Multi-Agent Meta-Reinforcement Learning for Self-Powered and Sustainable Edge Computing Systems

Md. Shirajum Munir, *Graduate Student Member, IEEE*, Nguyen H. Tran, *Senior Member, IEEE*, Walid Saad, *Fellow, IEEE*, and Choong Seon Hong, *Senior Member, IEEE*

***Abstract*—The stringent requirements of mobile edge computing (MEC) applications and functions fathom the high capacity and dense deployment of MEC hosts to the upcoming wireless networks. However, operating such high capacity MEC hosts can significantly increase energy consumption. Thus, a base station (BS) unit can act as a self-powered BS. In this article, an effective energy dispatch mechanism for self-powered wireless networks with edge computing capabilities is studied. First, a two-stage linear stochastic programming problem is formulated with the goal of minimizing the total energy consumption cost of the system while fulfilling the energy demand. Second, a semi-distributed data-driven solution is proposed by developing a novel multi-agent meta-reinforcement learning (MAMRL) framework to solve the formulated problem. In particular, each BS plays the role of a local agent that explores a Markovian behavior for both energy consumption and generation while each BS transfers time-varying features to a meta-agent. Sequentially, the meta-agent optimizes (i.e., exploits) the energy dispatch decision by accepting only the observations from each local agent with its own state information. Meanwhile, each BS agent estimates its own energy dispatch policy by applying the learned parameters from meta-agent. Finally, the proposed MAMRL framework is benchmarked by analyzing deterministic, asymmetric, and stochastic environments in terms of non-renewable energy usages, energy cost, and accuracy. Experimental results show that the proposed MAMRL model can reduce up to 11% non-renewable energy usage and by 22.4% the energy cost (with 95.8% prediction accuracy), compared to other baseline methods.**

***Index Terms*—Mobile edge computing (MEC), stochastic optimization, meta-reinforcement learning, self-powered, demand response.**

I. Introduction

NEXT-GENERATION wireless networks are expected to significantly rely on *edge* applications and functions that include edge computing and edge artificial intelligence (edge AI) [1]–[7]. To successfully support such edge services within a wireless network with mobile edge computing (MEC) capabilities, energy management (i.e., demand and supply) is one of the most critical design challenges. In particular, it is imperative to equip next-generation wireless networks with alternative energy sources, such as renewable energy, in order to provide extremely reliable energy dispatch with less energy consumption cost [8]–[15]. An efficient energy dispatch design requires energy sustainability, which not only saves energy consumption cost, but also fulfills the energy demand of the edge computing by enabling its own renewable energy sources. Specifically, sustainable energy is the practice of seamless energy flow to the MEC system that emerges to meet the energy demand without compromising the ability of future energy generation. Furthermore, to ensure a sustainable MEC operation, the retrogressive penetration of uncertainty for energy consumption and generation is essential. A summary of the challenges that are solved by the literature to enable renewable energy sources for the wireless network is presented in Table I.

To provide sustainable edge computing for next-generation wireless systems, each base station (BS) with MEC capabilities unit can be equipped with renewable energy sources. Thus, the energy source of such a BS unit not only relies solely on the power grid, but also on the equipped renewable energy sources. In particular, in a self-powered network, wireless BSs with MEC capabilities is equipped with its own renewable energy sources that can generate renewable energy, consume, store, and share energy with other BS units.

Delivering seamless energy flow with a low energy consumption cost in a self-powered wireless network with MEC capabilities can lead to uncertainty in both energy demand and generation. In particular, the randomness of the energy demand is induced by the uncertain resources (i.e., computation and communication) request by the edge services and

Manuscript received September 14, 2020; revised February 3, 2021; accepted February 4, 2021. Date of publication February 9, 2021; date of current version September 9, 2021. This work was partially supported by the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) under Grant 2020R1A4A1018607; in part by the Institute of Information & communications Technology Planning & Evaluation (IITP) funded by the Korea government (MSIT) under Grant 2019-0-01287; and in part by Evolvable Deep Learning Model Generation Platform for Edge Computing. The associate editor coordinating the review of this article and approving it for publication was S. Latre. *(Corresponding author: Choong Seon Hong.)*

Md. Shirajum Munir and Choong Seon Hong are with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea (e-mail: munir@khu.ac.kr; cshong@khu.ac.kr).

Nguyen H. Tran is with the School of Computer Science, University of Sydney, Sydney, NSW 2006, Australia (e-mail: nguyen.tran@sydney.edu.au).

Walid Saad is with the Wireless@VT Group, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea (e-mail: walids@vt.edu).

Digital Object Identifier 10.1109/TNSM.2021.3057960

TABLE I
SUMMARY OF THE CHALLENGES THAT ARE SOLVED BY THE LITERATURE FOR ENABLING RENEWABLE ENERGY SOURCES IN THE WIRELESS NETWORK

Ref.	Energy sources	MEC capabilities	Non-i.i.d. dataset	Energy dispatch	Energy cost	Remarks
[8]	Renewable	No	No	No	No	Activation and deactivation of BSs in a self-powered network
[9]	Hybrid energy	No	No	No	No	User scheduling and network resource management
[10]	Hybrid energy	Yes	No	No	No	Load balancing between the centralized cloud and edge server
[11]	Microgrid	Yes	No	Yes	No	MEC task assignment and energy demand-response (DR) management
[12]	Microgrid	Yes	No	Yes	No	Risk-sensitive energy profiling for microgrid-powered MEC network
[13]	Renewable	No	No	Yes	No	Energy load balancing among the SBSs with a microgrid
[14]	Smart grid enabled hybrid energy	No	No	Yes	No	Joint network resource allocation and energy sharing among the BSs
[15]	Hybrid energy	No	No	Yes	No	Overall system architecture for edge computing and renewable energy resources
This work	Smart grid enabled self-powered renewable energy	Yes	Yes	Yes	Yes	An effective energy dispatch mechanism for self-powered wireless networks with edge computing capabilities

applications. Meanwhile, the energy generation of a renewable source (i.e., a solar panel) at each self-powered BS unit varies on the time of a day. In other words, the pattern of energy demand and generation will differ from one self-powered BS unit to another. Thus, such fluctuating energy demand and generation pattern induces a non-independent and identically distributed (non-i.i.d.) of energy dispatch at each BS over time. To overcome this non-i.i.d. energy demand and generation, characterizing the expected amount of uncertainty is crucial to ensure a seamless energy flow to the self-powered wireless network. As such, when designing self-powered wireless networks, it is necessary to take into account this uncertainty in the energy patterns.

A. Related Works

The problem of energy management for MEC-enabled wireless networks has been studied in [16]–[22] (summary in Table II). In [16], the authors proposed a joint mechanism for radio resource management and users task offloading with the goal of minimizing the long-term power consumption for both mobile devices and the MEC server. The authors in [17] proposed a heuristic to solve the joint problem of computational resource allocation, uplink transmission power, and user task offloading problem. The work in [18] studied the tradeoff between communication and computation for a MEC system and the authors proposed a MEC server CPU scaling mechanism for reducing the energy consumption. Further, the work in [19] proposed an energy-aware mobility management scheme for MEC in ultra-dense networks, and they addressed the problem using Lyapunov optimization and multi-armed bandits. Recently, the authors in [21] proposed a distributed power control scheme for a small cell network by using the concept of a multi-agent calibrate learning. Further, the authors in [22] studied the problem of energy storage and energy harvesting (EH) for a wireless network using deviation theory and Markov processes. However, all of these existing works assume that the consumed energy is available from the energy utility source to the wireless network system [16]–[22]. Since the assumed models are often focused on energy management and user task offloading on network resource allocations, the random demand for computational (e.g., CPU computation, memory, etc.) and communication requirements of the edge applications and services are not considered. In fact, even if enough energy supply is available, the energy cost related to network operation can be significant because of the usage of non-renewable (e.g., coal, petroleum, natural gas). Indeed, it is necessary to include renewable energy sources towards the next-generation wireless networking infrastructure.

Recently, some of the challenges of renewable energy powered wireless networks have been studied in [8]–[14], [23]. In [8], the authors proposed an online optimization framework to analyze the activation and deactivation of BSs in a self-powered network. In [9], proposed a hybrid power source infrastructure to support heterogeneous networks (HetNets), a model-free deep reinforcement learning (RL) mechanism was proposed for user scheduling and network resource management. In [10], the authors developed an RL scheme for edge resource management while incorporating renewable energy in the edge network. In particular, the goal of [10] is to minimize a long-term system cost by load balancing between the centralized cloud and edge server. The authors in [11] introduced a microgrid enabled edge computing system. A joint optimization problem is studied for MEC task assignment and energy demand-response (DR) management. The authors in [11] developed a model-based deep RL framework to tackle the joint problem. In [12], the authors proposed a risk-sensitive energy profiling for microgrid-powered MEC network to ensure a sustainable energy supply for green edge computing by capturing the conditional value at risk (CVaR) tail distribution of the energy shortfall. The authors in [12] proposed a multi-agent RL system to solve the energy scheduling problem. In [13], the authors proposed a self-sustainable mobile networks, using graph-based approach for intelligent energy management with a microgrid. The authors in [14] proposed a smart grid-enabled wireless network and minimized grid energy consumption by applying energy sharing among the BSs. Furthermore, in [23], the authors

TABLE II
SUMMARY OF THE RELATED WORKS [16]–[28]

Ref.	Contributions	Method	Limitation
[16]	Radio resource management and users task offloading	Optimization	Usage of non-renewable, deterministic environment
[17]	Computational resource allocation, uplink transmission power, and user task offloading	Heuristic	Usage of non-renewable, energy dispatch, performance guarantee
[18]	MEC server CPU scaling mechanism for reducing the energy consumption	Optimization	Usage of non-renewable, energy dispatch
[19]	Energy-aware mobility management scheme for MEC	Lyapunov and multi-armed bandits	Energy dispatch, i.i.d. energy demand-response
[20]	Energy efficient green-IoT network	Heuristic	Edge computing, Energy dispatch, deterministic environment
[21]	Distributed power control scheme for a small cell network	Multi-agent calibrate learning	Usage of non-renewable, energy dispatch
[22]	Energy storage and energy harvesting (EH) for a wireless network	Deviation theory and Markov processes	MEC capabilities, i.i.d. energy demand-response
[23]	Non-coordinated energy shedding and mis-aligned incentives for mixed-use building	Auction theory	MEC capabilities, i.i.d. energy demand-response
[24]	Tradeoff between effectiveness and available amounts of training data	Deep meta-RL	Stochastic environment and a multi-agent scenario
[25]	Controling the meta-parameter in both static and dynamic environments	SGD-based meta-parameter learning	Single-agent, same environment
[26]	Learning to learn mechanism with the recurrent neural networks	Generalized transfer learning	Deterministic environment, single-agent
[27]	Asynchronous multi-agent RL framework	One-step Q-learning, one-step Sarsa, and n-step Q-learning	Deterministic environment
[28]	General-purpose multi-agent scheme	Extension of the actor-critic policy gradient	Same environment for all of the local actors

addressed challenges of non-coordinated energy shedding and mis-aligned incentives for mixed-use building (i.e., buildings and data centers) using auction theory to reduce energy usage. However, these works [9]–[14], [23] do not investigate the problem of energy dispatch nor do they account for the energy cost of MEC-enabled, self-powered networks when the demand and generation of each self-powered BS are non-i.i.d.. Dealing with non-i.i.d. energy demand and generation among self-powered BSs is challenging due to the intrinsic energy requirements of each BS evolve the uncertainty. In order to overcome this unique *energy dispatch* challenge, we propose to develop a *multi-agent meta-reinforcement learning framework* that can adapt new uncertain environment without considering the entire past experience.

Some interesting problems related to meta-RL and multi-agent deep RL are studied in [24]–[28] (summary in Table II). In [24], the authors focused on studying the challenges of the tradeoff between effectiveness and available amounts of training data for a deep-RL based learning system. To this end, the authors in [24] tackled those challenges by exploring a deep meta-reinforcement learning architecture. This learning architecture comprises of two learning systems: 1) lower-level system that can learn each new task very quickly, 2) higher-level system is responsible to improve the performance of each lower-level system task. In particular, this learning mechanism is involved with one lower-level system that can learn relatively quickly as compared with a higher-level system. This lower-level system can adapt to a new task while a higher-level system performs fine-tuning so as to improve the performance of the lower-level system. In particular, in deep meta-reinforcement learning, a lower-level system quantifies a reward based on the desired action and feeds back that reward to a higher-level system to tune the weights of a recurrent network. However, the authors in [24] do not consider a stochastic environment nor do they extend their work for a multi-agent scenario. The authors in [25] proposed a stochastic gradient-based meta-parameter learning scheme for tuning reinforcement learning parameters to the physical environmental dynamics. Particularly, the experiment in [25] performed in both animal and robot environments, where an animal must recognize food before it starves and a robot must recharge before the battery is empty. Thus, the proposed scheme can effectively find meta-parameter values and controls the meta-parameter in both static and dynamic environments. In [26], the authors investigated a learning to learn (i.e., meta-learning) mechanism with the recurrent neural networks, where the meta-learning problem was designed as a generalized transfer learning scheme. In particular, the authors in [26] considered a parametrized optimizer that can transfer the neural network parameters update to an optimizee. Meanwhile, the optimizee can determine the gradients without relying on the optimizer parameters. Moreover, the optimizee sends the error to the optimizer, and updates its own parameters based on the transferred parameters. This mechanism allows an agent to learn new tasks for a similar structure. An asynchronous multi-agent RL framework was studied in [27], where the authors investigated how parallel actor learners of asynchronous advantage actor-critic (A3C) can achieve better stability during the neural network training compared to asynchronous RL schemes. Such schemes include asynchronous one-step Q-learning, one-step Sarsa, and n-step Q-learning. The authors in [28] proposed a general-purpose multi-agent scheme by adopting the framework of centralized training with decentralized execution.

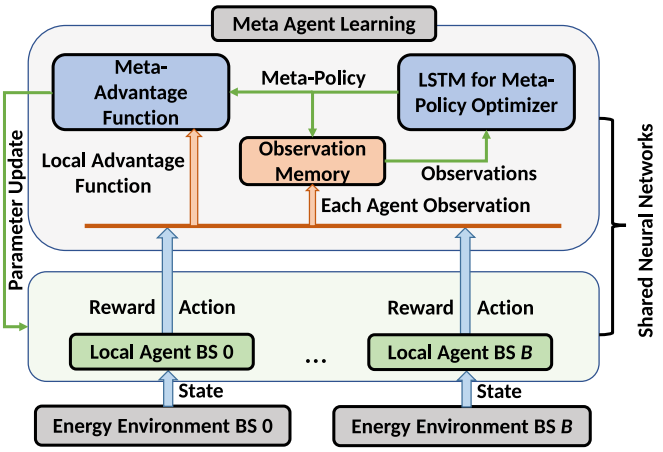


Fig. 1. Multi-agent meta-reinforcement learning framework of self-powered energy dispatch for sustainable edge computing.

In particular, the authors in [28] proposed an extension of the actor-critic policy gradient mechanism by modifying the role of the critic. This critic is augmented with an additional policy information from the other actors (agents). Sequentially, each local actor executes in a decentralized manner and sends its own policy to the centralized critic for further investigation. However, the environment (i.e., state information) of this model remains the same for all of the local actors while in our setting the environment of each BS agent is deferent from others based on its own energy demand and generation. Moreover, the works in [24]–[28], do not consider a multi-agent environment in which the policy of each agent relies on its own state information. In particular, such state information belongs to a non-i.i.d. learning environment when environmental dynamics become distinct among the agents.

B. Contributions

The main contribution of this article is a novel energy management framework for next-generation MEC in self-powered wireless network that is reliable against extreme uncertain energy demand and generation. We formulate a two-stage stochastic energy cost minimization problem that can balance renewable, non-renewable, and storage energy without knowing the actual demand. In fact, the formulated problem also investigates the realization of renewable energy generation after receiving the uncertain energy demand from the MEC applications and service requests. To solve this problem, we propose a multi-agent meta-reinforcement learning (MAMRL) framework that dynamically observes the non-i.i.d. behavior of time-varying features in both energy demand and generation at each BS and, then transfers those observations to obtain an energy dispatch decision and execute the energy dispatch policy to the self-powered BS. Fig. 1 illustrates how we propose to dispatch energy to ensure sustainable edge computing over a self-powered network using MAMRL framework. As we can see, each BS that includes small cell base stations (SBSs) and a macro base station (MBS) will act as a local agent and transfer their own decision (reward and action) to the meta-agent. Then, the meta-agent accumulates all of the non-i.i.d. observations from each local agent (i.e., SBSs and MBS) and optimizes the energy dispatch policy. The proposed MAMRL framework then provides feedback to each BS agent for exploring efficiently that acquire the right decision more quickly. Thus, the proposed MAMRL framework ensures autonomous decision making under an uncertain and unknown environment. Our key contributions include:

- We formulate a self-powered energy dispatch problem for MEC-supported wireless network, in which the objective is to minimize the total energy consumption cost of network while considering the uncertainty of both energy consumption and generation. The formulated problem is, thus, a two-stage linear stochastic programming. In particular, the first stage makes a decision when energy demand is unknown, and the second stage discretizes the realization of renewable energy generation after knowing energy demand of the network.
- To solve the formulated problem, we propose a new multi-agent meta-reinforcement learning framework by considering the skill transfer mechanism [24], [25] between each local agent (i.e., self-powered BS) and meta-agent. In this MAMRL scheme, each local agent explores its own energy dispatch decision using Markovian properties for capturing the time-varying features of both energy demand and generation. Meanwhile, the meta-agent evaluates (exploits) that decision for each local agent and optimizes the energy dispatch decision. In particular, we design a long short-term memory (LSTM) as a meta-agent (i.e., run at MBS) that is capable of avoiding the incompetent decision from each local agent and learns the right features more quickly by maintaining its own state information.
- We develop the proposed MAMRL energy dispatch framework in a semi-distributed manner. Each local agent (i.e., self-powered BS) estimates its own energy dispatch decision using local energy data (i.e., demand and generation), and provides observations to the meta-agent individually. Consequently, the meta-agent optimizes the decision centrally and assists the local agent toward a globally optimized decision. Thus, this approach not only reduces the computational complexity and communication overhead but it also mitigates the curse of dimensionality under the uncertainty by utilizing non-i.i.d. energy demand and generation from each local agent.
- Experimental results using real datasets establish a significant performance gain of the energy dispatch under the deterministic, asymmetric, and stochastic environments. Particularly, the results show that the proposed MAMRL model saves up to 22.44% of energy consumption cost over a baseline approach while achieving an average accuracy of around 95.8% in a stochastic environment. Our approach also decreases the usage of non-renewable energy up to 11% of total consumed energy.

The rest of this article is organized as follows. Section II presents the system model of self-powered edge computing. The problem formulation is described in Section III. Section IV provides MAMRL framework for solving energy

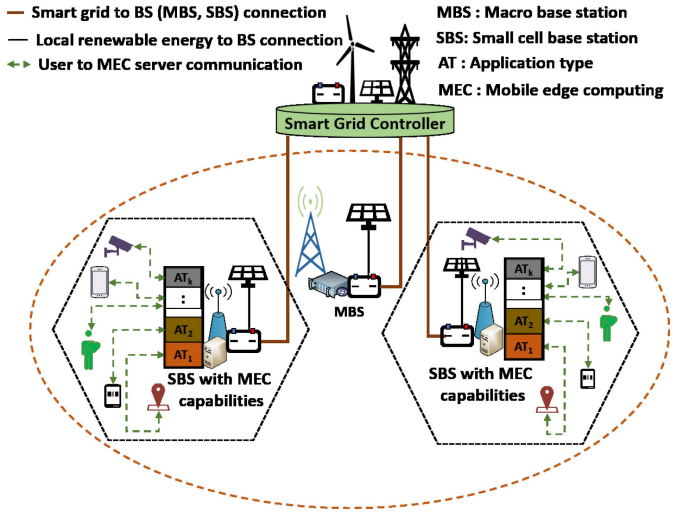


Fig. 2. System model for a self-powered wireless network with MEC capabilities.

dispatch problem. Experimental results are analyzed in Section V. Finally, conclusions are drawn in Section VI.

II. System Model of Self-Powered Edge Computing

Consider a self-powered wireless network that is connected with a smart grid controller as shown in Fig. 2. Such a wireless network enables edge computing services for various MEC applications and services. The energy consumption of the network depends on network operations energy consumption along with the task loads of the MEC applications. Meanwhile, the energy supply of the network relies on the energy generation from renewable sources that are attached to the BSs, as well as both renewable and non-renewable sources of the smart grid. Furthermore, the smart grid controller is a representative of the main power grid (i.e., smart grid), where an additional amount of energy can be supplied via the smart grid controller to the network. Therefore, we will first discuss the energy demand model that includes MEC server energy consumption, and network communication energy consumption. We will then describe the energy generation model that consists of the non-renewable energy generation cost, surplus energy storage cost, and total energy generation cost. Table III illustrates the summary of notations.

A. Energy Demand Model

Consider a set $\mathcal{B} = \{0, 1, 2, \ldots, B\}$ of $B + 1$ (0 for MBS) BSs that encompass B SBSs overlaid over a single MBS. Each BS $i \in \mathcal{B}$ includes a set $\mathcal{K}_i = \{1, 2, \ldots, K_i\}$ of K_i MEC application servers. We consider a finite time horizon $\mathcal{T} = 1, 2, \ldots, T$ with each time slot being indexed by t and having a duration of 15 minutes [29]. The observational period of each time slot t ends at the 15-th minute and is capable of capturing the changes of network dynamics [11], [12], [30]. A set $\mathcal{J}_i$ of J_i heterogeneous MEC application task requests from users will arrive to BS i with an average task arrival rate $\lambda_i(t)$ (bits/s) at time t. The task arrival rate $\lambda_i(t)$ at BS $i \in \mathcal{B}$ follows a Poisson process at time slot t. BS i integrates K_i heterogeneous active MEC application servers that has $u_{k_i}(t)$ (bits/s) processing capacity. Thus, J_i computational task requests will be accumulated into the service pool with an average traffic size $S_i(t)$ (bits) at time slot t. The average traffic arrival rate is defined as $\lambda_i(t) = \frac{1}{S_i(t)}$. Therefore, an *M/M/K* queuing model is suitable to model these J_i user tasks using K_i MEC servers at BS i and time t [31], [32]. The task size of this queuing model is exponentially distributed since the average traffic size $S_i(t)$ is already known. Hence, the service rate of the BS i is determined by $\mu_i(t) = \frac{1}{\mathbb{E}[\sum_{k_i \in \mathcal{K}_i} u_{k_i}(t)]}$. At any given time t, we assume that all of the tasks in $\mathcal{J}_i$ are uniformly distributed at each BS i. Thus, for a given MEC server task association indicator $\Upsilon_{jk_i}(t) = 1$ if task j is assigned to server k at BS i, and 0 otherwise, the average MEC server utilization is defined as follows [11]:

$$\rho_i(t) = \begin{cases} \sum_{j \in \mathcal{J}_i} \sum_{k_i \in \mathcal{K}_i} \Upsilon_{jk_i}(t) \frac{\lambda_i(t)}{\mu_i(t) K_i}, & \text{if } \Upsilon_{jk_i}(t) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

1) MEC Server Energy Consumption: In case of MEC server energy consumption, the computational energy consumption (dynamic energy) will be dependent on the CPU activity for executing computational tasks [16], [17], [33].

TABLE III
Summary of Notations

Notation	Description
$\mathcal{B}$	Set of BSs (SBSs and MBS)
$\mathcal{K}_i$	Set of active servers under the BS $i \in \mathcal{B}$
$\mathcal{J}_i$	Set of user tasks at BS $i \in \mathcal{B}$
$\mathcal{R}$	Set of renewable energy sources
$\rho_i(t)$	Server utilization in BS $i \in \mathcal{B}$
L	No. of CPU cores
$R_i(t)$	Average downlink data of BS i
W_{ij}	Fixed channel bandwidth of BS i for user task j
P_i	Transmission power of BS i
$g_{ij}(t)$	Downlink channel gain between user task j to BS i
$I_{ij}(t)$	Co-channel interference for user task j at BS i
δ_i	Energy coefficient for BS $i \in \mathcal{B}$
f	MEC server CPU frequency for a single core
τ	Server switching capacitance
$\eta_{\text{st}}^{\text{MEC}}(t)$	MEC server static energy consumption
$\eta_{\text{idle}}^{\text{MEC}}(t)$	MEC server idle state power consumption
ϖ_k	Scaling factor of heterogeneous MEC CPU core
$\eta_{\text{st}}^{\text{net}}(t)$	Static energy consumption of BS
c_t^{ren}	Renewable energy cost per unit
c_t^{non}	Non-renewable energy cost per unit
c_t^{sto}	Storage energy cost per unit
ξ_t^{ren}	Amount of renewable energy
ξ_t^{non}	Amount of non-renewable energy
ξ_t^{sto}	Amount of surplus energy
ξ_t^{d}	Energy demand at time slot t
ξ_t^{D}	Random variable for energy demand
$\xi_t^{\text{ren}_{\max}}$	Maximum capacity of renewable energy at BS $i \in \mathcal{B}$
$\mathcal{O}_i$	Set of observation at BS $i \in \mathcal{B}$
$O(.)$	Big O notation to represent complexity
β	Entropy regularization coefficient
γ	Discount factor
θ_i	Learning parameters for BS $i \in \mathcal{B}$
π_{θ_i}	Energy dispatch policy with parameters θ_i at BS $i \in \mathcal{B}$
ϕ	Meta-agent learning parameters

Further, such dynamic energy is also accounted with the thermal design power (TDP), memory, and disk I/O operations of the MEC server [16], [17], [33] and we denote as $\eta_{\text{st}}^{\text{MEC}}(t)$. Meanwhile, static energy $\eta_{\text{idle}}^{\text{MEC}}(t)$ includes the idle state power of CPU activities [16], [18]. We consider, a single core CPU with a processor frequency *f* (cycles/s), an average server utilization $\rho_i(t)$ (using (1)) at time slot *t*, and a switching capacitance $\tau = 5 \times 10^{-27}$ (farad) [17]. The dynamic power consumption of such single core CPU can be calculated by applying a cubic formula $\tau \rho_i(t) f^3$ [18], [34]. Thus, energy consumption of K_i MEC servers with *L* CPU cores at BS *i* is defined as follows:

$$\xi_i^{\text{MEC}}(t) = \begin{cases} \sum_{k \in K_i} \sum_{l \in L} \tau \rho_i(t) f_{k_i}^3 \varpi_{k_{il}} + \eta_{\text{st}}^{\text{MEC}}(t), & \text{if } \rho_i(t) > 0, \\ \eta_{\text{idle}}^{\text{MEC}}(t), & \text{otherwise}, \end{cases} \tag{2}$$

where $\varpi_{k_{il}}$ denotes a scaling factor of heterogeneous CPU core of the MEC server. Thus, the value of $\varpi_{k_{il}}$ is dependent on the processor architecture [35] that assures the heterogeneity of the MEC server.

2) Base Station Energy Consumption: The energy consumption needed for the operation of the network base stations (i.e., SBSs and MBS) includes two types of energy: dynamic and static energy consumption [36]. On one hand, a static energy consumption $\eta_{\text{st}}^{\text{net}}(t)$ includes the energy for maintaining the idle state of any BS, a constant power consumption for receiving packet from users, and the energy for wired transmission among the BSs. On the other hand, the dynamic energy consumption of the BSs depends on the amount of data transfer from BSs to users which essentially relates to the downlink [37] transmit energy. Thus, we consider that each BS $i \in \mathcal{B}$ operates at a fixed channel bandwidth W_{ij} and constant transmission power P_i [37]. Then the average downlink data of BS *i* will be given by [11]:

$$R_i(t) = \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{J}_i} W_{ij} \log_2 \left(1 + \frac{P_i g_{ij}(t)}{\sigma^2 + I_{ij}(t)}\right) \tag{3}$$

where $g_{ij}(t)$ represents downlink channel gain between user task *j* to BS *i*, σ^2 determines a variance of an Additive White Gaussian Noise (AWGN), and $I_{ij}(t)$ denotes the co-channel interference [38], [39] among the BSs. Here, the co-channel interference $I_{ij}(t) = \sum_{i' \in \mathcal{B}, i' \neq i} P_{i'} g_{i'j}(t)$ relates to the transmissions from other BSs $i' \in \mathcal{B}$ that use the same sub-channels of W_{ij}. $P_{i'}$ and $g_{i'j}(t)$ represent, respectively, the transmit power and the channel gain of the BS $i' \neq i \in \mathcal{B}$. Therefore, downlink energy consumption of the data transfer of BS $i \in \mathcal{B}$ is defined by $\frac{P_i S_i(t)}{R_i(t)}$ [watt-seconds or joule], where $\frac{S_i(t)}{R_i(t)}$ [seconds] determines the duration of transmit power P_i [watt]. Thus, the network energy consumption for BS *i* at time *t* is defined as follows [19], [36]:

$$\xi_i^{\text{net}}(t) = \sum_{j \in \mathcal{J}_i} \left(\delta_i^{\text{net}} \frac{P_i S_i(t)}{R_i(t)} + \eta_{\text{st}}^{\text{net}}(t)\right), \tag{4}$$

where δ_i^{net} determines the energy coefficient for transferring data through the network. In fact, the value of δ_i^{net} depends on the type of the network device (e.g., $\delta_i^{\text{net}} = 2.8$ for a 6 unit transceiver remote radio head [36]).

3) Total Energy Demand: The total energy consumption (demand) of the network consists of both MEC server computational energy (in (2)) consumption, and network the operational energy (i.e., BSs energy consumption in (4)). Thus, the overall energy demand of the network at time slot *t* is given as follows:

$$\xi_t^{\text{d}} = \sum_{i \in \mathcal{B}} \Big(\xi_i^{\text{net}}(t) + \xi_i^{\text{MEC}}(t)\Big). \tag{5}$$

The demand ξ_t^{d} is random over time and completely depends on the computational tasks load of the MEC servers.

B. Energy Generation Model

The energy supply of the self-powered wireless network with MEC capabilities relates to the network's own renewable (e.g., solar, wind, biofuels, etc.) sources as well as the main grid's non-renewable (e.g., diesel generator, coal power, and so on) energy sources [8], [9]. In this energy generation model, we consider a set $\mathcal{R} = \{\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_B\}$ of renewable energy sources of the network, with each element $\mathcal{R}_i$ representing the set of renewable energy sources of BS $i \in \mathcal{B}$. Each renewable energy source $q \in \mathcal{R}_i$ at BS $i \in \mathcal{B}$ can generate an amount $\xi_{iq}^{\text{ren}}(t)$ of renewable energy at time *t*. Therefore, the total amount of renewable energy generation $\xi_i^{\text{ren}}(t)$ at BS $i \in \mathcal{B}$ will be $\xi_i^{\text{ren}}(t) = \sum_{q \in \mathcal{R}_i} \xi_{iq}^{\text{ren}}(t)$ for time slot *t*. Thus, the total renewable energy generation for the considered network at time *t* is defined as $\xi_t^{\text{ren}} = \sum_{i \in \mathcal{B}} \xi_i^{\text{ren}}(t)$. The maximum limit of this renewable energy ξ_t^{ren} is less than or equal to the maximum capacity $\xi_t^{\text{ren}_{\max}}$ of renewable energy generation at time period *t*. Thus, we consider a maximum storage limit that is equal to the maximum capacity $\xi_t^{\text{ren}_{\max}}$ of the renewable energy generation [40]–[42]. Further, the self-powered wireless network is able to get an additional non-renewable energy amount ξ_t^{non} from the main grid at time *t*. The per unit renewable and non-renewable energy cost are defined by c_t^{ren} and c_t^{non}, respectively. In general, the renewable energy cost only depends on the maintenance cost of the renewable energy sources [40]–[42]. Therefore, the per unit non-renewable energy cost is greater than the renewable energy cost $c_t^{\text{non}} > c_t^{\text{ren}}$. Additionally, the surplus amount of the energy ξ_t^{sto} at time *t* can be stored in energy storage medium for the future usages [41], [42] and the energy storage cost of per unit energy store is denoted by c_t^{sto}.

1) Non-Renewable Energy Generation Cost: In order to fulfill the energy demand ξ_t^{d} when it is greater than the generated renewable energy ξ_t^{ren}, the main grid can provide an additional amount of energy ξ_t^{non} from its non-renewable sources. Thus, the non-renewable energy generation cost C_t^{non} of the network is determined as follows:

$$C_t^{\text{non}} = \begin{cases} c_t^{\text{non}} \Big[\xi_t^{\text{d}} - \xi_t^{\text{ren}}\Big], & \text{if } \xi_t^{\text{d}} > \xi_t^{\text{ren}}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where c_t^{non} represents a unit energy cost.

2) Surplus Energy Storage Cost: The surplus amount of energy is stored in a storage medium when $\xi_t^{\text{d}} < \xi_t^{\text{ren}}$ (i.e., energy demand is smaller than the renewable energy generation) at time *t*. We consider the per unit energy storage

cost c_t^{sto}. This storage cost depends on the storage medium and amount of the energy store at time t [23], [41], [43], [44]. With the per unit energy storage cost c_t^{sto}, the total storage cost at time t is defined as follows:

$$C_t^{\text{sto}} = \begin{cases} c_t^{\text{sto}}\left[\xi_t^{\text{ren}} - \xi_t^{\text{d}}\right], & \text{if } \xi_t^{\text{d}} < \xi_t^{\text{ren}}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

3) Total Energy Generation Cost: The total energy generation cost includes renewable, non-renewable, and storage energy cost. Naturally, this total energy generation cost will depend on the energy demand ξ_t^{d} of the network at time t. Therefore, the total energy generation cost at time t is defined as follows:

$$Q\left(\xi_t^{\text{ren}}, \xi_t^{\text{d}}\right) = c_t^{\text{ren}}\xi_t^{\text{ren}} + c_t^{\text{non}}\left[\xi_t^{\text{d}} - \xi_t^{\text{ren}}\right]_+ + c_t^{\text{sto}}\left[\xi_t^{\text{ren}} - \xi_t^{\text{d}}\right]_+, \tag{8}$$

where the energy cost of the renewable, non-renewable, and storage energy are given by $c_t^{\text{ren}}\xi_t^{\text{ren}}$, $c_t^{\text{non}}[\xi_t^{\text{d}} - \xi_t^{\text{ren}}]_+$, and $c_t^{\text{sto}}[\xi_t^{\text{ren}} - \xi_t^{\text{d}}]_+$, respectively. In (8), energy demand ξ_t^{d} and renewable energy generation ξ_t^{ren} are stochastic in nature. The energy cost of non-renewable energy (6) and storage energy (7) completely rely on energy demand ξ_t^{d} and renewable energy generation ξ_t^{ren}. Hence, to address the uncertainty of both energy demand and renewable energy generation in a self-powered wireless network, we formulate a two-stage stochastic programming problem. In particular, the first stage makes a decision of the energy dispatch without knowing the actual demand of the network. Then we make further energy dispatch decisions by analyzing the uncertainty of the network demand in the second stage. A detailed discussion of the problem formulation is given in the following section.

III. Problem Formulation With a Two-Stage Stochastic Model

We now consider the case in which the non-renewable energy cost is greater than the renewable energy cost, $c_t^{\text{non}} > c_t^{\text{ren}}$ that is often the case in a practical smart grid as discussed in [40]–[42], and [45]. Here, ξ_t^{ren} and ξ_t^{d} are the continuous variables over the observational duration t. The objective is to minimize the total energy consumption cost $Q(\xi_t^{\text{ren}}, \xi_t^{\text{d}})$. ξ_t^{ren} is the decision variable and the energy demand ξ_t^{d} is a parameter. When the energy demand ξ_t^{d} is known, the optimization problem will be:

$$\chi = \min_{\xi_t^{\text{ren}} \geq 0} Q\left(\xi_t^{\text{ren}}, \xi_t^{\text{d}}\right). \tag{9}$$

In problem (9), after removing the non-negativity constraints $\xi_t^{\text{ren}} \geq 0$, we can rewrite the objective function in the form of piecewise linear functions as follows:

$$Q\left(\xi_t^{\text{ren}}, \xi_t^{\text{d}}\right) = \max_{\xi_t^{\text{ren}}}\left\{\left((c_t^{\text{ren}} - c_t^{\text{non}})\xi_t^{\text{ren}} + c_t^{\text{non}}\xi_t^{\text{d}}\right), \left((c_t^{\text{ren}} + c_t^{\text{sto}})\xi_t^{\text{ren}} - c_t^{\text{sto}}\xi_t^{\text{d}}\right)\right\} \tag{10}$$

where $(c_t^{\text{ren}} - c_t^{\text{non}})\xi_t^{\text{ren}} + c_t^{\text{non}}\xi_t^{\text{d}}$ and $(c_t^{\text{ren}} + c_t^{\text{sto}})\xi_t^{\text{ren}} - c_t^{\text{sto}}\xi_t^{\text{d}}$ determine the cost of non-renewable (i.e., $\xi_t^{\text{d}} > \xi_t^{\text{ren}}$) and storage (i.e., $\xi_t^{\text{d}} < \xi_t^{\text{ren}}$) energy, respectively. Therefore, we have to choose one out of the two cases. In fact, if the energy demand ξ_t^{d} is known and also the amount of renewable energy ξ_t^{ren} is the same as the energy demand, then problem (10) provides the optimal decision in order to exact amount of demand ξ_t^{d}. However, the challenge here is to make a decision about the renewable energy ξ_t^{ren} usage before the demand becomes known. To overcome this challenge, we consider the energy demand ξ_t^{D} as a random variable whose probability distribution can be estimated from the previous history of the energy demand. We can re-write problem (9) using the expectation of the total cost as follows:

$$\min_{\xi_t^{\text{ren}} \geq 0} \mathbb{E}\left[Q\left(\xi_t^{\text{ren}}, \xi_t^{\text{D}}\right)\right]. \tag{11a}$$

The solution of problem (11) will provide an optimal result on average. However, in the practical scenario, we need to solve problem (11) repeatedly over the uncertain energy demand ξ_t^{D}. Thus, this solution approach does not significantly affect our model in terms of scalability while $B + 1$ number of BSs generates a large variety of energy demand over the observational period of t. In fact, energy demand and generation can change over time for each BS $i \in \mathcal{B}$, and they can also induce large variations of demand-generation among the BSs. Hence, the solution to problem (11) cannot rely on an iterative scheme due to a lake of the adaptation for uncertain change of energy demand and generation over time.

We consider the moment of random variable ξ_t^{D} that has a finitely supported distribution and takes values $\xi_{t0}^{\text{D}}, \ldots, \xi_{tB}^{\text{D}}$ with respective probabilities $p_0, \ldots, p_B$ of BSs $B + 1$. The cumulative distribution function (CDF) $H(\xi_t^{\text{D}})$ of energy demand ξ_t^{D} is a step function and jumps of size p_i at each demand ξ_{ti}^{d}. Therefore, the probability distribution of each BS energy demand ξ_{ti}^{d} belongs to the CDF $H(\xi_t^{\text{D}})$ of historical observation of energy demand ξ_t^{D}. In this case, we can convert problem (11) into a deterministic optimization problem and the expectation of energy usage cost $\mathbb{E}[Q(\xi_t^{\text{ren}}, \xi_t^{\text{D}})]$ is determined by $\sum_{i \in \mathcal{B}} p_i Q(\xi_t^{\text{ren}}, \xi_t^{\text{d}})$. Thus, we can rewrite the problem (9) as a linear programming problem using the representation in (10) as follows:

$$\min_{\xi_t^{\text{ren}}, \chi} \ \chi \tag{12}$$

$$\text{s.t.} \ \ \chi \geq (c_t^{\text{ren}} - c_t^{\text{non}})\xi_t^{\text{ren}} + c_t^{\text{non}}\xi_t^{\text{d}}, \tag{12a}$$

$$\chi \geq \left(c_t^{\text{ren}} + c_t^{\text{sto}}\right)\xi_t^{\text{ren}} - c_t^{\text{sto}}\xi_t^{\text{d}}, \tag{12b}$$

$$\xi_t^{\text{ren}_{\max}} \geq \xi_t^{\text{ren}} \geq 0. \tag{12c}$$

For a fixed value of the renewable energy ξ_t^{ren}, problem (12) is an equivalent of problem (10). Meanwhile, problem (12) is equal to $Q(\xi_t^{\text{ren}}, \xi_t^{\text{d}})$. We have converted the piecewise linear function from problem (10) into the inequality constraints (12a) and (12b). Constraint (12c) ensures a limit on the maximum allowable renewable energy usage. We consider p_i as a highest probability of energy demand at each BS $i \in \mathcal{B}$. Therefore, for $B + 1$ BSs, we define $p_0, \ldots, p_B$ as the probability of energy demand with respect to BSs $i = 0, \ldots, B$. Thus, we can rewrite the problem (11) for $B + 1$ BSs $\boldsymbol{\xi}_t^{\text{D}} = (\xi_{t0}^{\text{D}}, \ldots, \xi_{tB}^{\text{D}})$ is as follows:

$$\min_{\xi_t^{\text{ren}}, \chi_0, \ldots, \chi_B} \ \sum_{i \in \mathcal{B}} p_i \chi_i, \tag{13}$$

$$\text{s.t.}\ \ \chi_i \geq (c_t^{\text{ren}} - c_t^{\text{non}})\xi_t^{\text{ren}} + c_t^{\text{non}}\xi_{ti}^{\text{D}}, \forall i \in \mathcal{B}, \quad (13a)$$
$$\chi_i \geq \left(c_t^{\text{ren}} + c_t^{\text{sto}}\right)\xi_t^{\text{ren}} - c_t^{\text{sto}}\xi_{ti}^{\text{D}}, \forall i \in \mathcal{B}, \quad (13b)$$
$$\xi_t^{\text{ren}_{\max}} \geq \xi_t^{\text{ren}} \geq 0, \quad (13c)$$

where p_i represents the highest probability of the energy demand $\xi_{ti}^{\text{D}} = \xi_{ti}^{\text{d}}$, in which ξ_{ti}^{D} is a random variable and ξ_{ti}^{d} denotes a realization of energy demand on BS $i \in \mathcal{B}$ at time *t*. The value of p_i belongs to the empirical CDF $H(\boldsymbol{\xi}_{ti}^{\text{D}})$ of the energy demand ξ_{ti}^{D} for BS $i \in \mathcal{B}$. This CDF is calculated from the historical observation of the energy demand at BS $i \in \mathcal{B}$. In fact, for a fixed value of non-renewable energy ξ_t^{ren}, problem (13) is separable. As a result, we can decompose this problem with a structure of two-stage linear stochastic programming problem [46], [47].

To find an approximation for a random variable with a finite probability distribution, we decompose problem (13) in a two-stage linear stochastic programming under uncertainty. The decision is made using historical data of energy demand, which is fully independent from the future observation. As a result, the first stage of *self-powered energy dispatch* problem for sustainable edge computing is formulated as follows:

$$\min_{\xi_t^{\text{ren}} \geq 0} \ (c_t^{\text{ren}})^\top \xi_t^{\text{ren}} + \mathbb{E}\Big[Q\Big(\xi_t^{\text{ren}}, \boldsymbol{\xi}_t^{\text{D}}\Big)\Big], \quad (14)$$
$$\text{s.t.}\ \ \xi_t^{\text{ren}_{\max}} \geq \xi_t^{\text{ren}} \geq 0, \quad (14a)$$

where $Q(\xi_t^{\text{ren}}, \boldsymbol{\xi}_t^{\text{D}})$ determines an optimal value of the second stage problem. In problem (14), the decision variable ξ_t^{ren} is calculated before the realization of uncertain energy demand $\boldsymbol{\xi}_t^{\text{D}}$. Meanwhile, at the first stage of the formulated problem (14), the cost $(c_t^{\text{ren}})^\top \xi_t^{\text{ren}}$ is minimized for the decision variable ξ_t^{ren} which then allows us to estimate the expected energy cost $\mathbb{E}[Q(\xi_t^{\text{ren}}, \boldsymbol{\xi}_t^{\text{D}})]$ for the second stage decision. Constraint (14a) provides a boundary for the maximum allowable renewable energy usage. Thus, based on the decision of the first stage problem, the second stage problem can be defined as follows:

$$\min_{\xi_t^{\text{non}}, \xi_t^{\text{sto}}} \ (c_t^{\text{non}})^\top \xi_t^{\text{non}} - \left(c_t^{\text{sto}}\right)^\top \xi_t^{\text{sto}}, \quad (15)$$
$$\text{s.t.} \quad \xi_t^{\text{sto}} = |\xi_t^{\text{ren}} - \xi_t^{\text{non}}|, \quad (15a)$$
$$0 \leq \xi_t^{\text{non}} \leq \left(\boldsymbol{\xi}_t^{\text{D}}\right)^\top, \quad (15b)$$
$$\xi_t^{\text{non}} \geq 0. \quad (15c)$$

In the second stage problem (15), the decision variables ξ_t^{non} and ξ_t^{sto} depend on the realization of the energy demand $\boldsymbol{\xi}_t^{\text{D}}$ of the first stage problem (14), where, ξ_t^{ren} determines the amount of renewable energy usage at time *t*. The first constraint (15a) is an equality constraint that determines the surplus amount of energy ξ_t^{sto} must be equal to the absolute value difference between the usage of renewable ξ_t^{ren} and non-renewable ξ_t^{non} energy amount. The second constraint (15b) is an inequality constraint that uses the optimal demand value from the first stage realization. In particular, the value of demand comes from (5) that is the historical observation of energy demand. Finally, the constraint (15c) protects from the non-negativity for the non-renewable energy ξ_t^{non} usage.

The formulated problems (14) and (15) can characterize the uncertainty between network energy demand and renewable energy generation. Particularly, the second stage problem (15) contains random demand $\boldsymbol{\xi}_t^{\text{D}}$ that leads the optimal cost $\mathbb{E}[Q(\xi_t^{\text{ren}}, \boldsymbol{\xi}_t^{\text{D}})]$ as a random variable. As a result, we can rewrite the problems (14) and (15) in a one large linear programming problem for $B + 1$ BSs and the problem formulation is as follows:

$$\min_{\xi_t^{\text{ren}}, \xi_t^{\text{non}}, \xi_t^{\text{sto}}} \ \sum_{t \in \mathcal{T}} \Bigg((c_t^{\text{ren}})^\top \xi_t^{\text{ren}} + \sum_{i \in \mathcal{B}} p_i \times \left[(c_t^{\text{non}})^\top \xi_{ti}^{\text{non}} - \left(c_t^{\text{sto}}\right)^\top \xi_{ti}^{\text{sto}} \right] \Bigg), \quad (16)$$
$$\text{s.t.} \ \ \xi_{ti}^{\text{sto}} = |\xi_{ti}^{\text{ren}} - \xi_{ti}^{\text{non}}|, \forall i \in \mathcal{B}, \quad (16a)$$
$$0 \leq \xi_{ti}^{\text{non}} \leq \xi_{ti}^{\text{D}}, \forall i \in \mathcal{B}, \quad (16b)$$
$$\xi_{ti}^{\text{non}} \geq 0, \forall i \in \mathcal{B}, \quad (16c)$$
$$\xi_t^{\text{ren}_{\max}} \geq \xi_{ti}^{\text{ren}} \geq 0, \forall i \in \mathcal{B}. \quad (16d)$$

In problem (16), for $B + 1$ BSs, energy demand $\xi_{t0}^{\text{D}} \dots \xi_{tB}^{\text{D}}$ happens with positive probabilities $p_0 \dots p_B$ and $\sum_{i \in \mathcal{B}} p_i = 1$. The decision variables are ξ_t^{ren}, ξ_t^{non} and ξ_t^{sto}, which represent the amount of renewable, non-renewable, and storage energy, respectively. Constraint (16a) defines a relationship among all of the decision variables ξ_t^{ren}, ξ_t^{non} and ξ_t^{sto}. In essence, this constraint discretizes the surplus amount of energy for storage. Hence, constraint (16b) ensures the utilization of non-renewable energy based on the energy demand of the network. Constraint (16c) ensures that the decision variable ξ_t^{non} will not be a negative value. Finally, constraint (16d) restricts the renewable energy ξ_t^{ren} usages in to maximum capacity $\xi_t^{\text{ren}_{\max}}$ at time *t*. Problem (16) is an integrated form of the first-stage problem in (14) and the second-stage problem in (15), where the solution of ξ_t^{non} and ξ_t^{sto} completely depends on realization of demand ξ_{ti}^{D} for all $B + 1$ BSs. The decision of the ξ_t^{ren} comes before the realization of demand ξ_{ti}^{D} and, thus, the estimation of renewable energy generation ξ_t^{ren} will be independent and random. Therefore, problem (16) holds the property of relatively complete recourse. In problem (16), the number of variables and constraints is proportional to the numbers of BSs, $B + 1$. Additionally, the complexity of the decision problem (16) leads to $\mathcal{O}(2^{|\mathcal{T}| \times |\mathcal{B}|})$ due to the combinatorial properties of the decisions and constraints [46]–[48].

The goal of the *self-powered energy dispatch* problem (16) is to find an optimal energy dispatch policy that includes amount of renewable ξ_{ti}^{ren}, non-renewable ξ_{ti}^{non}, and storage ξ_{ti}^{sto} energy of each BS $i \in \mathcal{B}$ while minimizing the energy consumption cost. Meanwhile, such energy dispatch policy relies on an empirical probability distribution $H(\boldsymbol{\xi}_t^{\text{D}})$ of historical demand at each BS $i \in \mathcal{B}$ at time *t*. In order to solve problem (16), we choose an approach that does not rely on the conservativeness of a theoretical probability distribution of energy demand in problem (16), and also will capture the uncertainty of renewable energy generation from the historical data. In contrast, we can construct a theoretical probability distribution of energy demand $\boldsymbol{\xi}_t^{\text{D}}$ when we know what the exact distribution is as well as what its parameters will be (e.g., mean, variance, and standard deviation). In fact, in practice, the distribution of energy demand $\boldsymbol{\xi}_t^{\text{D}}$ is unknown and

instead, a certain amount of historical energy demand data are available. As a result, we cannot rely on this distribution to measure uncertainty while the renewable energy generation and energy demand are random over time. Hence, we can obtain time-variant features of both energy demand and generation by characterizing the Markovian properties from the historical observation over time. In particular, we capture the dynamics of Markovian by considering a data-driven approach. This approach can overcome the conservativeness of theoretical probability distribution as historical observation goes to finitely many.

To prevalence the aforementioned contemporary, we propose a multi-agent meta-reinforcement learning framework that can explore the Markovian behavior from historical energy demand and generation of each BS $i \in \mathcal{B}$. Meanwhile, meta-agent can cope with such time-varying features to a globally optimal energy dispatch policy for each BS $i \in \mathcal{B}$.

We design an MAMRL framework by converting the cost minimization problem (16) to a reward maximization problem that we then solve with a data-driven approach. In the MAMRL setting, each agent works as a local agent for each BS $i \in \mathcal{B}$ and determines an observation (i.e., exploration) for the decision variables, renewable ξ_{ti}^{ren}, non-renewable ξ_{ti}^{non}, and storage ξ_{ti}^{sto} energy. The goal of this exploration is to find time-varying features from the local historical data so that the energy demand ξ_{ti}^{d} of the network is satisfied. Furthermore, using these observations and current state information, a meta-agent is used to determine a stochastic energy dispatch policy. Thus, to obtain such dispatch policy, the meta-agent only requires the observations (behavior) from each local agent. Then, the meta-agent can evaluate (exploit) behavior toward an optimal decision for dispatching energy. Further, the MAMRL approach is capable of capturing the exploration-exploitation tradeoff in a way that the meta-agent optimizes decisions of the each self-powered BS under uncertainty. A detailed discussion of the MAMRL framework is given in the following section.

IV. Energy Dispatch With Multi-Agent Meta-Reinforcement Learning Framework

In this section, we developed our proposed multi-agent meta-reinforcement learning framework (as seen in Fig. 3) for energy dispatch in the considered network. The proposed MAMRL framework includes two types of agents: A local agent that acts as a local learner at each self-powered with MEC capabilities BS and a meta-agent that learns the global energy dispatch policy. In particular, each local BS agent can discretize the Markovian dynamics for energy demand-generation of each BS (i.e., both SBSs and MBS) separately by applying deep-reinforcement learning. Meanwhile, we train a long short-term memory (LSTM) [49], [50] as a meta-agent at the MBS that optimizes [26] the accumulated energy dispatch of the local agents. As a result, the meta-agent can handle the non-i.i.d. energy demand-generation of the each local agent with own state information of the LSTM. To this end, MAMRL mitigates the curse of dimensionality for the uncertainty of energy demand and generation while providing

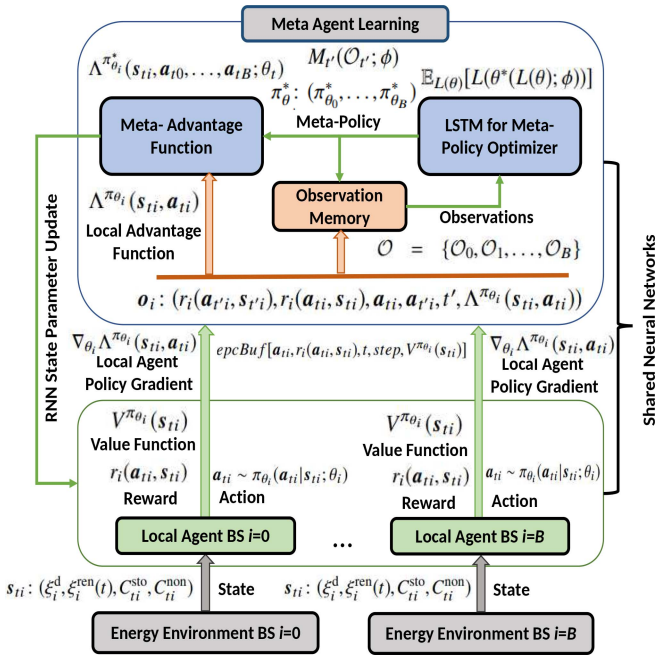


Fig. 3. Multi-agent meta-reinforcement learning framework.

an energy dispatch solution with a less computational and communication complexity (i.e., less message passing between the local agents and meta-agent).

A. Preliminary Setup

In the MAMRL setting, each BS $i \in \mathcal{B}$ acts as a local agent and the number of local agents is equal to $B + 1$ BSs (i.e., 1 MBS and B SBSs). We define a set $\mathcal{S} = \{\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_B\}$ of state spaces and a set $\mathcal{A} = \{\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_B\}$ of actions for the $B + 1$ agents. The state space of a local agent i is defined by $\boldsymbol{s}_{ti} : (\xi_i^{\text{d}}, \xi_i^{\text{ren}}(t), C_{ti}^{\text{sto}}, C_{ti}^{\text{non}}) \in \mathcal{S}_i$, where $\xi_i^{\text{d}}, \xi_i^{\text{ren}}(t)$, C_{ti}^{sto}, and C_{ti}^{non} represent the amount of energy demand, renewable generation, storage cost, and non-renewable energy cost, respectively, at time t. We execute Algorithm 1 to generate the state space for every BSs $i \in \mathcal{B}$, individually. In Algorithm 1, lines 3 to 6 calculate the individual energy consumption of the MEC computation and network operation using (2) and (4), respectively. Overall, the energy demand of the BS i is computed in line 7 and the self-powered energy generation is estimated by line 8 in Algorithm 1. Non-renewable and storage energy costs are calculated in lines 9 and 10 for time slot t. Finally, line 11 creates state space tuple (i.e., $\boldsymbol{s}_{ti}$) for time t in Algorithm 1.

B. Local Agent Design

Consider each local BS agent $i \in \mathcal{B}$ that can take two types of actions $\xi_i^{\text{sto}}(t)$ and $\xi_i^{\text{non}}(t)$ which are the amount of storage energy $\xi_i^{\text{sto}}(t)$, and the amount of non-renewable energy $\xi_i^{\text{non}}(t)$ at time t. We consider a discrete set of actions that consists of two actions $\boldsymbol{a}_{ti} : (\xi_i^{\text{sto}}(t), \xi_i^{\text{non}}(t)) \in \mathcal{A}_i$ for each BS unit $i \in \mathcal{B}$. Since the state $\boldsymbol{s}_{ti}$ and action $\boldsymbol{a}_{ti}$ both contain a time varying information of the agent $i \in \mathcal{B}$, we consider the dynamics of Markovian and represent problem (16)

Algorithm 1 State Space Generation of BS $i \in \mathcal{B}$ in MAMRL Framework

Input: $W_{ij}, P_i, g_{ij}(t), \sigma^2, I_{ij}(t), \Upsilon_{jk_i}(t), \tau, f_{k_i}, \varpi_{k_il}, \eta_{\text{st}}^{\text{MEC}}(t), S_i(t)$
Input: $\delta_i^{\text{net}}, \eta_{\text{st}}^{\text{net}}(t), c_t^{\text{non}}, c_t^{\text{sto}}$
Output: $\boldsymbol{s}_{ti} : (\xi_i^{\text{d}}, \xi_i^{\text{ren}}(t), C_{ti}^{\text{sto}}, C_{ti}^{\text{non}}), \forall \boldsymbol{s}_{ti} \in \mathcal{S}_i \in \mathcal{S}, \forall t \in \mathcal{T}$
Initialization: $\mathcal{R}_i, \mathcal{K}_i, \mathcal{J}_i, \mathcal{S}_i, \lambda_i(t), \mu_i(t), \rho_i(t), R_i(t)$
for *each* $t \in \mathcal{T}$ **do**
  for *each* $i \in \mathcal{B}$ **do**
    for *each* $j \in \mathcal{J}_i$ **do**
      Calculate: $\xi_i^{\text{MEC}}(t)$ using eq. (2)
      Calculate: $\xi_i^{\text{net}}(t)$ using eq. (4)
    end for
    Calculate: $\xi_t^{\text{d}} = \xi_i^{\text{net}}(t) + \xi_i^{\text{MEC}}(t)$ using eq. (5)
    Calculate: $\xi_t^{\text{ren}} = \sum_{q \in \mathcal{R}} \xi_{iq}^{\text{ren}}(t)$
    Calculate: C_t^{non} using eq. (6)
    Calculate: C_t^{sto} using eq. (7)
    Assign: $\boldsymbol{s}_{ti}$:$(\xi_i^{\text{d}}, \xi_i^{\text{ren}}(t), C_{ti}^{\text{sto}}, C_{ti}^{\text{non}})$
  end for
  Append: $\boldsymbol{s}_{ti} \in \mathcal{S}_i$
end for
return $\forall \mathcal{S}_i \in \mathcal{S}$

as a discounted reward maximization problem for each agent *i* (i.e., each BS). Thus, the objective function of the discounted reward maximization problem of agent *i* is defined as follows [28]:

$$r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}) = \max_{\boldsymbol{a}_{ti} \in \mathcal{A}_i} \mathbb{E}_{\boldsymbol{a}_{ti} \sim \boldsymbol{s}_{ti}} \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} \Upsilon_t(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}) \right], \tag{17}$$

where $\gamma \in (0, 1)$ is a discount factor and each reward $\Upsilon_t \boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}$ is considered as,

$$\Upsilon_t(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}) = \begin{cases} 1, & \text{if } \frac{\xi_{ti}^{\text{ren}}}{\xi_{ti}^{\text{d}}} > 1, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

In (18), $\frac{\xi_{ti}^{\text{ren}}}{\xi_{ti}^{\text{d}}}$ determines a ratio between renewable energy generation and energy demand (supply-demand ratio) of the BS agent $i \in \mathcal{B}$ at time *t*. When renewable energy generation-demand ratio $\frac{\xi_{ti}^{\text{ren}}}{\xi_{ti}^{\text{d}}}$ is larger than 1 then the BS agent *i* achieves a reward of 1 because the amount of renewable energy exceeds the demand that can be stored in the storage unit.

Each action $\boldsymbol{a}_{ti}$ of BS agent $i \in \mathcal{B}$ determines a stochastic policy π_{θ_i}. θ_i is a parameter of π_{θ_i} and the energy dispatch policy is defined by $\pi_{\theta_i} : \mathcal{S}_i \times \mathcal{A}_i \mapsto [0, 1]$. Policy π_{θ_i} decides a state transition function $\Gamma : \mathcal{S}_i \times \mathcal{A}_B \mapsto \mathcal{S}_i$ for the next state $\boldsymbol{s}_{t'i} \in \mathcal{S}_i$. Thus, the state transition function Γ of BS agent $i \in \mathcal{B}$ is determined by a reward function (18), where $\Upsilon_t(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}) : \mathcal{S}_i \times \mathcal{A}_i \mapsto \mathbb{R}$. Further, each BS agent $i \in \mathcal{B}$ chooses an action $\boldsymbol{a}_{ti}$ from a parametrized energy dispatch policy $\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_i)$. Therefore, for a given state $\boldsymbol{s}_{ti}$, the state value function with a cumulative discounted reward will be:

$$V^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}) = \mathbb{E}_{\boldsymbol{a}_{ti} \sim \pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_i)} \times \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} \Upsilon_{t+t'}(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}) | \boldsymbol{s}_{ti}, \boldsymbol{a}_{ti} \right], \tag{19}$$

where $\gamma^{t'-t}$ is a discount factor and ensures the convergence of state value function $V^{\pi_{\theta_i}}(\boldsymbol{s}_{ti})$ over the infinity time horizon. Thus, for a given state $\boldsymbol{s}_{ti}$, the optimal policy $\pi_{\theta_i}^*(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti})$ for the next state $\boldsymbol{s}_{t'i}$ can be determined by an optimal state value function while a Markovian property is imposed. Therefore, the optimal value function is given as follows:

$$V^{\pi_{\theta_i}^*}(\boldsymbol{s}_{ti}) = \max_{a_t \in \mathcal{A}} \mathbb{E}_{\pi_{\theta_i}^*} \left[\sum_{i \in \mathcal{B}} r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}) + \sum_{t'=t}^{\infty} \gamma^{t'-t} V_{t'}^{\pi_{\theta_i}}(\boldsymbol{s}_{t'i}) | \boldsymbol{s}_{ti}; \theta_i, \boldsymbol{a}_{ti} \right]. \tag{20}$$

Here, the optimal value function (20) learns a parameterized policy $\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_i)$ by using an LSTM-based Q-networks for the parameters θ_i. Thus, each BS agent $i \in \mathcal{B}$ determines its parametrized energy dispatch policy $\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_i) = P(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_i)$, where $P(\xi_i^{\text{sto}}(t)) = P(\boldsymbol{a}_{ti} = \xi_i^{\text{sto}}(t)|\boldsymbol{s}_{ti}; \theta_i)$ and $P(\xi_i^{\text{non}}(t)) = 1 - P(\xi_i^{\text{sto}}(t))$ for the parameters θ_i. The decision of each BS agent $i \in \mathcal{B}$ relies on θ_i. In particular, energy dispatch policy π_{θ_i} is the probability of taking action $\boldsymbol{a}_{ti}$ for a given state $\boldsymbol{s}_{ti}$ with parameters θ_i. In this setting, each local agent $i \in \mathcal{B}$ is comprised of an actor and a critic [27], [51]. The policy of energy dispatch is determined by choosing an action in (20) that can be seen as an actor of BS agent *i*. Meanwhile, the value function (19) is estimated by a critic of each local BS agent *i*. The critic can criticize actions that are made by the actor of each BS agent *i*. Therefore, each BS agent $i \in \mathcal{B}$ can determine a temporal difference (TD) error [51] based on the current energy dispatch policy of the actor and value estimation by the critic. The TD error is considered as an advantage function and the advantage function of agent *i* is defined as follows:

$$\Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}) = \left(r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}) + \sum_{t'=t}^{\infty} \gamma^{t'-t} V^{\pi_{\theta_i}}(\boldsymbol{s}_{t'i}) \right) - V^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}). \tag{21}$$

Thus, the policy gradient of each BS agent $i \in \mathcal{B}$ is determined as,

$$\nabla_{\theta_i} \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}) = \mathbb{E}_{\pi_{\theta_i}} \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} \nabla_{\theta_i} \log \pi_{\theta_i} \times (\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_i) \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}) \right], \tag{22}$$

where $\log \pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_i)$, and $\Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti})$ represent the actor and critic, respectively, for each local BS agent $i \in \mathcal{B}$.

Using (22), we can discretize the energy dispatch decision $\boldsymbol{a}_{ti} : (\xi_i^{\text{sto}}(t), \xi_i^{\text{non}}(t))$ for each self-powered BS $i \in \mathcal{B}$ in the network. In fact, we can achieve a centralized solution for $\forall i \in \mathcal{B}$ when all of the BSs state information (i.e., demand and generation) are known. However, the space complexity for computation increases as $O(2^{|\mathcal{S}_i| \times |\mathcal{A}_i| \times |\mathcal{B}| \times T})$ and also the computational complexity becomes $O(|\mathcal{S}_i| \times |\mathcal{A}_i| \times |\mathcal{B}|^2 \times T)$ [21]. Further, the solution does not meet the exploration-exploitation

dilemma since the centralized (i.e., single agent) method ignores the interactions and energy dispatch decision strategies of other agents (i.e., BSs) which creates an imbalance between exploration and exploitation. In other words, this learning approach optimizes the action policy by exploring its own state information. Therefore, when we change the energy environment (i.e., demand and generation), this method cannot cope with an unknown environment due to the lack of diverse state information during the training. Next, we propose an approach that not only reduces the complexity but also explores alternative energy dispatch decision to achieve the highest expected reward in (17).

C. Multi-Agent Meta-Reinforcement Learning Modeling

We consider a set $\mathcal{O} = \{\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_B\}$ of $B+1$ observations [24], [52] and for an BS agent $i \in \mathcal{B}$, a single observation tuple is given by $\boldsymbol{o}_i \in \mathcal{O}_i$. For a given state $\boldsymbol{s}_{ti}$, the observation $\boldsymbol{o}_i$ of the next state $\boldsymbol{s}_{t'i}$ consists of $\boldsymbol{o}_i : (r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}), r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}, t', \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}))$, where $r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i})$, $r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti})$, $\boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}$, t' and $\Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti})$ are next-state discounted rewards, current state discounted rewards, next action, current action, time slot, and TD error, respectively. Here, a complete information of the observation $\boldsymbol{o}_i$ is correlated with the state space $\boldsymbol{o}_i : \mathcal{S}_i \mapsto \mathcal{O}_i$ while observation $\boldsymbol{o}_i$ does not require the complete state information of the previous states.

Thus, the space complexity for computation at each BS agent $i \in \mathcal{B}$ leads to $O((|\mathcal{S}_i| + |\mathcal{A}_i|)^2 \times T)$. Meanwhile, the computational complexity for each time slot t becomes $O(|\mathcal{S}_i|^2 \times \mathcal{A}_i \times \theta_t + H)$, where θ_t is the learning parameter and H represents the numbers of LSTM units. Each BS agent $i \in \mathcal{B}$ requires to send an amount of $|\mathcal{O}_i|$ observational data (i.e., payload) to the meta-agent. Therefore, the communication overhead for each BS agent $i \in \mathcal{B}$ leads to $O(\frac{|\mathcal{O}| \times T}{B+1})$. On the other hand, the computational complexity of the meta-agent leads to $O(|\mathcal{O}|^2 \times \phi + H)$ while ϕ represents learning parameter at meta-agent. In particular, for a fixed number of output memory ϕ, the meta-agent's update complexity at each time slot t becomes $O(\phi^2)$ [53]. Further, when transferring the learned parameters $\theta_{t'}$ from the meta-agent to all local agents $\forall i \in \mathcal{B}$, the communication overhead goes to the $O(\theta_{t'} \times (B+1))$ at each time slot t. Here, the size of $\theta_{t'}$ depends on the memory size of the LSTM cell at the meta-agent [see Appendix A].

In the MAMRL framework, the local agents work as an optimize and the meta-agent performs the role of optimizer [26]. To model our meta-agent, we consider an LSTM architecture [49], [50] that stores its own state information (i.e., parameters) and the local agent (i.e., optimizee) only provides the observation of a current state. In the proposed MAMRL framework, a policy π_{θ_i} is determined by updating the parameters[1] θ_i. Therefore, we can represent the state value function (20) for time t is as follows: $V^{\pi^*_{\theta_i}}(\boldsymbol{s}_{ti}) \approx V^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}; \theta_t)$, and the advantage (temporal difference) function (21) is presented by, $\Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}) \approx \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}; \theta_t)$. As a result, the parameterized policy is defined by, $\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}) \approx \pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_t)$. Considering all of the BS agents $B+1$ and the advantage function (21) is rewritten as,

$$\begin{aligned}
&\Lambda^{\pi^*_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}; \theta_t) \\
&\quad = r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \\
&\qquad + \sum_{\boldsymbol{s}_{t'i} \in \mathcal{S}_i, t'=t}^{\infty} \gamma^{t'-t} \Gamma(\boldsymbol{s}_{t'i}|\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \\
&\qquad \times V^{\pi_{\theta_i}}\Big(\boldsymbol{s}_{t'i}, \pi^*_{\theta_0}, \ldots, \pi^*_{\theta_B}\Big) - V^{\pi_{\theta_i}}\Big(\boldsymbol{s}_{ti}, \pi^*_{\theta_0}, \ldots, \pi^*_{\theta_B}\Big),
\end{aligned} \tag{23}$$

where $\pi^*_\theta : (\pi^*_{\theta_0}, \ldots, \pi^*_{\theta_B})$ is a joint energy dispatch policy and $\Gamma(\boldsymbol{s}_{t'i}|\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \mapsto [0, 1]$ represents state transition probability. Using (23), we can get the value loss function for agent i and the objective is to minimize the temporal difference [27],

$$\begin{aligned}
L(\theta_i) = \min_{\pi_{\theta_i}} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{2} \Bigg(\Bigg(r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}) + \sum_{t'=t}^{\infty} \gamma^{t'-t} V^{\pi^*_{\theta_i}}(\boldsymbol{s}_{t'i}|\theta_t) \Bigg) \\
- V^{\pi^*_{\theta_i}}(\boldsymbol{s}_{ti}) \Bigg)^2.
\end{aligned} \tag{24}$$

To improve the exploration with a low bias, we consider an entropy regularization[2] $\beta h(\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_t))$ that cope with the non-i.i.d. energy demand and generation for all of the BS agents $\forall i \in \mathcal{B}$. Here, β is a coefficient for the magnitude of regularization and $h(\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_t))$ determines the entropy of the policy π_{θ_i} for the parameter θ_i. Additionally, a larger value of $\beta h(\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_t))$ encourages the agents to have a more diverse exploration to estimate the energy dispatch policy. Thus, we can redefine the policy loss function as follows:

$$L(\theta_i) = -\mathbb{E}_{\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}}\big[\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}) + \beta h\big(\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_t)\big)\big]. \tag{25}$$

Therefore, the policy gradient of the loss function (25) is defined in terms of temporal difference and entropy. The policy gradient of the loss function is defined as follows:

$$\begin{aligned}
\nabla_{\theta_i} L(\theta_i) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t'=t}^{\infty} \nabla_{\theta_i} \log \pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}) \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}|\theta_t) \\
+ \beta h(\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_t)).
\end{aligned} \tag{26}$$

To design our meta-agent, we consider meta-agent parameters ϕ and optimized parameters θ^* of the optimizee (i.e., local agent). The meta-agent is defined as $M_t(\mathcal{O}_t; \phi) := M_t(\nabla_{\theta_t} L(\theta_t); \phi)$, where $M_t(.)$ is modeled by an LSTM. Consider an observational vector $\mathcal{O}_{it'} \in \mathcal{O}$ of a local BS agent $i \in \mathcal{B}$ at time t' and each observation is $\boldsymbol{o}_i : (r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}),$

[1]We consider recurrent neural networks (RNNs) state parameters for the parameterization of energy dispatch policy. In particular, we consider a long short-term memory (LSTM) for RNN, in which cell state and hidden state are considered as parameters.

[2]Entropy [54]–[57] can allow us to manage non-i.i.d. datasets when changes in the environment over time lead to an uncertainty. Therefore, we use entropy regularization to handle the non-i.i.d. energy demand and generation over time by managing with the uncertainty for each BS agent $i \in \mathcal{B}$.

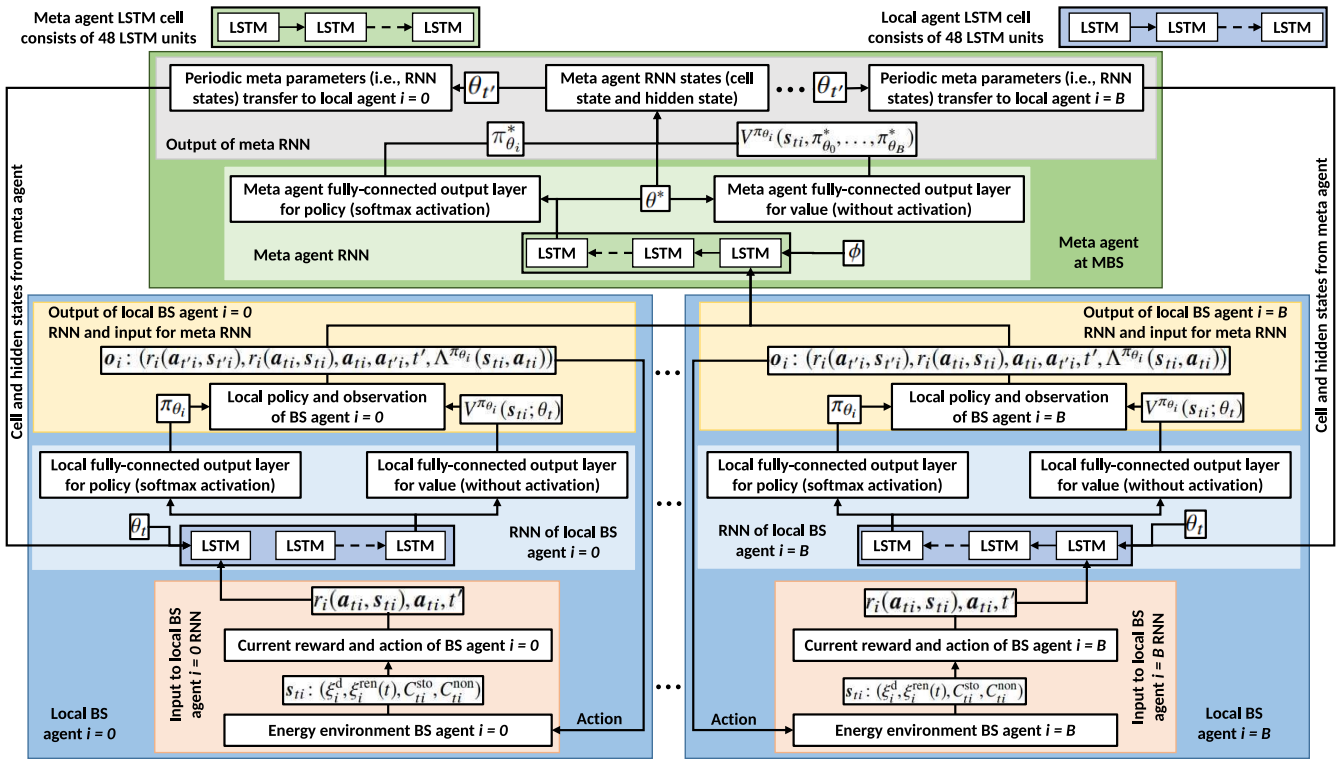


Fig. 4. Recurrent neural network architecture for the proposed multi-agent meta-reinforcement learning framework.

$r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti})$, $\boldsymbol{a}_{ti}$, $\boldsymbol{a}_{t'i}$, t', $\Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti})) \in \mathcal{O}_{it'}$. The LSTM-based meta-agent takes the observational vector $\mathcal{O}_{it'}$ as an input. Meanwhile, the meta-agent holds long-term dependencies by generating its own state with parameters ϕ. To do this, the LSTM model creates several gates to determine an optimal policy $\pi^*_{\theta_i}$ and advantage function $\Lambda^{\pi^*_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}; \theta_t)$ for the next state $\boldsymbol{s}_{t'i}$. As a result, the structure of the recurrent neural network for the meta-agent is the same as the LSTM model [49], [50]. In particular, each LSTM unit for the meta-agent consists of four gate layers such as forget gate $\boldsymbol{F}_{t'}$, input gate $\boldsymbol{I}_{t'}$, cell state $\hat{\boldsymbol{E}}_{t'}$, and output $\boldsymbol{Z}_{t'}$ layer. The cell state gate $\hat{\boldsymbol{E}}_{t'}$ usages a $\tanh$ activation function and other gates are used sigmoid $\sigma(.)$ as an activation function. Thus, the outcome of the meta policy for a single unit LSTM cell is presented as follows:

$$M_{t'}(\mathcal{O}_{t'}; \phi) = \text{softmax}\Big((\boldsymbol{H}_{t'})^\top\Big), \tag{27}$$

where

$$\boldsymbol{F}_{t'} = \sigma\Big(\phi_{FO}(\mathcal{O}_{it'})^\top + \phi_{FH}(\boldsymbol{H}_t)^\top + \boldsymbol{b}_F\Big), \tag{27a}$$

$$\boldsymbol{I}_{t'} = \sigma\Big(\phi_{IO}(\mathcal{O}_{it'})^\top + \phi_{IH}(\boldsymbol{H}_t)^\top + \boldsymbol{b}_I\Big), \tag{27b}$$

$$\hat{\boldsymbol{E}}_{t'} = \tanh\Big(\phi_{EO}(\mathcal{O}_{it'})^\top + \phi_{EH}(\boldsymbol{H}_t)^\top + \boldsymbol{b}_E\Big), \tag{27c}$$

$$\boldsymbol{E}_{t'} = \hat{\boldsymbol{E}}_{t'} \odot \boldsymbol{I}_{t'} + \boldsymbol{F}_{t'} \odot \boldsymbol{E}_t, \tag{27d}$$

$$\boldsymbol{Z}_{t'} = \sigma\Big(\phi_{ZO}(\mathcal{O}_{it'})^\top + \phi_{ZH}(\boldsymbol{H}_t)^\top + \boldsymbol{b}_Z\Big), \tag{27e}$$

$$\boldsymbol{H}_{t'} = \tanh(\boldsymbol{E}_{t'}) \odot \boldsymbol{Z}_{t'}. \tag{27f}$$

In the meta-agent policy formulation (27), the forget gate vector (27a) determines what information is needed to throw away. Input gate vector (27b) helps to decide which information is needed to update, the cell state (27c) creates a vector of new candidate values using $\tanh(\cdot)$ function, and updates the cell state information by applying (27d). The output layer (27e) that determines what parts of the cell state are going to output and calculate the cell outputs using the equation (27f). Further, the cell state through the $\tanh(\cdot)$ will restrict the values between -1 and $+1$. This entire process is followed for each LSTM block and finally, (27) determines the meta-policy for $\pi^*_{\theta_i}$ of the state $s_{t'}$. In addition, optimized RNN state parameters θ^* are obtained from the cell state (27d) and hidden state (27f) of an LSTM unit. Thus, the loss function $L(\phi) = \mathbb{E}_{L(\theta)}[L(\theta^*(L(\theta); \phi))]$ of meta-agent depends on the distribution of $L(\theta_t)$ and the expectation of the meta-agent loss function is defined as follows [26]:

$$L(\phi) = \mathbb{E}_{L(\theta)}\left[\sum_{t=1}^{T} L(\theta_t)\right]. \tag{28}$$

In the proposed MAMRL framework, we transfer the learned parameters (i.e., cell state and hidden state) of meta-agent to the local agents so that each local agent will be estimated an optimal energy dispatch policy by updating its own learning parameters. Thus, the parameters of each agent (i.e., BS) is updated with $\theta_{t'} = \theta^*$ while $\pi^*_{\theta_i} = M_t(\nabla_{\theta_t} L(\theta_t); \phi)$ to decide the energy dispatch policy.

We consider an LSTM-based recurrent neural network (RNN) for the both local agents and the meta-agent. This LSTM RNN consists of 48 LSTM units for each LSTM cell as shown in Fig. 4. In particular, the configuration of the LSTM for the meta-agent and each local agent is the same while the objective of the loss functions differ from local

agent to meta-agent. In which, local BS agent determines its own energy dispatch policy by exploring its own environmental state information for reducing the TD error. Meanwhile, meta-agent deals with the observations of each local BS agent by exploiting its own RNN states information using entropy based loss function to capture non-i.i.d. energy demand and generation of each local BS. Therefore, having different loss functions for local and meta agent leads the proposed MAMRL model to learn a domain specific generalized model so that it can cope with an unknown environment. Further, this RNN consists of a branch of two fully connected output layers on top of the LSTM cell. In particular, fully connected layer with a softmax activation is considered for energy dispatch policy determination, and another fully connected output layer without activation function is deployed for value function estimation. Thus, the advantage is calculated based on value function estimation from the second fully connected layer. Each local LSTM-based RNN receives a current reward $r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti})$, current action $\boldsymbol{a}_{ti}$, and next time slot t' as an input for each BS agent $i \in \mathcal{B}$. Meanwhile, this local LSTM model estimates a policy π_{θ_i} and value $V^{\pi_{\theta_i}}(\boldsymbol{s}_{ti})$ for BS agent $i \in \mathcal{B}$. On the other hand, meta agent LSTM-based RNN feeds input as an observational tuple $\boldsymbol{o}_i : (r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}), r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}, t', \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}))$ from each BS agent $i \in \mathcal{B}$. This observation consists of the current and next reward, current and next action, next time slot, and TD error for each BS agent *i*. Thus, this meta agent estimates parameters $\theta_{t'}$ to find a globally optimal energy dispatch policy $\pi^*_{\theta_i}$ for each BS $i \in \mathcal{B}$. The learned parameters of the meta-agent are transferred to each local BS agent $i \in \mathcal{B}$ asynchronously while this local agent updates its own parameters for estimating the globally optimal energy dispatch policy via the local LSTM-based RNN. In particular, the learned parameters (i.e., RNN states) are transferred from meta-agent to each local agent $i \in \mathcal{B}$. Additionally, these RNN state parameters include cell state and hidden state of the LSTM cell, which do not depend on any of the fully connected out layers of the proposed RNN architecture. Meanwhile, each local agent $i \in \mathcal{B}$ updates its own RNN states using the transferred parameters by the meta-agent. We consider a cellular network for exchanging observations and parameters between local BS agent and meta-agent.

We run the proposed Algorithm 2 at each self-powered BS $i \in \mathcal{B}$ with MEC capabilities as local agent *i*. The input of Algorithm 2 is the state information $\mathcal{S}_i$ of local agent *i*, which is the output from Algorithm 1. The cumulative discounted reward (17) and state value in (19) are calculated in lines 5 and 6, respectively (in Algorithm 2) for each step (until the maximum step size[3] for time step *t*). Consequently, based on a chosen action $\boldsymbol{a}_{ti}$ from the estimated policy $\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti})$ (in line 7), episode buffer is generated and appended in line 8. Advantage function (21) of local agent *i* is evaluated in line 12 and the policy gradient (22) is calculated in line 13 using an LSTM-based local

[3]To capture the heterogeneity for energy demand and generation of each BS separately, we consider the same number of user tasks that are executed by each BS agent $i \in \mathcal{B}$ during one observational period *t* as the steps size.

Algorithm 2 Local Agent Training of Energy Dispatch of BS $i \in \mathcal{B}$ in MAMRL Framework

Input: $\boldsymbol{s}_{ti} : (\xi_i^{\mathrm{d}}, \xi_i^{\mathrm{ren}}(t), C_{ti}^{\mathrm{sto}}, C_{ti}^{\mathrm{non}}), \forall \boldsymbol{s}_{ti} \in \mathcal{S}_i, \forall t \in T$
Output: $\boldsymbol{o}_i : (r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}), r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}, t', \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}))$, $\boldsymbol{o}_i \in \mathcal{O}_i$, $\nabla_{\theta_t} L(\theta_t)$
Initialization: LocalLSTM(.), $\theta_i, i \in \mathcal{B}, \gamma, \mathcal{O}_i$
1: **for** *episode* $= 1$ to *maximum episodes* **do**
2: *Initialization: epcBuf*[]
3: **for** *each* $t \in \mathcal{T}$ **do**
4: **for** *step* $= 1$ to *MaxStep* **do**
5: *Calculate:* $r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti})$ using eq. (17)
6: *Calculate:* $V^{\pi_{\theta_i}}(\boldsymbol{s}_{ti})$ using eq. (19)
7: *Choose Action:* $\boldsymbol{a}_{ti} \sim \pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti})$
8: *Append:* $epcBuf[\boldsymbol{a}_{ti}, r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), t, step, V^{\pi_{\theta_i}}(\boldsymbol{s}_{ti})]$
9: **end for**
10: $LocalLSTM(r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, t' = t + 1)$ {LSTM-based local RNN block}
11: {
12: *Evaluate:* $\Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti})$ using eq. (21)
13: *Local agent policy gradient:* $\nabla_{\theta_i} \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti})$ using eq. (22) {In (22), $\pi_{\theta_i}(\boldsymbol{a}_{ti}|\boldsymbol{s}_{ti}; \theta_i)$ is determined by a fully connected output layer with a softmax activation function and $\Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti})$ is calculated through a fully connected output layer without activation function}
14: }
15: *Append:* $\boldsymbol{o}_i$:$(r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}), r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}, t', \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}))$, $\boldsymbol{o}_i \in \mathcal{O}_i$
16: *Get Meta-agent policy* $\pi^*_{\theta_i}$ *and RNN states* θ^*: $M_t(\mathcal{O}_t; \phi)$ using Algorithm 3
17: Update: $\theta_{t'} = \theta^*$ {RNN states update}
18: **end for**
19: **end for**
20: **return** *new_state*$(\boldsymbol{s}_{t'i} = \mathrm{argmax}_{\pi^*_{\theta_i}}(\boldsymbol{a}_{ti})), i \in \mathcal{B}$

neural network. Algorithm 2 generates observational tuple $\boldsymbol{o}_i : (r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}), r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}, t', \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}))$ in line 15. Here, we transfer the knowledge of local BS agent $i \in \mathcal{B}$ to the meta-agent learner (deployed in MBS) in Algorithm 3 so as to optimize the energy dispatch decision (in Algorithm 2 line 16). Hence, the observation tuple $\boldsymbol{o}_i$ of local BS agent *i* consists of only the decision from BS *i*, where does not require to send all of the state information to meta-agent learner. Employing the meta-agent policy gradient, each local agent is capable of updating the energy dispatch decision policy in line 17 in Algorithm 2. Finally, the energy dispatch policy is executed in line 20 at the BS $i \in \mathcal{B}$ by local agent *i*.

The meta-agent learner (Algorithm 3 in MBS) receives the observations $\mathcal{O}_i \in \mathcal{O}$ from each local BS agent $i \in \mathcal{B}$ asynchronously. Then the meta-agent asynchronously updates the meta policy gradient of the each BS agent $i \in \mathcal{B}$. Lines from 4 to 12 of Algorithm 3 represent the LSTM block for the meta-agent. In Algorithm 3, entropy loss (25) and gradient of the loss (26) are estimated in lines 6 and 7, respectively. In order to estimate this, Algorithm 3 deploys a fully connected output layer without activation function, so that advantage loss can be calculated without affecting the value that is calculated by the value function of the proposed MAMRL framework. The meta-agent energy dispatch policy is updated in line 10 of Algorithm 3. Before that, a fully connected output layer with a softmax activation function of the LSTM cell assists to determine the energy dispatch policy and meta policy loss in lines 8 and 9 (in Algorithm 3), respectively, for the meta-agent.

Algorithm 3 Meta-Agent Learner of Energy Dispatch in MAMRL Framework

Input: $\boldsymbol{o}_i : (r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}), r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}, t', \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}))$, $\forall \boldsymbol{o}_i \in \mathcal{O}_i$, $t \in \mathcal{T}$, $i \in \mathcal{B}$
Output: ϕ
Initialization: MetaLSTM(.), ϕ, π_{θ_i}, γ
1: **for** *each* $t \in \mathcal{T}$ **do**
2: **for** *each* $i \in \mathcal{B}$ **do**
3: $\boldsymbol{o}_i : (r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}), r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}, t', \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}))$, $\boldsymbol{o}_i \in \mathcal{O}_i$
4: $MetaLSTM(\mathcal{O}_i, \pi_{\theta_i})$ {LSTM-based RNN block}
5: {
{Lines from 6 to 7 using fully connected output layer without activation function}
6: *Entropy loss:* $L(\theta_i)$ using eq. (25)
7: *Gradient of the loss:* $\nabla_{\theta_t} L(\theta_t)$ using eq. (26)
{Policy is estimated using a fully connected output layer with softmax activation function}
8: *Calculate:* $\pi_{\theta_i} = M_t(\nabla_{\theta_t} L(\theta_t); \phi)$ using eq. (27)
9: *Get meta policy loss* $L(\phi)$ using eq. (28)
10: *Update:* $\pi^*_{\theta_i} = \pi_{\theta_i}$
11: *Get RNN states:* θ^*
{cell state and hidden state from the LSTM cell}
12: }
13: **end for**
14: *Send:* Meta-agent policy $\pi^*_{\theta_i}$ and RNN states θ^*
15: **end for**
16: **return**

Additionally, the meta-agent utilizes the observations of the local agents and determines its own state information that helps to estimate the energy dispatch policy of the meta-agent. In line 11, the meta-agent RNN states θ^* (i.e., cell and hidden states) are received from the considered LSTM cell in Algorithm 3. Finally, the meta-agent policy and RNN states are transferred to each BS agent for updating the parameters (i.e., RNN states) of each local BS agent. To this end, a meta-agent learner deployed at center node (i.e., MBS) in the considered network and sends the learning parameters of the optimal energy dispatch policy to each local BS (i.e., MBS and SBS) through the network.

The proposed MAMRL framework established a guarantee to converge with an optimal energy dispatch policy. In fact, the MAMRL framework can be reduced to a $|\mathcal{B}|$-player Markovian game [58], [59] as a base problem that establishes more insight into convergence and optimality. The proposed MAMRL model has at least one Nash equilibrium point that ensures an optimal energy dispatch policy. This argument is similar from the previous studies of $|\mathcal{B}|$-player Markovian game [58], [59]. Hence, we can conclude with the following proposition:

Proposition 1: $\pi^*_{\theta_i}$ is an optimal energy dispatch policy that is an equilibrium point with an equilibrium value $V^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \pi^*_{\theta_0}, \ldots, \pi^*_{\theta_B})$ for BS i [see Appendix B].

We can justify the convergence of MAMRL framework via the following Proposition:

Proposition 2: Consider a stochastic environment with a state space $\boldsymbol{s}_{ti} \in \mathcal{S}, i \in \forall \mathcal{B}$ of $|\mathcal{B}|$ BS agents such that all BS agents are initialized with an equal probability of 0.5 for a binary actions, $P(\xi_i^{\text{sto}}(t)) = P(\xi_i^{\text{non}}(t)) = \theta_i \approx 0.5$, where $\boldsymbol{a}_{ti} : (\xi_i^{\text{sto}}(t), \xi_i^{\text{non}}(t)) \in \mathcal{A}_i, \forall i \in \mathcal{B}$, and $r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB})$. Therefore, to estimate the gradient of loss function (24), we can establish a relationship among the gradient of approximation $\hat{\nabla}_{\theta_i} L(\theta_i)$ and true gradient $\nabla_{\theta_i} L(\theta_i)$,

$$P\Big(\hat{\nabla}_{\theta_i} L(\theta_i), \nabla_{\theta_i} L(\theta_i) > 0\Big) \propto (0.5)^{|\mathcal{B}|}. \tag{29}$$

See Appendix C.

Propositions 1 and 2 validate the optimality and convergence, respectively for the proposed MAMRL framework. Proposition 1 guarantees an optimal energy dispatch policy. Meanwhile, Proposition 2 ensures that the proposed MAMRL model can meet the convergence for a single state $\boldsymbol{s}_{ti} \in \mathcal{S}, i \in \forall \mathcal{B}$. That implies this model is also able to converge for $\forall \boldsymbol{s}_{ti} \in \mathcal{S}, i \in \forall \mathcal{B}$.

The significance of the proposed MAMRL model are explained as follows:

- First, each BS (i.e., local agent) can explore its own energy dispatch policy based on individual requirements for the energy generation and consumption. Meanwhile, the meta-agent exploits each BS energy dispatch decision from its own recurrent neural networks state information. As a result, individual BS anticipates its own energy demand and generation while meta-agent handles the non-i.i.d. energy demand and generation for all BS agents to efficiently meet the exploration-exploitation tradeoff of the proposed MAMRL.
- Second, the proposed MAMRL model can effectively handle distinct environment dynamics for non-i.i.d. energy demand and generation among the agents.
- Third, the proposed MAMRL model ensures less information exchange between the local agents and meta-agent. In particular, each local BS agent only sends an observational vector to meta-agent and received neural network parameters at the end of 15 minutes observation period. Additionally, the proposed MAMRL model does not require sending an entire environment state from each local agent to the meta-agent.
- Finally, the meta-agent can learn a generalized model toward the energy dispatch decision and transfer its skill to each local BS agent. This, in turn, can significantly increase the learning accuracy as well as reduce the computational time for each local BS agent thus enhancing the robustness of the energy dispatch decision.

We benchmark the proposed MAMRL framework by performing an extensive experimental analysis, and the experimental analysis and discussion are given in the later section.

V. Experimental Results and Analysis

In our experiment, we use the CRAWDAD nyupoly/video packet delivery dataset [60] to discretize the self-powered SBS network's energy consumption. Further, we choose a state-of-the-art UMass solar panel dataset [61] to evaluate renewable energy generation. We create deterministic, asymmetric, and stochastic environments by selecting different days of the same solar unit for the generation. Meanwhile, usage several session from the network packet delivery dataset. We train our proposed meta-reinforcement learning (Meta-RL)-based MAMRL framework using deterministic environment and evaluate the testing performance for the three environments.

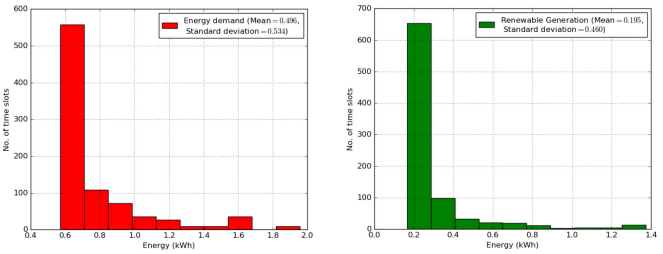
TABLE IV
SUMMARY OF EXPERIMENTAL SETUP

Description	Value
No. of SBSs (no. of local agents)	9
No. of MEC servers in each SBS	2
No. of MBS (meta-agent)	1
Channel bandwidth	180 kHz [62]
System bandwidth	20 MHz [17]
Transmission power	27 dB [16]
Channel gain	$140.7 + 36.7 \log d$ [17]
A variance of an AWGN	-114 dBm/Hz [62]
Energy coefficient for data transfer δ_i^{net}	2.8 [36]
MEC server CPU frequency f	2.5 GHz [16]
Server switching capacitance τ	5×10^{-27} (farad) [17]
MEC static energy $\eta_{\text{st}}^{\text{MEC}}(t)$	[7.5, 25] Watts [63]
Task sizes (uniformly distributed)	[31, 1546060] bytes [60]
No. of task requests at BS i	$[1, 10, 000]$ [11]
Unit cost renewal energy c_t^{ren}	\$50 per MW-hour [45]
Unit cost non-renewal energy c_t^{non}	\$102 per MW-hour [45]
Unit cost storage energy c_t^{sto}	10% additional [44]
Initial discount factor γ	0.9
Initial action selection probability	[0.5, 0.5]
One observation period t	15 minutes
No. of episodes	800
No. of epochs T for each day	96
No. of steps for each epoch at each agent i	$J_i = [1, 10, 000]$ [60]
No. of actions	2 (i.e., $\xi_i^{\text{sto}}(t)$, $\xi_i^{\text{non}}(t)$)
No. of LSTM units in one LSTM cell	48
No. of LSTM cells	10 (i.e., B+1)
LSTM cell API BasicLSTMCell(.)	tf.contrib.rnn [64]
Entropy regularization coefficient β	0.05
Learning rate	0.001
Optimizer	Adam [65]
Output layer activation function	Softmax [51]

Three environments[4] are as follows: 1) In the deterministic environment, both network energy consumption and renewable generation are known, 2) network energy consumption is known but renewable generation is unknown in the asymmetric environment, and 3) the stochastic environment contains both energy consumption and renewable generation are unknown. To benchmark the proposed MAMRL framework intuitively, we have considered a centralized single-agent deep-RL, multi-agent centralized A3C deep-RL with a same neural networks configuration as the proposed MAMRL, and a pure greedy model as baselines. These are as follows:

- We consider the neural advantage actor-critic (A2C) [51], [66] method as a centralized single-agent deep-RL. In particular, the learning environment encompasses the state information of all BSs $\forall i \in \mathcal{B}$ and is learned by a neural A2C [51], [66] scheme with the same configuration as the MAMRL model.
- An asynchronous advantage actor-critic (A3C) based multi-agent RL framework [28] is considered a second benchmark in a cooperative environment [27]. In particular, each local actor can find its own policy in a decentralized manner while a centralized critic is augmented with additional policy information. Therefore, this model is learned by a centralized training with decentralized execution [28]. We call this model a multi-agent centralized A3C deep-RL [28]. The environment (i.e., state information) of this model remains the same for all of the local actor agents. To ensure a meaningful comparison with the proposed MAMRL model, we employ this joint energy dispatch policy using the same advantage function (23) as the MAMRL model.
- We deploy a pure greedy-based algorithm [51] to find the best action-value mapping. In particular, this algorithm never takes the risk to choose an unknown action. Meanwhile, it explores other strategies and learns from them so as to infer more reasonable decisions. Thus, we choose this upper confidence bounded action selection mechanism [51] as one of the baselines used for benchmarking our proposed MAMRL model.

[4]For example, we train and test the MAMRL model using the known (i.e., deterministic environment) network energy consumption, and renewable generation data of day 1. Then we have tested the trained model using day 2 data, where network energy consumption is known, and renewable generation is unknown which represents an asymmetric environment. In a stochastic environment, let us consider day 3 data, where both energy consumption and renewable generation are unknown to the trained model.



(a) MEC network energy demand (b) Renewable energy generation

Fig. 5. Histogram of energy demand and renewable energy generation for 9 SBSs and each SBS consists of 96 time slots after preprocessing using Algorithm 1.

We implement our MAMRL framework using multi-threading programming in Python platform,[5] along with TensorFlow APIs [68]. Table IV shows the key parameters of this experiment setup.

We prepossess both of the datasets ([60] and [61]) using Algorithm 1 that generates the state space information. The histograms of the network energy demand (in 5(a)) and a renewable energy generation (in 5(b)) of the deterministic environment are shown in Fig. 5. To the best of our knowledge, there are no publicly available datasets that comprises both of energy consumption and generation of a self-powered network with MEC capabilities. Additionally, if we change the environment using other datasets, the proposed MAMRL framework can deal with the new, unknown environment by using the skill transfer feature from the meta-agent to each local BS agent. In particular, the MAMRL approach can readily deal with the case in which the BS agent achieves a much lower reward due to more variability in consumption and generation. As a result, the above experiment setup is reasonable for the benchmarking of the proposed MAMRL framework.

Fig. 6 illustrates the reward achieved by each local SBS along with a meta-agent, where we take an average reward for each 50 episodes. In the MAMRL setting, we design a maximum reward of 96 (15 minute slot for 24 hours), where meta-agent converges with a high reward value (around 90). Hence, all of the local agents converge with around $80 - 85$

[5]MAMRL

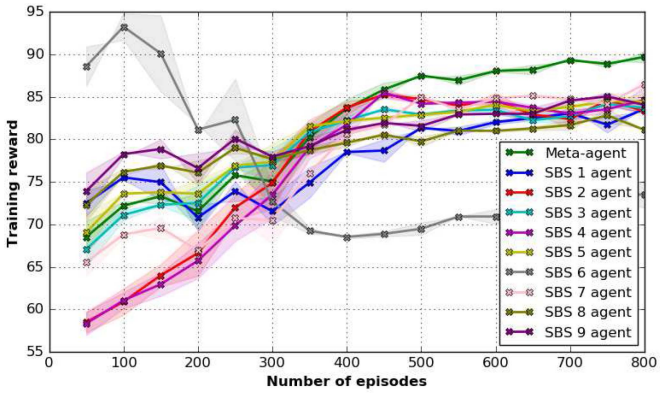


Fig. 6. Reward value achieved for proposed Meta-RL training of the meta-agent alone with other SBS agents.

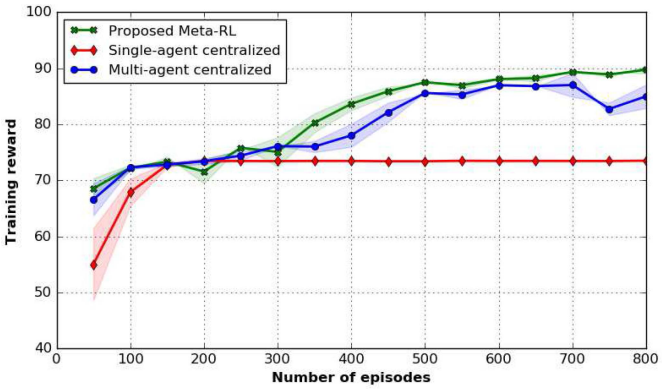
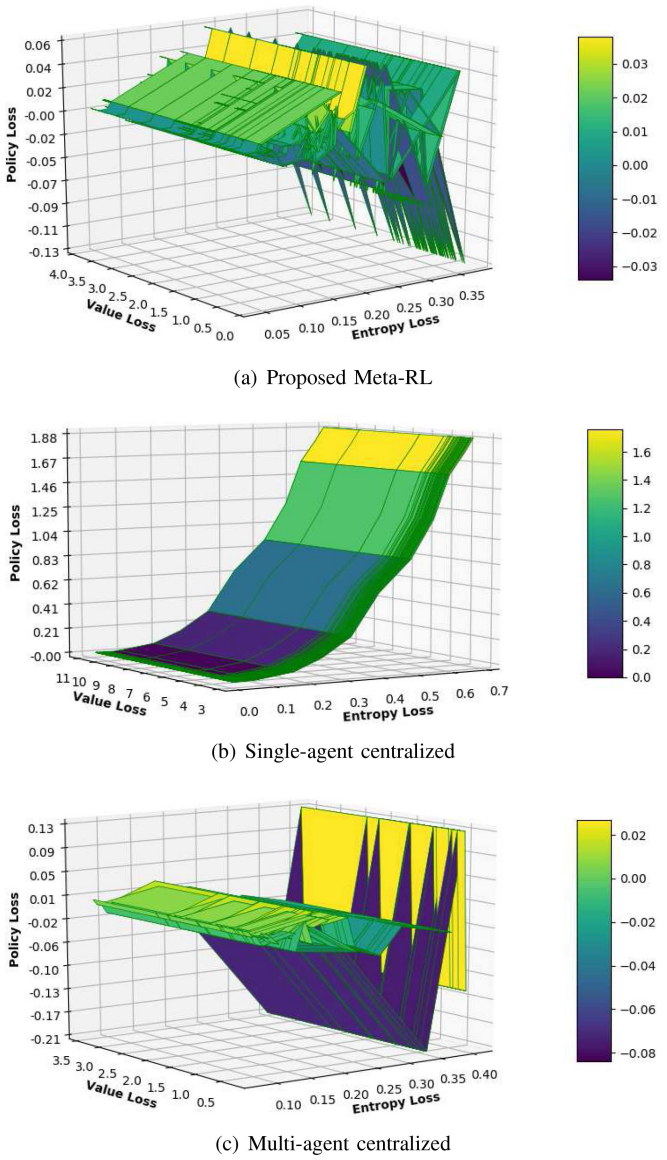


Fig. 7. Reward value achieved of proposed Meta-RL, single-agent centralized, and multi-agent centralized methods.



(a) Proposed Meta-RL

(b) Single-agent centralized

(c) Multi-agent centralized

Fig. 8. Relationship among the entropy loss, value loss, and policy loss in the training phase of proposed Meta-RL, single-agent centralized, and multi-agent centralized methods.

reward value except the SBS 6 that achieves a reward of 70 at convergence because its energy consumption and generation vary more than the others. In fact, this variation of reward among the BSs is leading to anticipate the non-i.i.d. energy demand and generation of the considered network as well as densification of the exploration and exploitation tradeoff for energy dispatch. The proposed approach can adapt the uncertain energy demand and generation over time by characterizing the expected amount of uncertainty in an energy dispatch decision for each BS $i \in \mathcal{B}$ individually. Meanwhile, the meta-agent exploits the energy dispatch decision by employing a joint policy toward the globally optimal energy dispatch for each BS $i \in \mathcal{B}$. Therefore, the challenges of distinct energy demand and generation of the state space among the BSs can be efficiently handled by applying learned parameters from the meta-agent to each BS $i \in \mathcal{B}$ during the training that establishes a balance between exploration and exploitation.

We compare the achieved reward of proposed MAMRL model with single-agent centralized and multi-agent centralized models in Fig. 7. The single agent centralized (diamond mark with red line) model converges faster than the other two models but it achieves the lowest reward due to the lack of exploitation as it has only one agent. Further, the multi-agent centralized (circle mark with blue line) model converges with a higher reward than the single agent method. The proposed MAMRL (cross mark with green line) model outperforms the other two models while converges with the highest reward value. In addition, multi-agent centralized needs the entire state information. In contrast, the meta-agent requires only the observation from local agents, and it can optimize the neural network parameters by using its own state information.

We analyze the relationship among the value loss, entropy loss, and policy loss in Fig. 8, where the maximum policy loss of the proposed MAMRL (in 8(a)) model is around 0.06 whereas single-agent centralized (in 8(b)) and multi-agent centralized (in 8(c)) methods gain about 1.88 and 0.12, respectively. Therefore, the training accuracy increases due to more variation between exploration and exploitation. Thus, our MAMRL model is capable of incorporating the decision of each local BS agent that solves the challenge of non-i.i.d. demand-generation for the other BSs.

In Fig. 9, we examine the testing accuracy [69] of the storage energy $\xi_i^{\text{sto}}(t)$ and the non-renewable energy generation

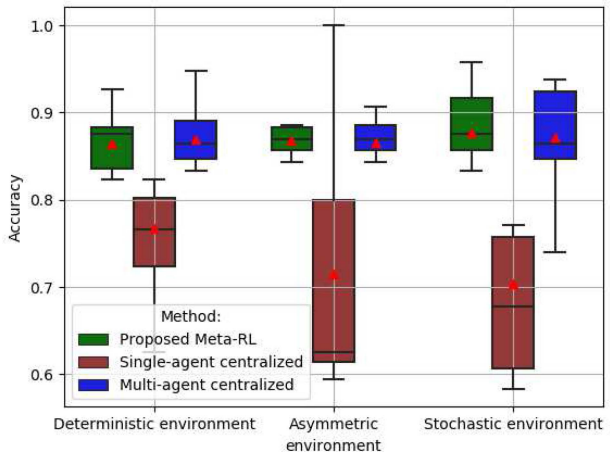


Fig. 9. Testing accuracy of the proposed Meta-RL, single-agent centralized, and multi-agent centralized methods with deterministic, asymmetric, and stochastic environments of the 9 SBSs.

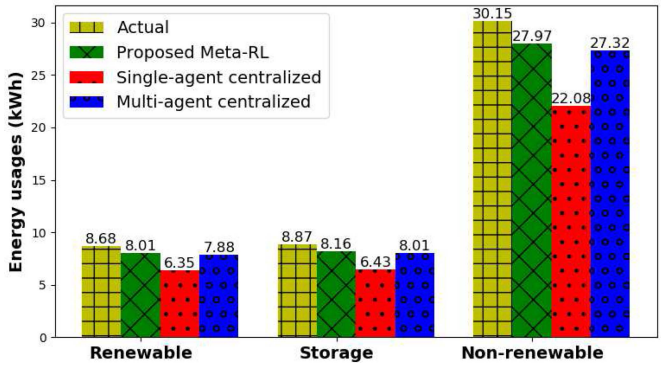


Fig. 10. Prediction result of renewable, storage, and non-renewable energy usages of a single SBS (SBS 2) for 24 hours (96 time slots) under the stochastic environment.

decision[6] $\xi_i^{\text{non}}(t)$ for 96 time slots (1 days) of 9 SBSs under the deterministic, asymmetric, and stochastic environments. In the experiment, we have used the well-known UMass solar panel dataset [61] for renewable energy generation information as well as, the CRAWDAD nyupoly/video dataset [60], for estimating the energy consumption of the self-powered network. Further, we preprocess both of the datasets ([60] and [61]) using Algorithm 1 that generates the state space information. Thus, the *ground truth* comes from this state-space information of the considered datasets, where the actions are depended on the renewable energy generation and consumption of a particular BS $i \in \mathcal{B}$. The proposed MAMRL (green box) and multi-agent centralized (blue box) methods achieve a maximum accuracy of around 95% and 92%, respectively, under the stochastic environment (in Fig. 9). Further, Fig. 9 shows that the mean accuracy (88%) of the proposed method is also higher than the centralized solution (86%). Similarly, in the deterministic and asymmetric environment,

[6]Each BS agent $i \in \mathcal{B}$ can calculate its action from a globally optimal energy dispatch policy $\pi^*_{\theta_i}$ by using $\text{argmax}(.)$ (i.e., $\text{argmax}_{\pi^*_{\theta_i}}(\boldsymbol{a}_{ti})$). In which, at the end of 15 minutes duration of each time slot t, the each BS agent $i \in \mathcal{B}$ can choose one action (i.e., storage or non-renewable) from the energy dispatch policy $\pi^*_{\theta_i}$.

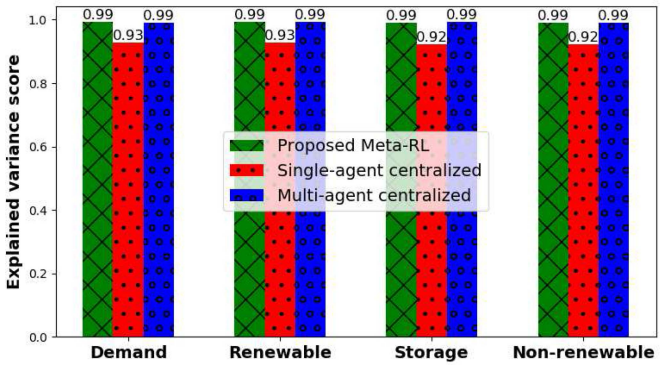


Fig. 11. Explained variance score of a single SBS (SBS 2) for 24 hours (96 time slots) under the stochastic environment.

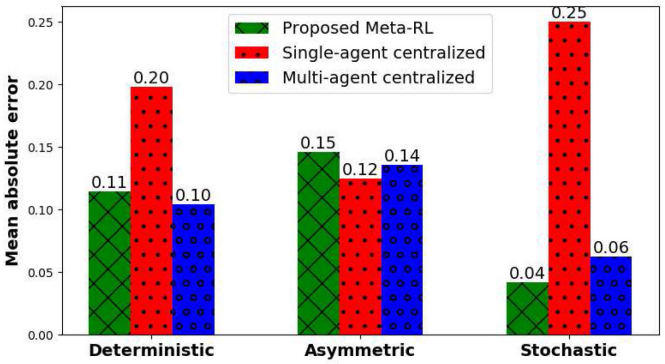


Fig. 12. Mean absolute error of a single SBS (SBS 2) for 24 hours (96 time slots) under the stochastic environment.

the average accuracy (around 87%) of the proposed low complexity semi-distributed solution is almost the same as the baseline method.

The prediction results of renewable, storage and non-renewable energy usage for a single SBS (SBS 2) for 24 hours (96 time slots) under the stochastic environment are shown in Fig. 10. The proposed MAMRL outperforms all other baselines and achieves an accuracy of around 95.8%. In contrast, the accuracy of the other two methods is 75% and 93.7% for the single-agent centralized and multi-agent centralized, respectively.

In Figs. 11 and 12, we validate our approach with two standard regression model evaluation metric, explained variance[7] (i.e., explained variation) and mean absolute error (MAE) [69], respectively. Fig. 11 shows that the explained variance score of the proposed MAMRL method almost the same as the multi-agent centralized. However, in the case of renewable energy generation, MAMRL method significantly performs better (i.e., 1% more score) than the multi-agent centralized solution. In particular, the proposed MAMRL model has pursued the uncertainty of renewable energy generation by the dynamics of Markovian for each BS. Further, meta-agent anticipates the energy dispatch by other BSs decisions and its

[7]We measure the discrepancy for energy dispatch decisions between the proposed and baseline models on the ground truth of the datasets ([60] and [61]). We deploy the explained variance regression score function using *sklearn* API [70] to measure and compare this discrepancy.

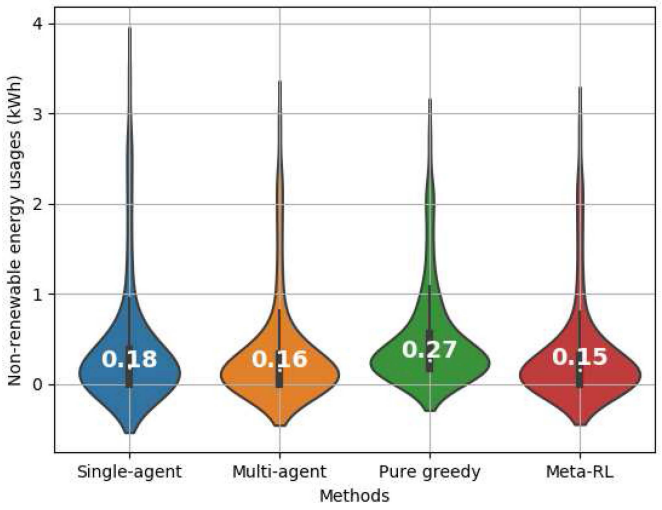


Fig. 13. Kernel density analysis of non-renewable energy usages for 24 hours (96 time slots) under the stochastic environment.

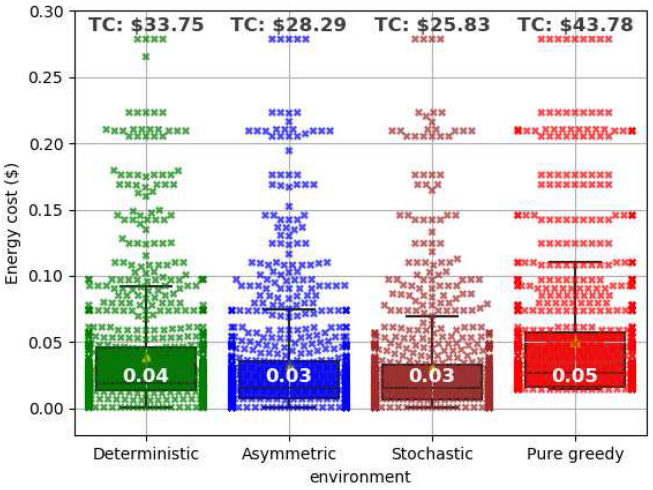


Fig. 14. Energy consumption cost analysis of 9 SBSs for 24 hours (96 time slots) under deterministic, asymmetric, and stochastic environments using the proposed Meta-RL method over pure greedy method.

own state information. We analyze the MAE[8] for the three environments (i.e., deterministic, asymmetric, and stochastic) among the proposed MAMRL, single-agent centralized, and multi-agent centralized methods in Fig. 12. The MAE of the proposed MAMRL is 11%, 15%, and 4% for deterministic, asymmetric, and stochastic, respectively since meta-agent has the capability to adopt the uncertain environment very fast. This adaptability is enhanced by the exploration mechanism that is taken into account at each BS, and exploitation that performs by capitalizing the non-i.i.d. explored information of all BSs.

Fig. 13 illustrates the efficacy of the proposed MAMRL model in terms of the non-renewable energy usages into a stochastic environment with other benchmarks. This figure considers a kernel density analysis for 24 hours (96 time slots) under a stochastic environment, where the median of the non-renewable energy usages 0.15 (kWh), and 0.27 (kWh) for the proposed MAMRL, and pure greedy, respectively, at each 15 minutes time slot. Further, the proposed MAMRL can significantly reduce the usages of non-renewable energy for the considered self-powered wireless network, where the MAMRL can save up to 13.3% of the non-renewable energy usages. Here, the meta agent of the MAMRL model can discretize uncertainty from each local BS agent and transfer the knowledge (i.e., learning parameters) to each local agent that can take a globally optimal energy dispatch decision.

Fig. 14 presents the energy consumption cost analysis for 9 SBSs over 24 hours (96 time slots) under deterministic, asymmetric, and stochastic environments using the proposed Meta-RL method while comparing it to the pure greedy method. The total energy cost achieved by the proposed approach for a particular day will be \$33.75, \$28.29, and \$25.83 for deterministic, asymmetric, and stochastic environments, respectively. Fig. 14 also shows that the proposed method

[8]This performance metric provides us with the average magnitude of errors for the energy dispatch decision of a single SBS (SBS 2) for 24 hours (96 time slots). Particularly, we analyze the average error over the 96 time slots of the absolute differences between prediction and actual observation. To evaluate this metric, we have used the mean absolute error regression loss function of *sklearn* API [71].

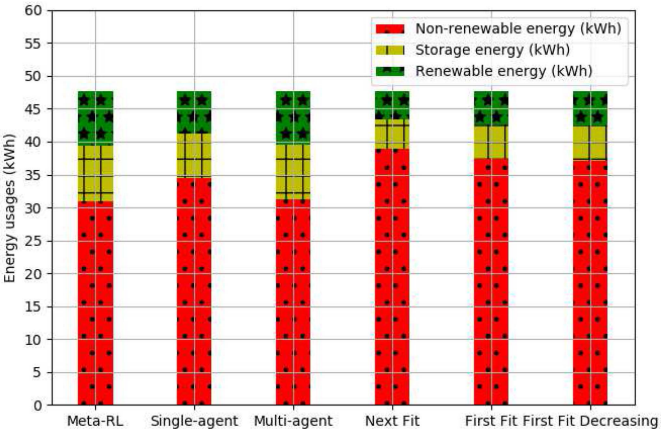


Fig. 15. Amount of renewable, non-renewable, and storage energy estimation for 24 hours (96 time slots) for proposed meta-RL, single-agent RL, multi-agent RL, next fit, first fit, and first fit decreasing methods.

significantly reduces the energy consumption cost (by at least 22.4%) for all three environments over the pure greedy method. The median of the energy cost at each time slot is \$0.04, \$0.03, and \$0.03 for the deterministic, asymmetric, and stochastic environments, respectively. In contrast, Fig. 14 has shown that a median energy cost for the pure greedy baseline is \$0.05 at each time slot that is due to a lack of the competence to cope with an unknown environment for energy consumption and generation. Therefore, the proposed MAMRL model can overcome the challenges of an unknown environment as well as non-i.i.d. characteristics for energy consumption and generation of a self-powered MEC network.

In Fig. 15, we compare our proposed meta-RL model with single-agent RL, multi-agent RL, next fit, first fit, and first fit decreasing methods in terms of amount of renewable, non-renewable, and storage energy usages for 24 hours (96 time slots). Fig. 15 shows that the proposed MAMRL model outperforms the others that achieves around 22% less non-renewable energy usages than the next fit scheduling algorithm. Additionally, next fit, first fit, and first fit decreasing scheduling methods [72] cannot capture the uncertainty of energy generation and consumption, as well as provide a near

TABLE V
COMPARISON BETWEEN THE PROPOSED METHOD AND OTHER METHODS WITH GROUND TRUTH FOR A SINGLE SBS (SBS 2) FOR 24 HOURS (96 TIME SLOTS) UNDER THE STOCHASTIC ENVIRONMENT

Method	**Non-renewable energy usage (kWh)**	**Storage energy usage (kWh)**	**Renewable energy usage (kWh)**	**Non-renewable energy usage cost ($)**	**Storage energy usage cost ($)**	**Renewable energy usage cost ($)**	**Total energy usage cost ($)**	**Cost difference with ground truth (%)**
Ground truth (i.e., optimal)	30.15	8.87	8.67	3.07	0.49	0.43	3.99	NA
MAMRL (proposed)	**30.88**	**8.50**	**8.32**	**3.14**	**0.47**	**0.42**	**4.03**	**0.90**
Single-agent RL	34.53	6.65	6.50	3.52	0.37	0.33	4.21	5.43
Multi-agent RL	31.24	8.31	8.14	3.19	0.46	0.41	4.05	1.36
Next Fit	38.92	4.44	4.34	3.97	0.24	0.21	4.43	10.86
First Fit	37.37	5.22	5.10	3.81	0.29	0.26	4.35	8.94
First Fit Decreasing	37.12	5.34	5.23	3.79	0.30	0.26	4.34	8.63
Without renewable	47.69	NA	NA	4.86	NA	NA	4.86	21.72

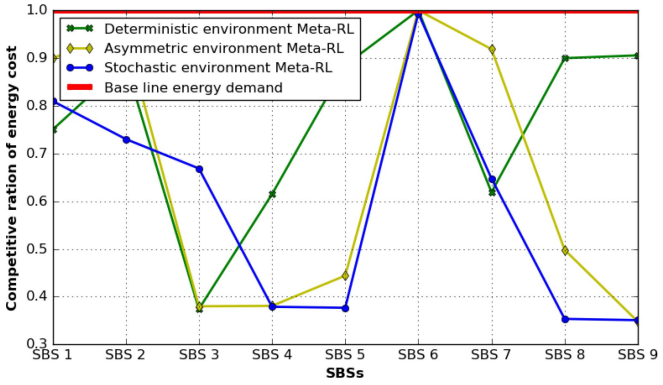


Fig. 16. Competitive cost ratio of the proposed Meta-RL method for 24 hours (96 time slots) under the deterministic, asymmetric, and stochastic environments.

optimal solution. Further, a comparison between the proposed method and other methods with the ground truth for a single SBS (SBS 2) for 24 hours (96 time slots) under the stochastic environment is illustrated in Table V. The proposed method can achieve significant outcomes with respect to energy cost as compared with the ground truth. In particular, the experiment shows that the energy usage cost difference between the proposed method and ground truth is around 1% for a single BS (in Table V). This leads to one of the evidence that the proposed MAMRL can adopt the unknown environment and can utilize it during the execution for each BS energy dispatch.

Finally, in Fig. 16, we examine the competitive cost ratio [29] of the proposed MAMRL framework. From this figure, we observe that the proposed MAMRL framework effectively minimizes the energy consumption cost for each BS under deterministic, asymmetric, and stochastic environments. In fact, Fig. 16 ensures the robustness of the proposed MAMRL framework that is performed a tremendous performance gain by coping with non-i.i.d. energy consumption and generation under the uncertainty. Furthermore, in MAMRL training, each local agent has captured the time-variant features of energy demand and generation from the historical data while meta-agent optimizes energy dispatch decisions by obtaining those features with its own parameters of LSTM. In the case of testing, a generalized MAMRL trained model is employed that makes a fully independent and unbiased energy dispatch from an unknown environment. To this end, the proposed MAMRL framework shows the efficacy of solving the energy dispatch of a self-powered wireless network with MEC capabilities with a higher degree of reliability.

VI. CONCLUSION

In this article, we have investigated an energy dispatch problem of a self-powered wireless network with MEC capabilities. We have formulated a two-stage stochastic linear programming energy dispatch problem for the considered network. To solve the energy dispatch problem in a semi-distributed manner, we have proposed a novel multi-agent meta-reinforcement learning framework. In particular, each local BS agent obtains the time-varying features by capturing the Markovian properties of the network's energy consumption and renewable generation for each BS unit, and predict its own energy dispatch policy. Meanwhile, a meta-agent optimizes each BS agent's energy dispatch policy from its own state information, and it transfers global learning parameters to each BS agent so that they can update their energy dispatch policy into an optimal policy. We have shown that the proposed MAMRL framework can capture the uncertainty of non-i.i.d. energy demand and generation for the self-powered wireless network with MEC capabilities. Our experimental results have shown that the proposed MAMRL framework can save a significant amount of non-renewable energy with higher accuracy prediction that ensures the energy sustainability of the network. In particular, the performance of energy dispatch over deterministic, asymmetric, and stochastic environments outperform other baseline approaches, where average accuracy achieves up to 95.8% and reduces the energy cost about 22.4% of the self-powered wireless network. To this end, the proposed MAMRL model can reduce by at least 11% of the non-renewable energy usage for the self-powered wireless network.

APPENDIX A
EXAMPLE OF INFORMATION EXCHANGE BETWEEN LOCAL BS AGENT AND META AGENT IN MAMRL FRAMEWORK

For example, consider an LSTM cell with 48 LSTM units [49], [64]. Thus, the dimension of forget gate, input gate, gate/memory/activation gate, and output gate will be 48

for each gate. Now consider a local BS agent $i \in \mathcal{B}$ that embedding a dimension of 3 inputs $(r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, t')$ to a local LSTM cell. This input comes from the state information $\boldsymbol{s}_{ti} : (\xi_i^{\mathrm{d}}, \xi_i^{\mathrm{ren}}(t), C_{ti}^{\mathrm{sto}}, C_{ti}^{\mathrm{non}})$ of a local BS agent *i*. As a result, inputs are appended to all gates during the training. Therefore, the number of learning parameters will be $4 \times (48(48+3)+48)$ (i.e., $gates \times [units(units+input)+units]$). Additionally, the size of hidden state and cell state parameters remain 48 for each due to an LSTM cell with 48 LSTM units. Further, on top of the LSTM cell, we have two fully connected output layers, a fully connected output layer with a Softmax activation to determine the local energy dispatch policy. Meanwhile, advantage is determined from another fully connected output layer without activation function by value function estimation. The hidden and cell state of each local agent are updated by receiving the state parameters with a 48×2 dimensional data from the meta-agent. In case of the meta-agent, the configuration of LSTM cell is the same as each local LSTM cell. Therefore, at the end of each time slot duration, the meta-agent sends a 48×2 dimensional state information to each local BS agent $i \in \mathcal{B}$. Subsequently, the meta-agent receives a 6 dimensional observation $\boldsymbol{o}_i : (r_i(\boldsymbol{a}_{t'i}, \boldsymbol{s}_{t'i}), r_i(\boldsymbol{a}_{ti}, \boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}, \boldsymbol{a}_{t'i}, t', \Lambda^{\pi_{\theta_i}}(\boldsymbol{s}_{ti}, \boldsymbol{a}_{ti}))$ as an input from each local BS agent, where the number of learning parameters at the meta-agent will be $4 \times (48(48+6)+48)$ for each iteration. The output layer of the meta-agent also consists of two fully connected output layers for determining meta-policy (i.e., joint policy) and meta advantage. Thus, these output layers do not affect the dimension of hidden and cell states for the meta agent's LSTM cell. In fact, these RNN states are used as an input to these fully connected layers. As a result, for each epoch (i.e., end of a time slot duration), the meta-agent sends 48×2 dimensional RNN states to each local agent along with an energy dispatch policy, and each local agent sends 6 dimensional observation to the meta-agent.

APPENDIX B
PROOF OF PROPOSITION 1

Proof: For a BS agent *i*, energy dispatch policy $\pi^*_{\theta_i}$ is the best response for the equilibrium responses from all other BS agents. Thus, the BS agent *i* can not be improved the value $V^{\pi^*_{\theta_i}}(\boldsymbol{s}_{ti})$ any more by deviating of policy $\pi^*_{\theta_i}$. Therefore, (24) holds the following property,

$$\begin{aligned} V^{\pi^*_{\theta_i}}(\boldsymbol{s}_{ti}) \geq\ & r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \\ & + \sum_{\boldsymbol{s}_{t'i} \in \mathcal{S}_i, t'=t}^{\infty} \gamma^{t'-t} \Gamma(\boldsymbol{s}_{t'i} | \boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \\ & \times\ V^{\pi_{\theta_i}}\left(\boldsymbol{s}_{t'i}, \pi^*_{\theta_0}, \ldots, \pi^*_{\theta_B}\right). \end{aligned} \tag{30}$$

Hence, the meta-agent $M_t(\mathcal{O}_t; \phi)$ of the $|\mathcal{B}|$-agent energy dispatch model (i.e., MAMRL) reaches a Nash equilibrium point for policy $\pi^*_{\theta_i}$ with parameters θ_i. As a result, the optimal value of BS agent $i \in \mathcal{B}$ can be as follows:

$$V^{\pi^*_{\theta_i}}(\boldsymbol{s}_{ti}) = M_t\big(\nabla_{\theta_t} L(\theta_t); \phi\big). \tag{31}$$

(31) implies that $\pi^*_{\theta_i}$ is an optimal policy of energy dispatch decisions. Thus, the optimal policy $\pi^*_{\theta_i}$ belongs to a Nash equilibrium point and holds the following inequality,

$$V^{\pi^*_{\theta_i}}(\boldsymbol{s}_{ti}) \geq \mathbb{E}_{L(\theta)}[L(\theta^*(L(\theta); \phi))] \tag{32}$$

■

APPENDIX C
PROOF OF PROPOSITION 2

Proof: A probability of action $\boldsymbol{a}_{ti}$ of BS agent $i \in \mathcal{B}$ at time *t* can be presented as follows:

$$\begin{aligned} P(\boldsymbol{a}_{ti}) &= \theta_i^{\boldsymbol{a}_{ti}} (1-\theta_i)^{1-\boldsymbol{a}_{ti}} \\ &= \boldsymbol{a}_{ti} \log \theta_i + (1-\boldsymbol{a}_{ti}) \log(1-\theta_i). \end{aligned} \tag{33}$$

We consider a single state, and a policy gradient estimator can be defined as,

$$\begin{aligned} \frac{\hat{\partial}}{\partial \theta_i} L(\theta_i) &= r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \frac{\partial}{\partial \theta_i} \log P(\boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \\ &= r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \frac{\partial}{\partial \theta_i} \sum_{\forall i \in \mathcal{B}} \boldsymbol{a}_{ti} \log \theta_i \\ &\quad + (1-\boldsymbol{a}_{ti}) \log(1-\theta_i) \\ &= r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \frac{\partial}{\partial \theta_i} \\ &\quad \times (\boldsymbol{a}_{ti} \log \theta_i + (1-\boldsymbol{a}_{ti}) \log(1-\theta_i)) \\ &= r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) \left(\frac{\boldsymbol{a}_{ti}}{\theta_i} - \frac{(1-\boldsymbol{a}_{ti})}{(1-\theta_i)} \right) \\ &= r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB})(2\boldsymbol{a}_{ti} - 1), \ \text{for } \theta_i = 0.5. \end{aligned} \tag{34}$$

Thus, an expected reward for $|\mathcal{B}|$ BS agents can be represented as, $\mathbb{E}[r_i] = \sum_{\forall i \in \mathcal{B}} r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB})(0.5)^{|\mathcal{B}|}$, where by applying $r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB}) = \mathbf{1}|r_i(\boldsymbol{s}_{ti}, \boldsymbol{a}_{t0}, \ldots, \boldsymbol{a}_{tB})$, we can get $\mathbb{E}[r_i] = (0.5)^{|\mathcal{B}|}$. Now, we can define an expectation of a gradient estimation as, $\mathbb{E}[\frac{\hat{\partial}}{\partial \theta_i} L(\theta_i)] = \frac{\partial}{\partial \theta_i} L(\theta_i) = (0.5)^{|\mathcal{B}|}$. Therefore, a variance of the estimated gradient can be defined as,

$$\begin{aligned} \mathbb{V}\left[\frac{\hat{\partial}}{\partial \theta_i} L(\theta_i) \right] &= \mathbb{E}\left[\frac{\hat{\partial}}{\partial \theta_i} L^2(\theta_i) \right] - \left(\mathbb{E}\left[\frac{\hat{\partial}}{\partial \theta_i} L(\theta_i) \right] \right)^2 \\ &= (0.5)^{|\mathcal{B}|} - (0.5)^{2|\mathcal{B}|}. \end{aligned} \tag{35}$$

Now, we can analyze the step of gradient for $P((\hat{\nabla}_{\theta_i} L(\theta_i), \nabla_{\theta_i} L(\theta_i)) > 0)$ (in (29)), where

$$P\Big(\hat{\nabla}_{\theta_i} L(\theta_i), \nabla_{\theta_i} L(\theta_i)\Big) = (0.5)^{|\mathcal{B}|} \sum_{\forall i \in \mathcal{B}} \frac{\hat{\partial}}{\partial \theta_i} L(\theta_i). \tag{36}$$

As a result, $P((\hat{\nabla}_{\theta_i} L(\theta_i), \nabla_{\theta_i} L(\theta_i)) > 0) = (0.5)^{|\mathcal{B}|}$ implies that the gradient step not only moves in the correct direction but also decreases exponentially with an increasing number of BS agents. ■

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020, doi: 10.1109/MNET.001.1900287.

[2] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019, doi: 10.1109/JPROC.2019.2941458.

[3] E. Dahlman, S. Parkvall, J. Peisa, H. Tullberg, H. Murai, and M. Fujioka, "Artificial intelligence in future evolution of mobile communication," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Okinawa, Japan, Feb. 2019, pp. 102–106.

[4] M. S. Munir, S. F. Abedin, and C. S. Hong, "Artificial intelligence-based service aggregation for mobile-agent in edge computing," in *Proc. 20th Asia–Pac. Netw. Oper. Manag. Symp. (APNOMS)*, Matsue, Japan, Sep. 2019, pp. 1–6.

[5] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[6] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021, doi: 10.1109/TWC.2020.3024629.

[7] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 1387–1395.

[8] G. Lee, W. Saad, M. Bennis, A. Mehbodniya, and F. Adachi, "Online SKI rental for ON/OFF scheduling of energy harvesting base stations," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2976–2990, May 2017.

[9] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor–critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Jan. 2018.

[10] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.

[11] M. S. Munir, S. F. Abedin, N. H. Tran, and C. S. Hong, "When edge computing meets microgrid: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7360–7374, Oct. 2019.

[12] M. S. Munir, S. F. Abedin, D. H. Kim, N. H. Tran, Z. Han, and C. S. Hong, "A multi-agent system toward the green edge computing with microgrid," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–7.

[13] N. Piovesan, D. A. Temesgene, M. Miozzo, and P. Dini, "Joint load control and energy sharing for autonomous operation of 5G mobile networks in micro-grids," *IEEE Access*, vol. 7, pp. 31140–31150, 2019.

[14] X. Huang, T. Han, and N. Ansari, "Smart grid enabled mobile networks: Jointly optimizing BS operation and power distribution," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1832–1845, Jun. 2017.

[15] W. Li *et al.*, "On enabling sustainable edge computing with renewable energy resources," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 94–101, May 2018.

[16] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[17] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[18] P. Chang and G. Miao, "Resource provision for energy-efficient mobile edge computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, 2018, pp. 1–6.

[19] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.

[20] S. F. Abedin, M. G. R. Alam, R. Haw, and C. S. Hong, "A system model for energy efficient green-IoT network," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2015, pp. 177–182.

[21] X. Zhang, M. R. Nakhai, G. Zheng, S. Lambotharan, and B. Ottersten, "Calibrated learning for online distributed power allocation in small-cell networks," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 8124–8136, Nov. 2019, doi: 10.1109/TCOMM.2019.2938514.

[22] S. Akin and M. C. Gursoy, "On the energy and data storage management in energy harvesting wireless communications," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 8056–8071, Nov. 2019.

[23] N. H. Tran, C. Pham, M. N. H. Nguyen, S. Ren, and C. S. Hong, "Incentivizing energy reduction for emergency demand response in multi-tenant mixed-use buildings," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3701–3715, Jul. 2018.

[24] J. X. Wang *et al.*, "Learning to reinforcement learn," in *Proc. CogSci*, 2017, p. 1319.

[25] N. Schweighofera and K. Doya, "Meta-learning in reinforcement learning," *Neural Netw.*, vol. 16, no. 1, pp. 5–9, Jan. 2003.

[26] M. Andrychowicz *et al.*, "Learning to learn by gradient descent by gradient descent," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 3981–3989.

[27] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, Jan. 2016, pp. 1928–1937.

[28] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor–critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.

[29] Y. Zhang, M. H. Hajiesmaili, S. Cai, M. Chen, and Q. Zhu, "Peak-aware online economic dispatching for microgrids," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 323–335, Jan. 2018.

[30] T. Han and N. Ansari, "Network utility aware traffic load balancing in backhaul-constrained cache-enabled small cell networks with hybrid power supplies," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2819–2832, Oct. 2017.

[31] S. F. Abedin, A. K. Bairagi, M. S. Munir, N. H. Tran, and C. S. Hong, "Fog load balancing for massive machine type communications: A game and transport theoretic approach," *IEEE Access*, vol. 7, pp. 4204–4218, 2018.

[32] Z. Chang, Z. Zhou, T. Ristaniemi, and Z. Niu, "Energy efficient optimization for computation offloading in fog computing system," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[33] A. Ndikumana *et al.*, "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1359–1374, Jun. 2020, doi: 10.1109/TMC.2019.2908403.

[34] T. Rauber, G. Rünger, M. Schwind, H. Xu, and S. Melzner, "Energy measurement, modeling, and prediction for processors with frequency scaling," *J. Supercomput.*, vol. 70, no. 3, pp. 1454–1476, 2014.

[35] R. Bertran, M. Gonzalez, X. Martorell, N. Navarro, and E. Ayguade, "A systematic methodology to generate decomposable and responsive power models for CMPs," *IEEE Trans. Comput.*, vol. 62, no. 7, pp. 1289–1302, 1st Quart., 2016.

[36] G. Auer *et al.*, "How much energy is needed to run a wireless network?" *IEEE Wireless Commun.*, vol. 18, no. 5, pp. 40–49, Oct. 2011.

[37] ETSI, *5G; NR; Physical Layer Procedures for Data*. Accessed: Jul. 18, 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138214/15.03.00_60/ts_138214v150300p.pdf

[38] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.

[39] F. Pantisano, M. Bennis, W. Saad, S. Valentin, and M. Debbah, "Matching with externalities for context-aware user-cell association in small cell networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Atlanta, GA, USA, 2013, pp. 4483–4488.

[40] N. L. Panwar, S. C. Kaushik, and S. Kothari, "Role of renewable energy sources in environmental protection: A review," *Renew. Sustain. Energy Rev.*, vol. 15, no. 3, pp. 1513–1524, Apr. 2011.

[41] F. A. Chacra, P. Bastard, G. Fleury, and R. Clavreul, "Impact of energy storage costs on economical performance in a distribution substation," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 684–691, May 2005.

[42] H. Kanchev, D. Lu, F. Colas, V. Lazarov, and B. Francois, "Energy management and operational planning of a microgrid with a PV-based active generator for smart grid applications," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4583–4592, Oct. 2011.

[43] A. Mishra, D. Irwin, P. Shenoy, J. Kurose, and T. Zhu, "GreenCharge: Managing renewable energy in smart buildings," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1281–1293, Jul. 2013.

[44] X. Xu, Z. Yan, M. Shahidehpour, Z. Li, M. Yan, and X. Kong, "Data-driven risk-averse two-stage optimal stochastic scheduling of energy and reserve with correlated wind power," *IEEE Trans. Sustain. Energy*, vol. 11, no. 1, pp. 436–447, Jan. 2020, doi: 10.1109/TSTE.2019.2894693.

[45] Business Insider. (May 2018). *One Simple Chart Shows Why an Energy Revolution Is Coming*. Accessed: Jul. 23, 2019. [Online]. Available: https://www.businessinsider.com/solar-power-cost-decrease-2018-5

[46] Y. Liu and N. K. C. Nair, "A two-stage stochastic dynamic economic dispatch model considering wind uncertainty," *IEEE Trans. Sustain. Energy*, vol. 7, no. 2, pp. 819–829, Apr. 2016.

[47] D. Zhou, M. Sheng, B. Li, J. Li, and Z. Han, "Distributionally robust planning for data delivery in distributed satellite cluster network," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3642–3657, Jul. 2019.

[48] S. F. Abedin, M. G. R. Alam, S. M. A. Kazmi, N. H. Tran, D. Niyato, and C. S. Hong, "Resource allocation for ultra-reliable and enhanced mobile broadband IoT applications in fog network," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 489–502, Jan. 2019.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] Z. M. Fadlullah *et al.*, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, 4th Quart., 2017.
[51] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, vol. 1, 2nd ed. Cambridge, MA, USA: MIT Press, 2017.
[52] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 157–163.
[53] R. J. Williams, and D. Zipser, "Gradient-based learning algorithms for recurrent," in *Backpropagation: Theory, Architectures, and Applications*, vol. 433. New York, NY, USA: Psychology, 1995.
[54] I. Bialynicki-Birula and J. Mycielski, "Uncertainty relations for information entropy in wave mechanics," *Commun. Math. Phys.*, vol. 44, pp. 129–132, Jun. 1975.
[55] T. Seidenfeld, "Entropy and uncertainty," *Philosophy Sci.*, vol. 53, no. 4, pp. 467–491, 1986.
[56] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "An asynchronous mini-batch algorithm for regularized stochastic optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3740–3754, Dec. 2016, doi: 10.1109/TAC.2016.2525015.
[57] A. Agarwal and J. C. Duchi, "The generalization ability of online algorithms for dependent data," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 573–587, Jan. 2013, doi: 10.1109/TIT.2012.2212414.
[58] A. M. Fink, "Equilibrium in a stochastic *n*-person game," *J. Sci. Hiroshima Univ. A-I (Math.)*, vol. 28, no. 1, pp. 89–93, 1964.
[59] P. J. Herings and R. J. A. P. Peeters, "Stationary equilibria in stochastic games: Structure, selection, and computation," *J. Econ. Theory*, vol. 118, no. 1, pp. 32–60, Sep. 2004.
[60] S. Fu and Y. Zhang. (Apr. 2015). *CRAWDAD Dataset Due/Packet-Delivery (V. 2015-04-01)*. Accessed: Jul. 3, 2019. [Online]. Available: https://crawdad.org/due/packet-delivery/20150401
[61] *Solar Panel Dataset*. Accessed: Jul. 3, 2019. [Online]. Available: http://traces.cs.umass.edu/index.php/Smart/Smart
[62] A. K. Bairagi, N. H. Tran, W. Saad, Z. Han, and C. S. Hong, "A game-theoretic approach for fair coexistence between LTE-U and Wi-Fi systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 442–455, Jan. 2019.
[63] Intel. *Intel Core i7-6500U Processor*. Accessed: Aug. 17, 2019. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/88194/intel-core-i7-6500u-processor-4m-cache-up-to-3-10-ghz.html
[64] *TensorFlow Core V2.2.0*. Accessed: May 27, 2020. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/compat/v1/nn/rnn_cell/BasicLSTMCell
[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–41.
[66] Y. Takahashi, G. Schoenbaum, and Y. Niv, "Silencing the critics: Understanding the effects of cocaine sensitization on dorsolateral and ventral striatum in the context of an actor/critic model," *Front. Neurosci.*, vol. 2, pp. 86–89, Jul. 2009.
[67] C. G. Li, M. Wang, and Q. N. Yuan, "A multi-agent reinforcement learning using actor–critic methods," in *Proc. Int. Conf. Mach. Learn. Cybern. Kunming*, 2008, pp. 878–882, doi: 10.1109/ICMLC.2008.4620528.
[68] *All Symbols in TensorFlow*. Accessed: Jul. 3, 2019. [Online]. Available: https://www.tensorflow.org/api_docs/python/
[69] *Model Evaluation: Quantifying the Quality of Predictions, Scikit-Learn*. Accessed: Aug. 3, 2019. [Online]. Available: http://scikit-learn.org/stable/modules/model_evaluation.html
[70] *Explained Variance Regression Score Function, Scikit-Learn*. Accessed: May 28, 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.explained_variance_score.html
[71] *Mean Absolute Error Regression Loss Scikit-Learn*. Accessed: May 28, 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html
[72] D. S. Johnson, *Near-Optimal Bin Packing Algorithms*, Massachusetts Inst. Technol., Cambridge, MA, USA, 1973.

Md. Shirajum Munir (Graduate Student Member, IEEE) received the B.S. degree in computer science and engineering from Khulna University, Khulna, Bangladesh, in 2010. He is currently pursuing the Ph.D. degree in computer science and engineering with Kyung Hee University, Seoul, South Korea. He served as a Lead Engineer with the Solution Laboratory, Samsung Research and Development Institute, Dhaka, Bangladesh, from 2010 to 2016. His current research interests include IoT network management, fog computing, mobile edge computing, software-defined networking, smart grid, and machine learning.

Nguyen H. Tran (Senior Member, IEEE) received the B.S. degree from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2005, and the Ph.D. degree in electrical and computer engineering from Kyung Hee University, Seoul, South Korea, in 2011. Since 2018, he has been with the School of Computer Science, University of Sydney, Sydney, NSW, Australia, where he is currently a Senior Lecturer. He was an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University from 2012 to 2017. His current research interests include applying analytic techniques of optimization, game theory, and stochastic modeling to cutting-edge applications, such as cloud and mobile edge computing, data centers, heterogeneous wireless networks, and big data for networks. He was a recipient of the Best KHU Thesis Award in Engineering in 2011 and the Best Paper Award of IEEE ICC 2016. He has been an Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING since 2016, and served as the Editor of the 2017 Newsletter of Technical Committee on Cognitive Networks on Internet of Things.

Walid Saad (Fellow, IEEE) received the Ph.D. degree from the University of Oslo in 2010. He is currently a Professor with the Department of Electrical and Computer Engineering, Virginia Tech, where he leads the Network Science, Wireless, and Security (NEWS) Laboratory. From 2015 to 2017, he was named the Stephen O. Lane Junior Faculty Fellow with Virginia Tech and he was named as the College of Engineering Faculty Fellow in 2017. His research interests include wireless networks, machine learning, game theory, security, unmanned aerial vehicles, cyber-physical systems, and network science. He received the Dean's Award for Research Excellence from Virginia Tech in 2019. He is also a recipient of the NSF CAREER Award in 2013, the AFOSR Summer Faculty Fellowship in 2014, and the Young Investigator Award from the Office of Naval Research, in 2015. He is the recipient of the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, of the 2017 IEEE ComSoc Best Young Professional in Academia Award, of the 2018 IEEE ComSoc Radio Communications Committee Early Achievement Award, and of the 2019 IEEE ComSoc Communication Theory Technical Committee. He was the author/coauthor of Eight Conference Best Paper Awards at WiOpt in 2009, ICIMP in 2010, IEEE WCNC in 2012, IEEE PIMRC in 2015, IEEE SmartGridComm in 2015, EuCNC in 2017, IEEE GLOBECOM in 2018, and IFIP NTMS in 2019. He currently serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. He is an Editor-at-Large for the IEEE TRANSACTIONS ON COMMUNICATIONS. He is an IEEE Distinguished Lecturer.

Choong Seon Hong (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University, Japan, in 1997. In 1988, he joined KT, where he was involved in broadband networks as a Member of Technical Staff. Since 1993, he has been with Keio University. He was with the Telecommunications Network Laboratory, KT, as a Senior Member of Technical Staff and as the Director of the Networking Research Team until 1999. Since 1999, he has been a Professor with the Department of Computer Science and Engineering, Kyung Hee University. His research interests include future Internet, ad hoc networks, network management, and network security. He has served as the General Chair, the TPC Chair/Member, or an Organizing Committee Member of international conferences, such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, TINA, SAINT, and ICOIN. He was an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, and the IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He currently serves as an Associate Editor of the *International Journal of Network Management*, and an Associate Technical Editor of the *IEEE Communications Magazine*. He is a member of the ACM, IEICE, IPSJ, KIISE, KICS, KIPS, and OSIA.