Multiagent DDPG-Based Deep Learning for Smart Ocean Federated Learning IoT Networks

Dohyun Kwon, Joohyung Jeon, Soohyun Park, Joongheon Kim, *Senior Member, IEEE*, and Sungrae Cho, *Member, IEEE*

***Abstract*—This article proposes a novel multiagent deep reinforcement learning-based algorithm which can realize federated learning (FL) computation with Internet-of-Underwater-Things (IoUT) devices in the ocean environment. According to the fact that underwater networks are relatively not easy to set up reliable links by huge fading compared to wireless free-space air medium, gathering all training data for conducting centralized deep learning training is not easy. Therefore, FL-based distributed deep learning can be a suitable solution for this application. In this IoUT network (IoUT-Net) scenario, the FL system needs to construct a global learning model by aggregating the local model parameters that are obtained from individual IoUT devices. In order to reliably deliver the parameters from IoUT devices to a centralized FL machine, base station like devices are needed. Therefore, a joint cell association and resource allocation (JCARA) method is required and it is designed inspired by multiagent deep deterministic policy gradient (MADDPG) to deal with distributed situations and unexpected time-varying states. The performance evaluation results show that our proposed MADDPG-based algorithm achieves 80% and 41% performance improvements than the standard actor–critic and DDPG, respectively, in terms of the downlink throughput.**

***Index Terms*—Deep reinforcement learning, federated learning (FL), smart ocean networks.**

I. Introduction

FOR THE last few decades, communication technologies have been consistently developed for application-specific extreme areas, such as spaces and deep underwater oceans [1]. Since the underwater environment is mostly unexplored by human beings, it is expected that there will be a lot of research and exploration attempts about the undersea spaces and resources. Because the underwater environment has the accessibility limitations of human beings due to depth, temperature, and underwater pressure, Internet of Underwater Things (IoUT) is challenging as well as worthy to conduct research for making the systems be reliable and robust. As unpredictable changes can occur in the deep underwater environment and underwater medium takes huge fading effects compared to the wireless air free-space medium, the technologies for underwater wireless networks to provide seamless communication services with IoUT devices are essentially desired [2].

Among many applications in IoUT network (IoUT-Net) scenarios, deep learning-based applications are actively and widely discussed, nowadays. In conventional deep neural network (DNN) training, the data set for the training should be located in a single centralized storage [3]. However, this assumption is occasionally not realistic especially in IoUT-Net scenarios due to the unreliable wireless networking by relative huge fading effects and unpredictable environmental changes. Therefore, distributed DNN deep learning training, also called federated learning (FL), is essentially required in our scenarios [4]–[9]. For FL in our considering IoUT-Net scenarios, each IoUT device gathers its own monitoring and surveillance training data. After that, each IoUT device conducts DNN training with the data and then constructs its own local model. The parameters of the local model in each IoUT device will be delivered to a central machine where the central machine constructs a global learning model based on the local model parameters. After conducting a certain number of iterations, the global model converges. In this case, some IoUT devices may have insufficient data. Then, the IoUT devices should gather more data before starting local model training. If the data are still not sufficient until the threshold time, there is no way to avoid local model training. When this local training with insufficient data conducts, the model suffers from overfitting. In order to overcome this problem, we can adaptively adjust the importance (i.e., weights) of individual local models when the global model is constructed via: 1) the weighted summation of local parameters [10] and 2) attention mechanism-based adaptive control of weights in local model parameter aggregation [11].

For FL in our considering IoUT-Net scenarios, different types of devices exist, as illustrated in Fig. 1. As shown in Fig. 1, an optical base station (OBS) exists for centralized computation by gathering all data from its associated base stations (BSs). Note that this OBS conducts the computation for the FL global model construction by aggregating data from its BSs. Then, the BSs except OBS in IoUT-Net scenarios can be

Manuscript received October 31, 2019; revised March 11, 2020; accepted April 12, 2020. Date of publication April 15, 2020; date of current version October 9, 2020. This work was supported in part by the National Research Foundation of Korea under Grant 2019M3E4A1080391, and in part by the Institute for Information and Communications Technology Promotion grant funded by the Korea Government (MSIT, A Development of Driving Decision Engine for Autonomous Driving Using Driving Experience Information) under Grant 2017-0-00068. *(Corresponding authors: Joongheon Kim; Sungrae Cho.)*

Dohyun Kwon is with Research Laboratory, Hyundai Autoever, Seoul 06179, South Korea.

Joohyung Jeon and Sungrae Cho are with the School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea (e-mail: srcho@cau.ac.kr).

Soohyun Park and Joongheon Kim are with the School of Electrical Engineering, Korea University, Seoul 02841, South Korea (e-mail: joongheon@korea.ac.kr).

Digital Object Identifier 10.1109/JIOT.2020.2988033

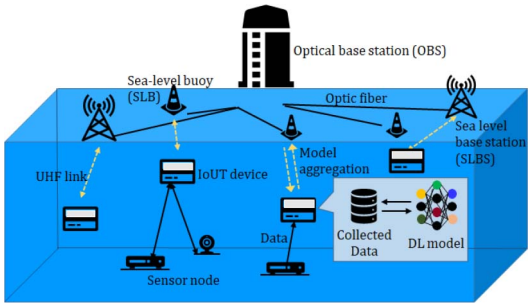


Fig. 1. Reference network model for smart ocean FL IoUT applications.

sea-level buoy (SLB) and seal-level BS (SLBS). The SLB and SLBS can setup underwater wireless links with IoUT devices. Each IoUT device trains its own model with its own local data. Then, the trained parameters should be delivered to the centralized OBS via BSs. Thus, the centralized OBS eventually gathers all parameters from its associated IoUT devices in order to construct the FL global model. Finally, the computed global model at OBS should be eventually delivered to its associated IoUT devices in order to let all the IoUT devices use the global parameters in their DNN models. Therefore, enhancing the downlink (DL) throughput, i.e., from BS to its associated IoUT devices for FL global model distribution, is important in this reference network model [9].

In order to improve the DL throughput performance, this article proposes a novel joint cell association and resource allocation (JCARA) method. Furthermore, in order to properly handle unexpected situations in underwater wireless networks and channel unreliability, one of the major multiagent deep reinforcement learning algorithms which is called multiagent deep deterministic policy gradient (MADDPG) is used in this article. The reason why traditional single-agent deep reinforce learning (DRL) approaches are hard to use is that they may fail to learn an optimal policy due to the partially observable and nonstationary environment. Therefore, this article solves the JCARA problem of FL in IoUT-Net scenarios with the MADDPG-based approach [12]. Based on the MADDPG strategy, each IoUT device can learn its own policies to cooperatively solve the JCARA problem [13]. The simulation results show that the proposed method outperforms other methods.

The major contributions of the proposed MADDPG algorithm can be summarized as follows.

1) This article proposes a novel multiagent deep reinforcement learning-based algorithm which can take of unexpected underwater environment changes and channel unreliability.
2) The proposed algorithm can work in a fully distributed manner thanks to the nature of multiagent computation in MADDPG. This is definitely helpful for our distributed FL applications.
3) To the best of our knowledge, this article is the first work which considers FL procedures and systems for smart ocean applications.

The remainder of this article is organized as follows. Section II presents our related work. The system model of IoUT-Net and our considering JCARA problem are presented in Section III. Based on the network architecture and problem definition, a multiagent DRL-based algorithm is proposed in Section IV. Then, Section V verifies the performance via data-intensive simulations. Finally, Section VI concludes this article and provides future research directions.

II. Related Work

This section summaries our previous and related work in the following categories.

Cell Association and Resource Allocation: There have been many research results for this JCARA problem. The resource allocation problem of ultrahigh-frequency (UHF) network was considered in [14] and [15]. In addition, the JCARA problems have been studied in [16]–[19]. However, because of the NP-hard nature and combinatorial optimization formulation of the problem, it is not possible to achieve global optimal solutions. Therefore, there are many approaches to efficiently handle the problem, including the graph-theoretical approach [20], integer programming method [21], matching game-based solution approach [22], and stochastic geometric strategies [23]. These approaches are still limited to solve the JCARA problem as they need nearly precise information, such as full knowledge of channel state information (CSI) or fading wireless channel models. In practice, such accurate information may not be available so that computing the optimal solutions of the JCARA problem is intractable. Therefore, this article proposes a multiagent DRL approach in order to solve the JCARA problem in a way that improves the DL throughput of IoUT devices in underwater FL. By enhancing the DL performance, the trained parameters in a centralized machine can be reliably distributed to its associated IoUT devices via BSs.

Reinforcement Learning Applications: The reinforcement learning has been widely applied to solve various types of complex sequential decision-making problems in distributed and wireless networks such as interference alignment in cache-enabled networks [24] and dynamic duty cycle selection technique in unlicensed spectrum [25]. Unlike the existing approaches, reinforcement learning needs only a few information to operate, such as the possible action spaces of a learning agent. Based on the interaction between an agent and its own environment, the reinforcement learning agent observes state transition and learns how to act good by updating its *policy* [26]. The agent estimates the expected total reward per possible actions for a given state and makes a decision on how to act on a sequential decision-making process. In [27], a power-efficient resource allocation framework for cloud radio access networks (RANs) is proposed based on DRL. Elsayed and Erol-Kantarci [28] proposed a DRL based latency reducing scheme of mission-critical services for wireless networks. They combined the long short-term memory (LSTM) [29] and Q-learning [30] to minimize the delay of the mission-critical services.

FL: The FL focuses on the issues in massively distributed, nonindependent and identically distributed (non-i.i.d.) unbalanced data, and limited communications which have a difference from typically distributed optimization [8], [31], [32]. On each training round, the *FedSGD* algorithm selects a

C-fraction of edge and conduct DNN model training overall local data held by these edges. The batch selecting mechanism on *FedSGD* differs with selecting a batch by choosing a random set uniformly, but the batch gradient g computed by *FedSGD* still satisfies $\mathbb{E}[g] = \nabla f(w)$. *FedAvg* let each edge conduct iterate the local update multiple times on each round. When the computation one each edge by *FedAvg* increases, FL can produce communication cost reduction [33], [34]. During the communication of centralized machine and edges, the centralized machine simply ignores the straggling edges which do not report back on time. This dropout property makes the FL system be a tolerance on communication constraints [35].

Underwater Wireless Communication: In the underwater environment acoustic waves, optical waves and radio waves are used for enabling wireless communications. Furthermore, the performances of underwater wireless communications are varying depending on each wave's propagation characteristics [36]. For the radio waves, attenuation occurs faster than the others, however, they are characterized by high data rates and lower power usages; and they do not demand the line-of-sight alignment of pollution nodes or clear waters [37], [38].

III. System Model and Problem Definition

This section presents the U-IoTNet system model which is a heterogeneous network (HetNet) based on the IoUT that is a technology of IoT at the sea. The 3-tier U-IoTNet HetNet system consists of OBS, SLBS, and SLB. In addition, each IoUT device cooperates with the BS and the corresponding wireless resource is allocated to each IoUT device in order to solve the JCARA problem in the U-IoTNet scenarios.

A. System Model

Our considering 3-tier IoUT-Net consists of K_O OBSs, K_B SLBSs, and K_b SLBs among K number of BSs, where $K = K_O + K_B + K_b$. In addition, there are N IoUT devices in the IoUT-Net scenarios. The set of BSs is denoted as $\mathcal{B}$ where

$$\mathcal{B} \triangleq \left\{b_1, \ldots, b_{K_O}, \ldots, b_{K_O+K_B}, \ldots, b_{K_O+K_B+K_b}\right\}. \quad (1)$$

To simplify the notations of the SLB and other BSs, the set of OBS and SLBSs and the set of SLBs are denoted as $\mathcal{B}_\alpha$ and $\mathcal{B}_\beta$, respectively, where

$$\mathcal{B}_\alpha \triangleq \left\{b_1, \ldots, b_{K_O}, \ldots, b_{K_O+K_B}\right\} \quad (2)$$

$$\mathcal{B}_\beta \triangleq \left\{b_{K_O+K_B+1}, \ldots, b_{K_O+K_B+K_b}\right\}. \quad (3)$$

Each IoUT device can be scheduled/matched with only one BS in each time slot. Thus, the scheduling information of the ith IoUT device with a BS among $\mathcal{B}$ can be represented as a binary vector $a_i^k \in \{0, 1\}$, i.e., the vector $a_i^k(t)$ is denoted as

$$a_i^k(t) \triangleq \left(a_i^1(t), \ldots, a_i^K(t)\right) \quad (4)$$

where $k \in [1, K]$ and $i \in [1, N]$. If the ith IoUT associates with the kth BS, $a_i^k(t) = 1$. Otherwise, $a_i^k(t) = 0$. Therefore, $a_i^k(t)$ can be represented as follows:

$$\sum_{k=1}^{K} a_i^k(t) \leq 1 \quad \forall i \in [1, N]. \quad (5)$$

Each IoUT device can utilize *carrier aggregation* which combines multiple subchannels of the associated BS, however, it is able to actually use only the spectrum at most $\bar{c}$ due to fair resource access limitation. We assume that OBS and SLBSs share $\mathcal{X}$ orthogonal channels and SLBs have $\mathcal{Y}$ UHF channels. Here, S means the set of orthogonal channels and it can be denoted as $S \triangleq \{s_1, \ldots, s_X\}$. In addition, C, the set of UHF channel, can be signified as $C \triangleq \{c_1, \ldots, c_\mathcal{Y}\}$. Therefore, the resource allocation vector between the ith IoUT device and the jth OBS/SLBS can be denoted as $f_i^j(t)$, where $f_i^j(t) = (s_1^j(t), \ldots, s_X^j(t))$, $i \in [1, N]$, and $j \in [1, K_O + K_B]$. If the IoUT device uses the uth channel among $f_i^j(t)$, $s_x^u(t) = 1$. Otherwise, $s_x^u(t) = 0$. However, if the IoUT device associates with the oth SLB, the resource allocation vector between the ith IoUT and the oth SLB can be denoted as $f_i^o(t) = (c_1^o(t), \ldots, c_\mathcal{Y}^o(t))$, where $i \in [1, N]$ and $o \in [K_O + K_B + 1, K_O + K_B + K_b]$. The way of setting the value of $f_i^o(t)$ follows the criteria of $f_i^j(t)$. Finally, the resource allocation toward the ith IoUT device is as follows:

$$\sum_{o=K_O+K_B+1}^{K} f_i^o(t) + \sum_{j=1}^{K_O+K_B} f_i^j(t) \leq \bar{c} \quad \forall i \in [1, N]. \quad (6)$$

In addition, we also have to consider co-channel interference because the SLBSs' coverage coexists in the coverage of OBS. Suppose that $p_{i,j}^s(t)$ stands for the possible transmit power-level (finite discrete value) vector of OBS or SLBS on the shared spectrum, $p_{i,j}^s(t) = (p_{i,j}^{s_1}(t), \ldots, p_{i,j}^{s_X}(t))$ represent the transmit power per each channel in S. Each IoUT device is assumed to measure instantaneous channel gain $h_i^j(t)$, where $i \in [1, N]$ and $j \in [1, K]$. Therefore, the signal-to-interference-plus-noise ratio (SINR) at the ith IoUT device which is associated with the kth BS among $\mathcal{B}$ (i.e., b_k) using channel $f_i^j(t)$ or $f_i^o(t)$ is as following (7), as shown at the bottom of the next page [denoted by $\Psi_{i,k}^{S,C}(t)$], where W is a channel bandwidth, N_0 stands for the noise power, $v \in \mathcal{B}$, $i \in [1, N]$, $j \in [1, K_O + K_B]$, and $o \in [K_O + K_B + 1, K_O + K_B + K_b]$, respectively.

According to (7), the DL throughput of the ith IoUT device, i.e., $\zeta_i(t)$, can be as follows:

$$\zeta_i(t) = \sum_{k=1}^{K} l_i^k(t) \sum_{\forall z \in S \cup C} W \log_2\left(1 + \Psi_{i,k}^z(t)\right). \quad (8)$$

B. JCARA Problem Formulation

Based on the aforementioned system model, the JCARA problem can be defined based on the concepts as follows: 1) IoUT devices associate with BS can utilize the wireless resource as well as and 2) IoUT devices associate with BS satisfy minimum QoS requirement χ, i.e.,

$$\sum_{k=1}^{K} l_i^k(t) \sum_{\forall z \in S \cup C} \Psi_{i,k}^z(t) \geq \chi. \quad (9)$$

According to the concept of the transmit power control at the kth BS toward the ith IoUT, the power-aware cost can be formulated as follows:

$$\kappa_i(t) = \sum_{k=1}^{K} \rho l_i^k(t) \left[\sum_{o=1}^{K_b} f_i^o(t) p_{i,j}^s(t) + \sum_{j=1}^{K_O+K_B} f_i^j(t) p_{i,j}^c(t)\right] \quad (10)$$

where ρ stands for the cost of unit power level. Overall, the total revenue $\Lambda_i(t)$ of the ith IoUT device is formulated as

$$\Lambda_i(t) = \eta\zeta_i(t) - \kappa_i(t) \tag{11}$$

where η stands for the positive profit of each channel capacity. Finally, the purpose of the JCARA problem is for optimizing the expected total return of (11) under (9). The expected total return of the revenue of the ith IoUT device can be expressed as $\Omega_i(t)$, i.e.,

$$\Omega_i(t) = \sum_{t=1}^{+\infty} \gamma^{t-1}\Lambda_i(t) \tag{12}$$

where $\gamma \in [0, 1)$ is the discounting factor in reinforcement learning that represents the uncertainty of future revenue. Throughout the equations from (5) to (12), IoUT devices and BSs dynamically transmit their resource utilization states and actions information. In addition, the transmission of the information is highly combinatorial as well as difficult to optimize. Therefore, multiagent DRL which is based on policy optimization is a reasonable solution in order to solve the JCARA problem in IoUT-Net scenarios.

IV. Multiagent DRL (MADDPG) for the Cooperative JCARA Problem in IoUT-Net Scenarios

Throughout multiple interactions with the IoUT-Net environment, each IoUT device accumulates its own experiences which are paired with $(s_i(t), a_i(t), r_i(t), s_i(t+1))$, where $s_i(t) \in \mathcal{S}$ means the observed local state of the ith IoUT device at time t, $a_i(t) \in \mathcal{A}$ denotes the action of the IoUT device at time t, and $r_i(t)$ signifies the temporal difference reward of the IoUT device at time t, respectively. The aforementioned traditional single-agent approaches for solving the JCARA problem are not capable of learning the cooperative spectrum access policy of IoUT devices, due to the nonstationary environment situations. $r_i(t)$ may differ from $s_i(t)$ and $a_i(t)$ in the set of experience pairs at the same time due to the fact that the observed state $s_i(t)$ of the ith IoUT device only contains local information. That is, the IoUT device only has local information of the IoUT-Net, and thus, the states and actions of other IoUT devices (which impact on the IoUT device's *reward*) may differ even the same local observations and actions of the IoUT device. Therefore, to solve the JCARA problem with multiple IoUT devices, the policy updating procedure of each IoUT device should take into account the actions of other IoUT devices, rather updating the policy only with its own action. Finally, it is clear that the multiagent approach is more suitable for optimizing the policies of IoUT devices to solve the JCARA problem in IoUT-Net scenarios.

A. Preliminaries of Reinforcement Learning

The reinforcement learning agent learns how to act (i.e., policy) in a sequential stochastic decision-making problem through the interactions with its environment. The decision making can be modeled as a Markov decision process (MDP) which is the pairs of $(s_i(t), a_i(t), r_i(t), s_i(t+1))$. Note that $\mathcal{S}$ and $\mathcal{A}$ stand for the sets of possible states of IoUT devices and the set of possible actions. The reinforcement learning agent aims to optimize its policy μ_{θ_i} which is parameterized with θ_i, i.e.,

$$a_i(t) = \mu_{\theta_i}(s_i(t)). \tag{13}$$

The policy updating procedure changes the parameter θ in a way the expected total return of the agent with respect to $a_i(t)$ for given $s_i(t)$ is improved. The values of actions for sequential observations (i.e., state s) are measured with the action-value function (or Q-function) to evaluate the expected total return per action. Here, the Q-function is formulated as

$$Q^{\mu}(s, a) = \mathbb{E}[\Omega_i(t)|s = s_i(t), a = a_i(t)] \tag{14}$$

$$= \mathbb{E}_{s'}\Big[r(s, a) + \gamma\mathbb{E}_{a'\sim\mu}\big[Q^{\mu}\big(s', a'\big)\big]\Big]. \tag{15}$$

B. MADDPG for JCARA

In this section, the proposed multiagent DRL strategy, i.e., deep deterministic policy gradient (PG), is presented to solve the JCARA problem. In DRL, DNN models are utilized to build the learning agent. The DNN takes the role of a nonlinear approximator to obtain the optimal policies μ^* in IoUT devices. Suppose that $\mu \triangleq \{\mu_1, \ldots, \mu_N\}$ be the set of all agent policies and $\theta \triangleq \{\theta_1, \ldots, \theta_N\}$ is the parameter set of the corresponding policy. Based on the estimation of the Q-function for each possible action, each IoUT device updates its own policy. The MADDPG is a PG-based off-policy *actor–critic* algorithm [13], where the objective function $\mathcal{J}(\theta)$ is an expected reward, i.e., $\mathcal{J}(\theta_i) = \mathbb{E}[\Omega_i(t)]$. That is, the optimal policy of each ith IoUT device can be represented as follows:

$$\mu^*_{\theta_i} = \arg\max_{\mu_{\theta_i}} \mathcal{J}(\theta_i). \tag{16}$$

For optimizing the objective function, the gradient of the function should be calculated with respect to θ_i as follows:

$$\nabla_{\theta_i}\mathcal{J}(\mu_i) = \mathbb{E}_{\mathbf{x},a\sim\mathcal{D}}\big[\nabla_{\theta_i}\mu_i(a_i|o_i)\nabla_{a_i}Q_i^{\mu}(\mathbf{x}, a_1, \ldots a_N)\big] \tag{17}$$

where $\mathbf{x} = (o_1, \ldots, o_N)$, $Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_N)$ is a centralized action-value function, and $\mathcal{D}$ is a replay buffer, respectively. Here, $\mathcal{D}$ contains transition tuples $(\mathbf{x}, a, r, \mathbf{x}')$, where $a = (a_1, \ldots, a_N)$ and $r = (r_1, \ldots, r_N)$. In addition, the centralized action-value function Q_i^{μ} in (17) is updated for minimizing the loss function (18) as follows:

$$\mathcal{L}(\theta_i) = \mathbb{E}_{\mathbf{x},a,r,\mathbf{x}'}\Big[\big(Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_N) - y\big)^2\Big] \tag{18}$$

$$\Psi_{i,k}^{S,C}(t) = \frac{h_i^k(t)f_i^j(t)p_{i,j}^s(t) + h_i^k(t)f_i^o(t)p_{i,j}^c(t)}{\sum_v^{\mathcal{B}_\alpha - \{b_k\}} h_i^v(t)f_i^j(t)p_{j,i}^s(t) + \sum_v^{\mathcal{B}_\beta - \{b_v\}} h_i^k(t)f_i^o(t)p_{j,i}^c(t) + WN_0} \tag{7}$$

and

$$y = r_i + \gamma Q_i^{\mu'}\left(\mathbf{x}', a'_1, \ldots, a'_N\right)|_{a'_j=\mu'_j(o_j)} \tag{19}$$

where $\mu' = \{\mu_{\theta'_1}, \ldots, \mu_{\theta'_N}\}$ stands for the target policies with delayed parameters θ'_i. In addition, our considering MADDPG is a *actor–critic*-based algorithm where the *actor* takes the role of sequential decision making over time slots while the *critic* evaluates the behaviors of the *actor*. Each IoUT device agent consists of the *actor* and *critic* with the behavior network and the target network. The *actor* updates the behavior policy network and periodically updates the target policy network by utilizing gradient ascent updates on $\mathcal{J}(\theta)$ with (17). In a similar way, the *critic* controls the behaviors of Q-function and periodically updates the target Q-function in a way that minimizes the loss function in (18). The IoUT devices have such an *actor* and *critic* to optimize their own policies to behave cooperatively, while they update their *critic*'s Q-functions to reasonably evaluate the actions. For more details, the optimization objective of such PG approach is for updating θ of the target network, which determines how the IoUT devices actually make actions. The value of DNN of the target network of the *actor* is fixed for a number of iterations, while the weights of DNN of the behavior network of the *actor* are updated.

That is, the multiagents in IoUT-Net observe their own local information and aim to act for maximizing their expected total return. They can stably update the policy parameter θ based on the local information and interactions between other IoUT devices and IoUT-Net environment. In other words, the environment is stationary even though the policies change. Suppose that P stands for the state transition probability, i.e.,

$$\begin{aligned} &P\left(s'|s, a_1, \ldots, a_N, \mu_1, \ldots, \mu_N\right) = P\left(s'|s, a_1, \ldots, a_N\right) \\ &= P\left(s'|s, a_1, \ldots, a_N, \mu'_1, \ldots, \mu'_N\right) \quad \forall \mu_i \neq \mu'_i. \end{aligned} \tag{20}$$

Therefore, the state transition probability from s to s' of each IoUT device is same even though the behavior policy and target policy are mutually different.

As explained before, reinforcement learning-based algorithms can be formulated by defining state space, action space, and reward structure, as follows.

1) *State Space:* The state space of each IoUT device in IoUT-Net is defined with twofold, i.e., QoS requirement satisfaction and accumulative DL throughput variation. The state of the ith IoUT device in terms of QoS, i.e., $s_i^{\text{qos}}(t)$, is set as follows:

$$s_i^{\text{qos}}(t) = \begin{cases} 1, & \text{if } \Psi_{i,k}^{C,M}(t) \geq \chi \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

In addition, the DL throughput of the current time slot is compared to the previous one to decide $s_i^{\text{dl}}(t)$, where $s_i^{\text{dl}}(t) = 1$ if the DL throughput of the current time slot is higher than the previous one [otherwise, $s_i^{\text{dl}}(t) = 0$]. Therefore, the state space of IoUT devices $s(t)$ can be defined as follows:

$$s(t) = (s_1(t), \ldots, s_N(t)) \tag{22}$$

$$= \left\{\left(s_1^{\text{qos}}(t), s_1^{\text{dl}}(t)\right), \ldots, \left(s_N^{\text{qos}}(t), s_N^{\text{dl}}(t)\right)\right\}. \tag{23}$$

Algorithm 1: Proposed MADDPG Algorithm for JCARA

Initialize the weights of *actor* and *critic* networks
Initialize a random process $\mathcal{N}$ for exploration of action
Receive the initial state $\mathbf{x}$
for *t = 1 to* $\mathcal{E}$ **do**
  ▷ Each IoUT device selects a cell association and resource utilization action $a_i = \mu_{\theta_i}(o_i) + \mathcal{N}_t$ based on the exploration and policy
  ▷ Execute actions $a = (a_1(t), \cdots, a_N(t))$
  ▷ Observe rewards $\mathcal{R}(t)$ and new state $\mathbf{x}'$
  ▷ Store $(\mathbf{x}, a, \mathcal{R}(t), \mathbf{x}')$ in $\mathcal{D}$
  ▷ $\mathbf{x} \leftarrow \mathbf{x}'$
  for *agent i = 1 to N* **do**
    ▷ Sample a random minibatch of $\mathcal{V}$ samples $(\mathbf{x}^j, a^j, \mathcal{R}^j, \mathbf{x}'^j)$ from $\mathcal{D}$
    ▷ Set $y^j = \mathcal{R}_i^j + \gamma Q_i^{\mu'}(\mathbf{x}'^j, a'_1, \cdots, a'_N)|_{a'_k=\mu'_k(o'_k)}$
    ▷ Update behavior critic by minimizing the loss:
      $\mathcal{L}(\theta_i) = \frac{1}{\mathcal{V}} \sum_j (y^j - Q_i^{\mu}(\mathbf{x}^j, a_1^j, \cdots, a_N^j))^2$
    ▷ Update behavior actor using the sampled PG:
      $\nabla_{\theta_i} \mathcal{J} \approx$
      $\frac{1}{\mathcal{V}} \sum_j \nabla_{\theta_i} \mu_i(o_i^j) \nabla_{a_i} Q_i^{\mu}(\mathbf{x}^j, a_1^j, \cdots, a_i, \cdots, a_N^j)|_{a_i=\mu_i(o_i^j)}$
  end
  ▷ Update the *target* network parameters of each IoUT device: $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$
end

2) *Action Space:* Each IoUT device decides actions to choose for every time slot. First of all, it should decide what kind of BS should be associated with this IoUT device between OBS/SLBS and SLB. In addition, it chooses which channels should be utilized for wireless communications. Thus, the action space of IoUT devices can be defined as follows:

$$a(t) = (a_1(t), \ldots, a_N(t)) \tag{24}$$

$$= \left\{(l_1^k(t), f_1^o(t), f_1^j(t)), \ldots, \left(l_N^k(t), f_N^o(t), f_N^j(t)\right)\right\} \tag{25}$$

where $\forall i \in [1, N]$ $\forall o \in [K_a + K_i + 1, K]$, and $\forall j \in [1, K_a + K_i]$. As the number of SLBs increases, the size of action spaces exponentially increases, thus it is intractable to solve the JCARA problem with traditional solution approaches.

3) *Reward Structure:* The immediate reward of the ith IoUT device can be denoted as $\mathcal{R}_i(t)$ and then it can be calculated based on the interaction between IoUT devices and the IoUT-Net environment, i.e., $((s_1(t), a_1(t)), \ldots, (s_N(t), a_N(t)))$. Thus, $\mathcal{R}_i(t) = \Lambda_i(t) - \Upsilon_i$. Note that Υ_i stands for the failure penalty of each ith IoUT device, which is took into account for the computation of the reward when: a) the IoUT device fails to make an association with a BS or b) it cannot access any wireless spectrum.

C. Algorithm Design

The details of the proposed MADDPG algorithm to solve the JCARA problem are presented in this section. The pseudocode of the proposed MADDPG algorithm is illustrated in Algorithm 1 and its detailed description is as follows.

1) First, the parameters of the *actor* and *critic* networks which activate and evaluate the actions of IoUT devices, are initialized (lines 1–3).
2) Next, for $\mathcal{E}$ iterations, the following procedures are conducted in order to update the target network parameters of IoUT devices. Given the initial state $\mathbf{x}$, each IoUT device determines its action based on exploration noise and its own policy (line 5). After the actions taking of each IoUT device are conducted, the actions are activated by the IoUT device (line 6). Next, the IoUT-Net interacts with the IoUT devices and then it returns corresponding rewards and next states (line 7). Then, each IoUT device observes the state transition pairs and stores in its own replay buffer $\mathcal{D}$ which contains the experiences of the IoUT device (line 8). After that, the episodic state $\mathbf{x}$ is updated to the next $\mathbf{x}'$ (line 9).
3) Throughout the entire episode, each IoUT device conducts the following procedures to update its *actor* and *critic* networks. First, each ith IoUT device samples the random minibatch of $\mathcal{V}$ samples among $\mathcal{D}$ (line 11). Notice that the superscript j stands for the approximation of all IoUT devices except the ith IoUT device. Then, the target value of Q-function y^j is set (line 12). By minimizing the difference between y^j and $Q_i^{\mu}(\mathbf{x}^j, a_1^j, \ldots, a_N^j)$ among $\mathcal{V}$ samples, θ of the behavior *critic* is updated (line 13). In a similar way, θ of $\mathcal{J}$ of the behavior *actor* is updated with the gradient for optimizing the policy parameter θ (line 14). Notice that the policy update is based on gradient ascent computation.
4) Finally, when all IoUT devices eventually update their behavior networks, the target network parameters are updated under the concept of *soft update* where the *soft* (*target*) *update* is for achieving learning stability by restricting the target value updating speed (line 16).

V. Performance Evaluation

This section consists of simulation settings (refer to Section V-A) and evaluation results (refer to Section V-B).

A. Settings

This section provides the performance evaluation setting of our proposed MADDPG algorithm for solving the JCARA problem. As a simulation environment, there are 1 OBS, 10 SLBSs, 50 SLBs, and 100 IoUT devices in IoUT-Net. Each of the above three kinds of BSs has different cell coverage regions, i.e., 3000, 500, and 100 m, respectively. In addition, the transmit power values of OBS, SLB, and SLB are 40, 35, and 20 dBm, respectively. We set $\mathcal{X}$ (the number of the orthogonal channels which are shared by OBS and SLBSs) and $\mathcal{Y}$ (the number of UHF channels in SLBs) are 30 and 5, respectively.

The channel bandwidth of OBS and SLB is set to 180 kHz and the DL center frequency is set to 2 GHz. Meanwhile, the channel bandwidth of SLB is set to 800 MHz and the DL center frequency is set to 28 GHz. The path loss of OBS and SLB can be calculated by $34 + 40\log(d)$ and the path loss of SLB can be also obtained by $37 + 30\log(d)$. The failure cost Υ_i of $\mathcal{R}_i(t)$ is set to 10^{-2} and the base line of QoS χ is set to 7 dBm. The noise power N_0 is set to -175 dBm/Hz and ρ is set to 10^{-3}. The proposed MADDPG deep reinforcement learning model has two-layered fully connected neural network architectures with 64 and 32 neurons. Note that Table I presents the hyperparameters of the model.

TABLE I
Hyperparameters of the MADDPG Model

Parameter	Value
Total episode $\mathcal{E}$	500
Time step T	100
Minibatch size	64
Discounting factor γ	0.95
Initial epsilon	0.9
Learning rate δ	0.05
Size of $\mathcal{D}$	1000
Optimizer	AdamOptimizer
Activation function	ReLU

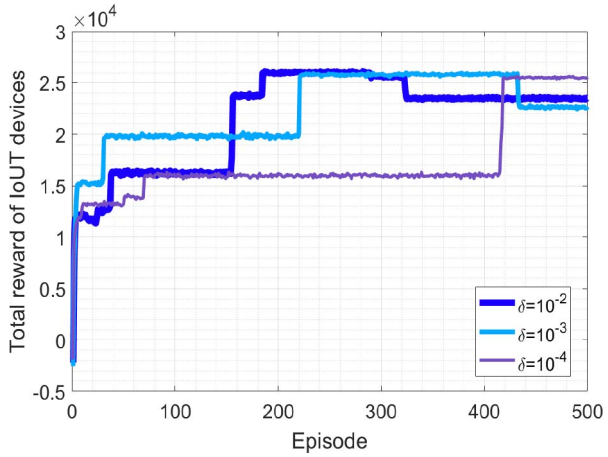


Fig. 2. Convergence rate comparison with learning rate δ.

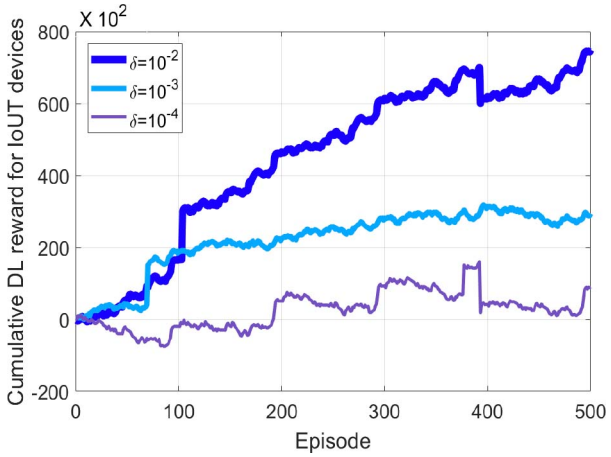


Fig. 3. Cumulative DL throughput reward for IoUT devices.

B. Evaluation Results

First, the learning curves of the reward for our considering the JCARA problem and the cumulative reward of DL using the MADDPG are as plotted in Figs. 2 and 3. Fig. 2

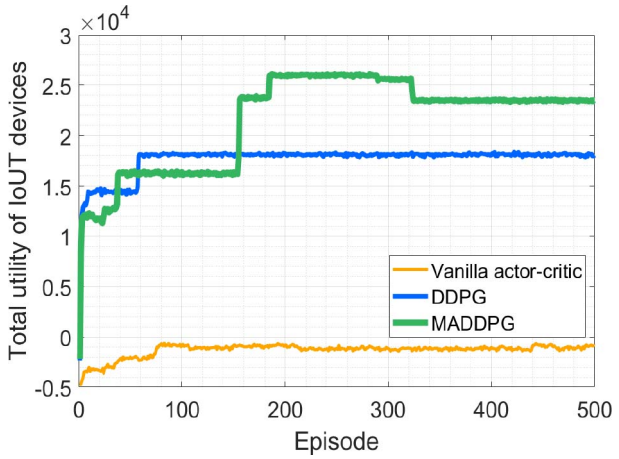


Fig. 4. Performance comparison on the JCARA problem.

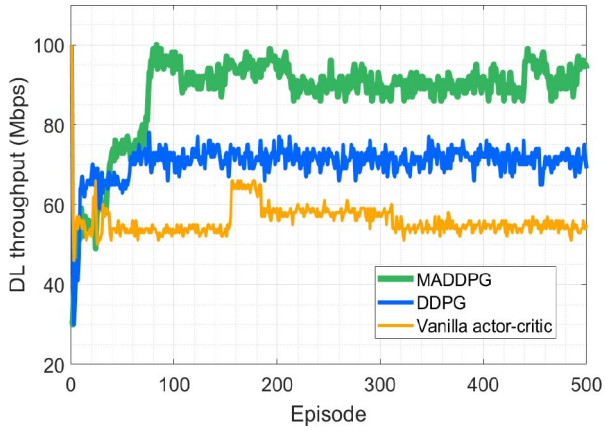


Fig. 5. DL throughput of each algorithm in IoUT-Net.

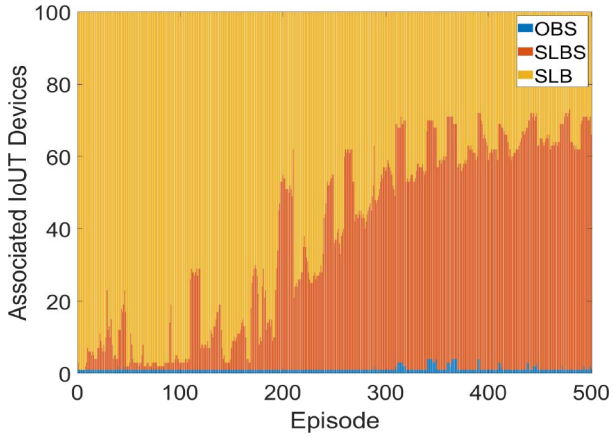


Fig. 6. Number of associated IoUT devices of each BS.

shows that each learning model has the slightly different convergence points depending on learning rates. In Fig. 2, it can be observed that the required episode to get the converged performance is decreased as the learning rate decreases. In addition, there are noticeable differences in convergence points due to the learning rates as shown in Fig. 3. The values of Fig. 3 are not considered the power cost of IoUT devices due to the fact that we assume that each IoUT has a spliced solar panel for self-charging.

Next, the performance of our proposed MADDPG algorithm is compared with other PG algorithms, i.e., vanilla actor–critic and DDPG (baseline algorithm among PG algorithms). The actor–critic algorithm does not use the concept of replay buffer; and it trains the models with states, actions, rewards, and next states, those are obtained in each step. Here, the *actor* network approximates its own policy and the *critic* network approximates the Q function. The vanilla actor–critic algorithm is a standard actor–critic among various actor–critic algorithms. This vanilla actor–critic uses a standard gradient for PG procedures [39]. Due to the fact that this actor–critic algorithm is the foundation of the other PG-based reinforcement learning algorithms, this can be used for baseline performance estimation. As presented in Fig. 4, the vanilla actor–critic approach is almost not good enough to solve the given JCARA problem, thus each IoUT device greedily accesses wireless spectrum and eventually suffers from wireless collisions. On the other hand, the DDPG and our proposed MADDPG algorithms show much better performance. However, the DDPG suffers from the nonstationary problem, therefore, the total reward of IoUT devices trained by the proposed MADDPG algorithm is eventually higher than that of the DDPG.

Finally, the performance of the average DL throughput of IoUT devices in IoUT-Net is presented in Fig. 5. As presented in Figs. 4 and 5, the proposed MADDPG algorithm learned the policies of IoUT devices in a way that cooperatively associates with BSs and utilizes wireless spectrum (the high total reward of IoUT devices as shown Fig. 4 and the high DL throughput as shown Fig. 5). Although the DDPG shows a little bit lower performance than that of our proposed MADDPG algorithm, it is still shown to learn cooperative policies as presented in Fig. 4. However, the vanilla actor–critic approach presents the lowest total reward of IoUT devices as shown in Fig. 4 and the DL throughput as shown in Fig. 5. This means the vanilla actor–critic approach learns selfish association and resource utilization policies in the nonstationary environment. In conclusion, our proposed MADDPG algorithm is successful to learn cooperative policies in order to solve the JCARA problem in our considering IoUT-Net scenarios.

In Figs. 6 and 7, the numbers of connected IoUT devices and selected channels are compared. In Fig. 6, 99 IoUT devices were initially associated with SLB; and as our proposed MADDPG-based algorithm works over time, the association will be distributed to be connected with SLBS and SLB. The numbers of IoUT devices that were associated with SLBS and LSB were initially 2 and 97, respectively; and then eventually, the numbers become 65 and 34, respectively. Each IoUT device updates its own policy considering the actions of other IoUT devices. In Fig. 7, the numbers of utilized channels for SLBS and SLB were initially 3 and 251. As our proposed MADDPG-based algorithm works over time, the numbers become 105 and 71, i.e., the use of channels is distributed. In means that most of the channel allocations belong to SLB at first, but eventually, the communication resources allocation becomes distributed for SLBS and SLB.

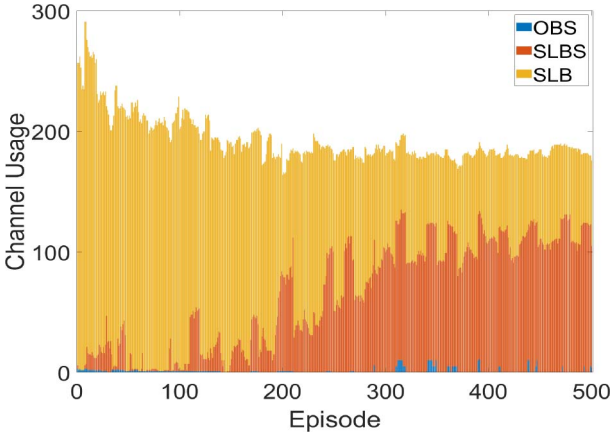


Fig. 7. Channel usage of each BS.

VI. Conclusion

This article proposes a multiagent DRL approach via MADDPG for solving JCARA in IoUT-Net-based smart ocean FL scenarios. In our considering scenarios, each IoUT device conducts distributed deep learning training with its own data, i.e., FL, and the results are aggregated at a centralized HPC machine in smart ocean BSs. Then, the centralized machine computes global optimal solutions and then distributes the results to its associated IoUT devices. Therefore, the DL throughput performance enhancement is the main goal of the proposed algorithm in this article. According to the fact that our considering the JCARA problem is NP-hard, traditional approaches are limited to solve the problem. Therefore, this article utilizes an MADDPG-based algorithm and it computes efficient solutions with a small number of iterations (i.e., low-complexity deep learning computation). Based on data-intensive simulation results, it is verified that the proposed MADDPG-based multiagent deep learning algorithm achieves better DL throughput performance compared to the other reinforcement learning-based JCARA methods.

As future research directions, we will implement and deploy this proposed MADDPG-based multiagent DRL algorithm in real-world scenarios under the consideration of specific FL data set and applications. Furthermore, data-intensive measurement-based performance evaluation can be done with various measurement metrics.

References

[1] Q. Guan, F. Ji, Y. Liu, H. Yu, and W. Chen, "Distance-vector-based opportunistic routing for underwater acoustic sensor networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3831–3839, Apr. 2019.

[2] N. Saeed, M. Alouini, and T. Y. Al-Naffouri, "Towards the Internet for X-things: New possibilities for underwater, underground, and outer space exploration," Apr. 2019. [Online]. Available: arXiv:1903.11996.

[3] S. Ahn, J. Kim, E. Lim, W. Choi, A. Mohaisen, and S. Kang, "ShmCaffe: A distributed deep learning platform with shared memory buffer for HPC architecture," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Vienna, Austria, Jul. 2018, pp. 1118–1128.

[4] H. T. Nguyen, N. C. Luong, J. Zhao, C. Yuen, and D. Niyato, "Resource allocation in mobility-aware federated learning networks: A deep reinforcement learning approach," Oct. 2019. [Online]. Available: arXiv:1910.09172.

[5] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi, "On the design of a decentralized and multiauthority access control scheme in federated and cloud-assisted cyber-physical systems," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5190–5204, Dec. 2018.

[6] J. Jeon, J. Kim, J. Kim, K. Kim, A. Mohaisen, and J.-K. Kim, "Privacy-preserving deep learning computation for geo-distributed medical big-data platforms," in *Proc. IEEE Int. Conf. Depend. Syst. Netw. (DSN)*, Portland, OR, USA, Jun. 2019, pp. 3–4.

[7] J. Jeon, D. Kim, and J. Kim, "Cyclic parameter sharing for privacy-preserving distributed deep learning platforms," in *Proc. IEEE Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Okinawa, Japan, Feb. 2019, pp. 435–437.

[8] J. Konečný, H. B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," Nov. 2015. [Online]. Available: arXiv:1511.03575.

[9] M. Jia, Z. Yin, Q. Guo, G. Liu, and X. Gu, "Downlink design for spectrum efficient IoT network," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3397–3404, Oct. 2018.

[10] W. Yang *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," Sep. 2019. [Online]. Available: arXiv:1909.11875v1.

[11] S. Ji, S. Pan, G. Long, X. Li, J. Jiang, and Z. Huang, "Learning private neural language modeling with attentive aggregation," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8.

[12] D. Kwon and J. Kim, "Multi-agent deep reinforcement learning for cooperative connected vehicles," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[13] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor–critic for mixed cooperative-competitive environments," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 6379–6390.

[14] S. A. Busari, K. M. S. Huq, G. Felfel, and J. Rodriguez, "Adaptive resource allocation for energy-efficient millimeter-wave massive MIMO networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.

[15] Z. Shi, Y. Wang, L. Huang, and T. Wang, "Dynamic resource allocation in mmWave unified access and backhaul network," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Hong Kong, Aug./Sep. 2015, pp. 2260–2264.

[16] Y. Liu, L. Lu, G. Y. Li, Q. Cui, and W. Han, "Joint user association and spectrum allocation for small cell networks with wireless backhauls," *IEEE Wireless Commun. Lett.*, vol. 5, no. 5, pp. 496–499, Jul. 2016.

[17] N. Wang, E. Hossain, and V. K. Bhargava, "Joint downlink cell association and bandwidth allocation for wireless backhauling in two-tier HetNets with large-scale antenna arrays," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3251–3268, Jan. 2016.

[18] Q. Kuang, W. Utschick, and A. Dotzler, "Optimal joint user association and resource allocation in heterogeneous networks via sparsity pursuit," Nov. 2015. [Online]. Available: arXiv:1408.5091.

[19] Y. Lin, W. Bao, W. Yu, and B. Liang, "Optimizing user association and spectrum allocation in HetNets: A utility perspective," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 6, pp. 1025–1039, Mar. 2015.

[20] Y. Chen, J. Li, W. Chen, Z. Lin, and B. Vucetic, "Joint user association and resource allocation in the downlink of heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5701–5706, Jul. 2015.

[21] J. Ortín, J. R. Gállego, and M. Canales, "Joint cell selection and resource allocation games with backhaul constraints," *Pervasive Mobile Comput.*, vol. 35, pp. 125–145, Feb. 2017.

[22] T. LeAnh *et al.*, "Matching theory for distributed user association and resource allocation in cognitive femtocell networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8413–8428, Mar. 2017.

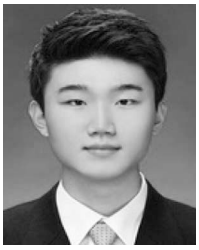
[23] W. Bao and B. Liang, "Structured spectrum allocation and user association in heterogeneous cellular networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr./May 2014, pp. 1069–1077.

[24] Y. He *et al.*, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Sep. 2017.

[25] N. Rupasinghe and I. Güvenç, "Reinforcement learning for licensed-assisted access of LTE in the unlicensed spectrum," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, New Orleans, LA, USA, Mar. 2015, pp. 1279–1284.

[26] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey on deep reinforcement learning," Sep. 2017. [Online]. Available: arXiv:1708.05866.

[27] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.
[28] M. Elsayed and M. Erol–Kantarci, "Deep reinforcement learning for reducing latency in mission critical services," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.
[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[30] C. J. Watkins and P. Dayan, "*Q*-learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, 1992.
[31] J. Konečnỳ, H. B. McMahan, and D. Ramage, "Federated optimization: Distributed machine learning for on-device intelligence," Oct. 2016. [Online]. Available: arXiv:1610.02527.
[32] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported Internet of Things," *IEEE Access*, vol. 7, pp. 69194–69201, 2019.
[33] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.
[34] J. Konečnỳ, H. B. McMahan, F. X. Yu, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," Oct. 2017. [Online]. Available: arXiv:1610.05492.
[35] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," Mar. 2019. [Online]. Available: arXiv:1902.01046.
[36] C. M. G. Gussen, P. S. R. Diniz, M. L. R. Campos, W. A. Martins, F. M. Costa, and J. N. Gois, "A survey of underwater wireless communication technologies," *J. Commun. Inf. Syst.*, vol. 31, no. 1, pp. 242–255, Oct. 2016.
[37] P. Saini, P. R. Singh, and A. Sinha, "Path loss analysis of RF waves for underwater wireless sensor networks," in *Proc. Int. Conf. Comput. Commun. Technol. Smart Nation (IC3TSN)*, New Delhi, India, Oct. 2017, pp. 104–108.
[38] X. Che, I. Wells, G. Dickers, P. Kear, and X. Gong, "Re-evaluation of RF electromagnetic communication in underwater sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 12, pp. 143–151, Dec. 2010.
[39] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor–critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1291–1307, Dec. 2012.



Dohyun Kwon received the B.S. and M.S. degrees in computer science and engineering from Chung-Ang University, Seoul, South Korea, in 2018 and 2020, respectively.

He is currently a Research Engineer with Hyundai Autoever, Seoul. His research focus includes deep reinforcement learning for mobile networks.

Joohyung Jeon received the B.S. degree in computer science and engineering from Chung-Ang University, Seoul, South Korea, in 2018, where he is currently pursuing the M.S. degree.

His research focus includes federated and distributed deep learning multi-GPU platforms.

Soohyun Park received the B.S. degree in computer science and engineering from Chung-Ang University, Seoul, South Korea, in 2019. She is currently pursuing the Ph.D. degree in electrical engineering with Korea University, Seoul.

Her research focuses include deep learning and machine learning algorithms and their applications.

Dr. Park was a recipient of the IEEE Vehicular Technology Society Seoul Chapter Award in 2019.

Joongheon Kim (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science and engineering from Korea University, Seoul, South Korea, in 2004 and 2006, respectively, and the Ph.D. degree in computer science from the University of Southern California (USC), Los Angeles, CA, USA, in 2014.

He was a Research Engineer with LG Electronics, Seoul, from 2006 to 2009, an Intern with InterDigital, San Diego, CA, USA, in 2012, a Systems Engineer with Intel Corporation, Santa Clara, CA, USA, from 2013 to 2016, and an Assistant Professor of computer science and engineering with Chung-Ang University, Seoul, from 2016 to 2019. In 2019, he joined Korea University, where he is currently an Assistant Professor of electrical engineering.

Dr. Kim was a recipient of the Annenberg Graduate Fellowship with his Ph.D. admission from USC in 2009, the Intel Corporation Next Generation and Standards Division Recognition Award in 2015, the Haedong Young Scholar Award by KICS in 2018, the IEEE Vehicular Technology Society Seoul Chapter Award in 2019, the Outstanding Contribution Award by KICS in 2019, the Gold Paper Award from IEEE Seoul Section Student Paper Contest in 2019, and the IEEE Systems Journal Best Paper Award in 2020.

Sungrae Cho (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Korea University, Seoul, South Korea, in 1992 and 1994, respectively, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2002.

He was an Assistant Professor with the Department of Computer Sciences, Georgia Southern University, Statesboro, GA, USA, from 2003 to 2006, and a Senior Member of Technical Staff with the Samsung Advanced Institute of Technology, Kiheung, South Korea, in 2003. He is a Professor and a Dean with the School of Software, Chung-Ang University, Seoul. From 1994 to 1996, he was a Research Staff Member with Electronics and Telecommunications Research Institute, Daejeon, South Korea. From 2012 to 2013, he held a visiting professorship with the National Institute of Standards and Technology, Gaithersburg, MD, USA.

Prof. Cho was an Editor of *Ad Hoc Networks* (Elsevier) from 2012 to 2017.