a DRL scheme based on PPO where each MU $k$ improves its own task selection policy without sharing any knowledge with other MUs.

In Fig. 2, we compare the performance in terms of average weighted sum of completed tasks of our proposed solution with the above mentioned reference schemes. Since OTA is a centralized task allocation scheme, it provides the upper bound by performing 58.55 weighted tasks on average. In comparison, MOTP performs on 44.57 weighted tasks. Our proposed FDRL-PPO scheme performs 42.44 weighted tasks which is approx. 95.22% of the MOTP performance and approx. 72.48% of the OTA performance. Note that the FDRL-PPO has no requirement of causal or non-causal knowledge about the amounts of harvested energy, communication channel conditions and future tasks. The RLIA performs 28.57 weighted tasks on average and finally APS performs 16.81 weighted tasks on average. With this, FDRL-PPO outperforms RLIA and APS by at least 48.54% and 152.46% , respectively. This is because with FDRL-PPO, each MU learns about its task participation decision the efforts required to perform the task. It also learns about the consequences of the current task participation decisions on future states.

In Fig. 3, the performance in terms of average weighted sum of completed tasks of the FDRL-PPO is compared with the referenced schemes by varying the number of associated MUs in the MCS scenario. We omit the OTA in this study since it is not scalable for higher number of MUs. For the rest of the schemes, we see improvement in the performance as the number of MUs increase. This is expected since now there are more MUs who can potentially perform the tasks with different difficulties. The only limiting factor is the budget. Due to this, the MCSP only selects the MUs according to (6). However, it is important to note that as $K$ increases, the performance of the FDRL-PPO converges to the performance of MOTP. For $K = 15$, the FDRL-PPO performs only 1% lower than the MOTP. This shows that the FDRL-PPO is robust and is scalable to higher number of MUs.

In Fig. 4, we compare the performances in terms of average weighted sum of completed tasks of the considered schemes by varying the task budget coefficient $\eta$. Due to this, the task budget $Z$ also varies. The performance of OTA increases with increase in task budget $Z$, because it has more budget to also incorporate expensive MUs and perform tasks. On the contrary, MOTP and APS exhibit similar downward trend in performance as $Z$ increases. This is because more MUs can be selected to perform tasks and they eventually exhaust their batteries by performing tasks. However, FDRL-PPO improves its performance slightly as the budget increases which show trend similar to that of OTA. For higher budget, FDRL-PPO performs only 5.76% lower than the MOTP.

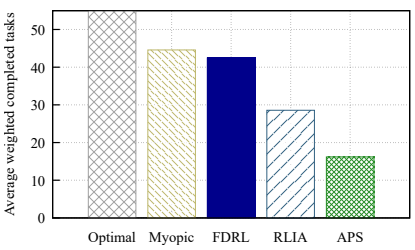


Fig. 2. Performance comparison for all approaches with $K = 10$ MUs

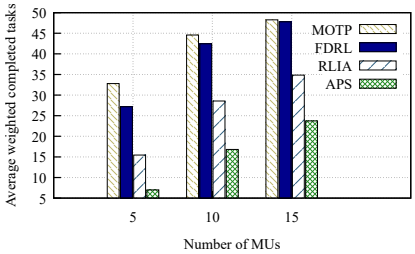


Fig. 3. Average weighted sum of completed tasks vs. number of MUs $K$

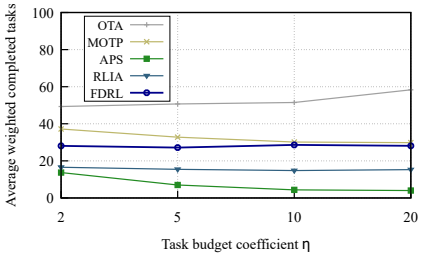


Fig. 4. Average weighted sum of completed tasks vs. task budget coefficient $\eta$

TABLE I
SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Total time steps $T$ | 100 time steps |
| Duration of one time step $t = \tau^{\text{int}}$ | 1 s |
| Number of available MUs $K$ | [5, 15] |
| MU distances to MCSP $[d_{\min}, d_{\max}]$ | $[200, 1000]$ m |
| Battery capacity $B_{\max}$ | 800 mW s |
| Maximum harvested energy $E^{\text{harv}}_{\max}$ per $t$ | 5% of $B_{\max}$ |
| Total Bandwidth $W$ per MU $k$ | 1 MHz |
| Noise power $\sigma^2$ | $10^{-16}$ W |
| Transmit power $p^{\text{tx}}_{\max}$ of sensor $k$ | 200 mW |
| Channel gain $\lvert h_{k,t}\rvert^2$ | $\sim d^{-3}$ (Urban scenario) |
| Sensing task size $M$ | $[1 - 7]$ Mbit |
| Deadline $\tau^{\text{dl}}$ | Uniform in $[\frac{\tau^{\text{int}}}{2}, \tau^{\text{int}}]$s |
| Task Budget $Z$ | $\eta V$ |

## VI. CONCLUSION

In this work, we studied the task participation problem in an budgeted and EH MCS scenario to maximize the average weighted sum of completed tasks in a finite time horizon. By considering their own preferences, the MUs make task participation decisions, i.e., they decide whether or not to propose to participate in a task and select their transmit power, which ultimately reflects in their task execution efforts. We showed that optimization-based approaches require at least causal knowledge about the amount of harvested energy, wireless communication channel conditions and future tasks' requirements. However, it is unrealistic to assume the availability of such knowledge at the MCSP in practical MCS systems. To this aim, we proposed a FDRL-PPO approach to solve the task participation problem from the MU's perspective and without this (non)-causal knowledge. By utilizing federated learning, FDRL-PPO improves the collective performance of all MUs. Specifically, the MUs share their learnt models about their own task participation strategies. As a result, the MCS system becomes robust to MU dropouts and new connections. Simulation results showed that FDRL-PPO performs only $4.78\%$ lower than the myopically optimal strategy which has un-realistic causal knowledge about the MCS system. Moreover, FDRL-PPO outperforms traditional multi-agent reinforcement learning-based schemes with independent agents and the always participating scheme by $48.54\%$ and $152.46\%$, respectively. In a nutshell, this work presented a scalable, privacy-preserving, and adaptable solution for the decentralized task participation problem in MCS, contributing to the advancement of efficient and effective MCS systems.

## REFERENCES


[1] R. K. Ganti *et al.*, "Mobile crowdsensing: current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, 2011.

[2] C. Dai *et al.*, "Stable task assignment for mobile crowdsensing with budget constraint," *IEEE Trans. on Mobile Comput.*, vol. 20, no. 12, pp. 3439–3452, 2021.

[3] J. An *et al.*, "Mobile crowd sensing for internet of things: A credible crowdsourcing model in mobile-sense service," in *IEEE Int. Conf. on Multimedia Big Data*, 2015, pp. 92–99.

[4] Foursquare. [Online]. Available: https://location.foursquare.com/

[5] Komoot. [Online]. Available: https://www.komoot.com/

[6] T. A. N. Dinh *et al.*, "Spatial-temporal coverage maximization in vehicle-based mobile crowdsensing for air quality monitoring," in *IEEE Wireless Commun. and Networking Conf. (WCNC)*, 2022, pp. 1449–1454.

[7] X. Dong *et al.*, "Optimal mobile crowdsensing incentive under sensing inaccuracy," *IEEE IoT Journal*, vol. 8, no. 10, pp. 8032–8043, 2021.

[8] R. Pryss *et al.*, "Requirements for a flexible and generic API enabling mobile crowdsensing mhealth applications," in *Int. Workshop on Requirements Engineering for Self-Adaptive, Collaborative, and Cyber Physical Systems (RESACS)*, 2018, pp. 24–31.

[9] appjobber. [Online]. Available: https://appjobber.de/

[10] S. Dongare *et al.*, "Deep reinforcement learning for task allocation in energy harvesting mobile crowdsensing," in *IEEE Global Commun. Conf.*, 2022, pp. 269–274.

[11] M. M. Sandhu *et al.*, "Task scheduling for energy-harvesting-based iot: A survey and critical analysis," *IEEE IoT Journal*, vol. 8, no. 18, pp. 13 825–13 848, 2021.

[12] Y. Huang *et al.*, "Opat: Optimized allocation of time-dependent tasks for mobile crowdsensing," *IEEE Trans. on Industrial Informatics*, vol. 18, no. 4, pp. 2476–2485, 2022.

[13] Y. Liu *et al.*, "Taskme: Multi-task allocation in mobile crowd sensing." New York, NY, USA: Assoc. for Comput. Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2971648.2971709

[14] J. Wang *et al.*, "Multi-task allocation in mobile crowd sensing with individual task quality assurance," *IEEE Trans. on Mobile Comput.*, vol. 17, no. 9, pp. 2101–2113, 2018.

[15] B. Simon *et al.*, "Delay- and incentive-aware crowdsensing: A stable matching approach for coverage maximization," in *ICC 2022 - IEEE Intl. Conf. on Commun.*, 2022, pp. 2984–2989.

[16] M. H. Cheung *et al.*, "Distributed time-sensitive task selection in mobile crowdsensing," *IEEE Trans. on Mobile Comput.*, vol. 20, no. 6, pp. 2172–2185, 2021.

[17] C. Xu *et al.*, "Intelligent task allocation for mobile crowdsensing with graph attention network and deep reinforcement learning," *IEEE Trans. on Network Science and Engg.*, vol. 10, no. 2, pp. 1032–1048, 2023.

[18] Y. Chen *et al.*, "Intelligentcrowd: Mobile crowdsensing via multi-agent reinforcement learning," *IEEE Trans. on Emerging Topics in Comput. Intelligence*, vol. 5, no. 5, pp. 840–845, 2021.

[19] X. Wei *et al.*, "Data quality aware task allocation with budget constraint in mobile crowdsensing," *IEEE Access*, vol. 6, pp. 48 010–48 020, 2018.

[20] J. Schulman *et al.*, "Proximal policy optimization algorithms," 2017.