This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

When Edge Computing Meets Microgrid: A Deep Reinforcement Learning Approach

Md. Shirajum Munir, Sarder Fakhrul Abedin, *Student Member, IEEE,* Nguyen H. Tran, *Senior Member, IEEE,* and Choong Seon Hong, *Senior Member, IEEE*

***Abstract*—The computational tasks at multi-access edge computing (MEC) are unpredictable in nature, which raises uneven energy demand for MEC networks. Thus, to handle this problem, microgrid has the potentiality to provides seamless energy supply from its energy sources (i.e., renewable, non-renewable, and storage). However, supplying energy from the microgrid faces challenges due to the high uncertainty and irregularity of the renewable energy generation over the time horizon. Therefore, in this paper, we study about the microgrid-enabled MEC networks' energy supply plan, where we first formulate an optimization problem and the objective is to minimize the energy consumption of microgrid-enabled MEC networks. The problem is a mixed integer nonlinear optimization with computational and latency constraints for tasks fulfillment, and also coupled with the dependencies of uncertainty for both energy consumption and generation. Therefore, we show that the problem is an NP-hard problem. As a result, second, we decompose our formulated problem into two subproblems: 1) energy-efficient tasks assignment problem for MEC into community discovery problem and 2) energy supply plan problem into Markov Decision Process (MDP). Third, we apply a low complexity density-based spatial clustering of applications with noise (DBSCAN) to solve the first subproblem for each BS distributedly. Sequentially, we use the output of the first subproblem as a input for solving the second subproblem, where we apply a model-based deep reinforcement learning (MDRL). Finally, the simulation results demonstrate the significant performance gain of the proposed model with a high accuracy energy supply plan.**

***Index Terms*—Multi-Access Edge Computing (MEC), Microgrid, Unsupervised Learning, Deep Reinforcement Learning, Demand Response, Energy Management, Computational Tasks, Internet of Things (IoT).**

I. INTRODUCTION

IN the era of fifth-generation (5G) networks with the sustainable expansion of smart services via Internet of Things (IoT) applications, multi-access edge computing (MEC) is a revolutionary technology. MEC is a key technology that brings application-oriented computational capabilities at the edge of carrier networks, where MEC hosts are deployed at the edge, central data network, or between them [1]. The potentiality of MEC includes low-latency, high-bandwidth, and real-time computation with the decision making feed-back for heterogeneous IoT applications and services (e.g., smart home, smart city, emergency service, virtual reality (VR), augmented reality (AR), autonomous vehicles, smart energy, industrial IoT, and so on) [2]. Moreover, toward the year of 2021, these applications will produce 49 exabytes of computational data per month, which includes 78% of video data, and also the MEC is the key enabler to compute the 63% of total computation [3]. However, these applications are very hungry for energy consumption, where high computational background services are deployed at MEC for fulfilling the application requirements [4]. In addition, to compute the huge amount of computational tasks at MEC, $30-50\%$ of the additional energy consumption is required [5]. In fact, the energy consumption of MEC is uneven since computational tasks request are random over time. However, the physical deployment of MEC can be flexible for the network operators while supported applications, available site facilities, operational, performance and/or security parameters are considered along with technical and business perspective [6].

The MEC is already included as an essential component in the smart infrastructures such as, the smart city, smart factory [2]. Meanwhile, the microgrid is also considered to be prominent in those MEC infrastructures [7]. As a result, microgrid can be useful energy supplement to MEC, while the necessity of the renewable energy powered base stations (BSs) operation is established more than a decade ago by the Ericsson (telecommunications company) [8]. Furthermore, joint operation of wireless networks and renewable energy supply was achieved energy saving, where a renewable energy aware grid-enabled cellular networks address energy harvesting challenges for energy load balancing [9]–[11], task scheduling [12], and demand-side management for intra-networks infrastructure, which save up to 18% of the energy usages [13]. In addition, suitable combination of the renewable, non-renewable and storage energy generation and distribution can save up to 30% of the energy usages for radio access networks [14]. However, MEC energy consumption has been overlooked in these activities. Therefore, a joint problem can be a possible way to tackle the uncertainty for both energy consumption of MEC networks and microgrid energy generation towards the energy saving.

MEC confronts with the sustainability issue to manage the computation with respect to energy consumption [15], [16] and the competence of MEC operation is depending on effi-

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2017R1A2A2A05000995). *Dr. CS Hong is the corresponding author.

Md. Shirajum Munir, Sarder Fakhrul Abedin, and Choong Seon Hong are with the Department of Computer Science and Engineering, Kyung Hee University, South Korea (e-mail: munir@khu.ac.kr; saab0015@khu.ac.kr; cshong@khu.ac.kr).

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

Corresponding author: Choong Seon Hong (e-mail: cshong@khu.ac.kr).

cient energy supply for microgrid empowered MEC networks. However, the MEC receives tasks with the uncertainty, where the characteristic of the energy consumption relies on the computational payload size. Furthermore, the reliability and stability of the microgrid energy supply contingent on energy generation of the renewable (e.g., solar, wind, biofuels, etc.) and non-renewable (e.g., diesel generator, coal power, and so on) energy sources [17]. In order to solve this, a strong coordination between MEC networks energy consumption and microgrid energy generation is required over the time horizon. In such case, we devise a microgrid-enabled MEC networks' energy supply plan problem, which not only guarantees computational and delay requirements but also ensures the sustainable energy supply to MEC networks. In order to achieve this we face several challenges:

- First, energy consumption of the MEC networks varies on the nondeterministic flow of computational tasks request. Furthermore, to execute these computation with a high degree of reliability, computational capacity of the MEC server and tolerable delay for the tasks are needed to be considered.
- Second, the coordination between MEC operation and microgrid energy supply is another critical challenge, where both of them face uncertainties over different time periods. As a result, the long-term estimation of the energy consumption and generation is more important than the immediate estimation.
- Third, a centralized solutions can no longer cope with the massive overhead in terms of required computation and signaling for dense MEC networks. Consecutively, a distributed solutions can be a possible way to overcome these overhead.

To address these challenges, we focus on approaches that not only coordinate both MEC network energy consumption and microgrid energy generation, but also adjust the task assignment to MEC such that energy consumption is minimized. We summarize our key contributions as follows:

- We first formulate the *multi-access edge server energy supply plan* problem, which is a mixed integer nonlinear optimization problem. The problem's objective is to minimize the energy consumption for the MEC servers such that the computational and latency requirements are satisfied. In addition, the formulated problem coordinates between the MEC networks energy consumption and the microgrid energy generation to determine the microgrid energy supply plan for fulfilling the energy demand of MEC networks.
- In order to solve the formulated problem with a data-driven and distributed approaches, we decompose this into two subproblems. The first subproblem of energy-efficient tasks assignment is formulated a community discovery problem and this problem is solved distributedly for each BS. Sequentially, the second subproblem pertaining to energy supply plan and that is decomposed by Markov Decision Process (MDP), where the input of second subproblem depends on the output (energy demand) of first subproblem and microgrid energy generation. Since, the decision of energy supply plan is strongly related on those time variant information, in which a long term estimation is performed using historical information along with the current observation. As a result, a strong connection between MEC energy demand and supply is established, which implies the relationship between the uncertainties of MEC task loads and microgrid energy generation.
- To derive the solution, we use a density-based spatial clustering of applications with noise (DBSCAN) based energy-efficient tasks assignment algorithm to solve the first subproblem distributedly. To an extent, this algorithm effectively manages nondeterministic and heterogeneous computational tasks for MEC operation while considering the computational and tolerable delay requirements. Consecutively, we obtain the solution for microgrid energy supply plan, where we use an algorithm based on model-based deep reinforcement learning (MDRL). The proposed solution achieves fast convergence with high forecasting accuracy, which mitigates the uncertainties for MEC operation along with energy consumption and microgrid energy generation.
- Finally, we perform an extensive simulation analysis for the proposed microgrid-enabled multi-access energy supply plan. The simulation results show that the proposed approach outperforms the greedy and the random-greedy approaches with the 97% of tasks execution success rate, saves up to 7.3% of the total energy consumption, and provides a high accuracy energy supply plan.

The remainder of the paper is organized as follows. A review of related works is given in Section II. Section III contains the system model and formulation of the multi-access edge server energy supply plan problem. Section IV presents the proposed problem decomposition. Section V represents a data-driven energy supply plan, which describes the solution procedure for the proposed model. Experimental results are discussed in Section VI. Finally, conclusions are given in Section VII.

II. Related Work

Energy management, for power grid-enabled wireless networks, have received much attention in recent years. Therefore, in the aspects of network deployment, and network operation for small cell networks, the feasibility analysis has been performed with renewable energy generation [9]. Nevertheless, this work did not study task computation or demand response (DR) management. The broadcasting nature of the wireless communications, wired energy transfer facility established by the grid architecture among multiple BSs has influenced the offloading decision between the BSs [10]. However, this research observed the energy loss for the offloaded traffic since it always requires more energy compared to the original associated nearest BS. A smart grid powered wireless Heterogeneous Network (HetNet) scenario was considered, where this study proposed a joint optimization problem and the objective is to minimize both transmit power among the networks and power loss for power distribution. This proposed model achieved an energy-efficient wireless data transmission with less power loss for a grid-supported wireless HetNet as compared to standalone HetNet [11]. These studies, focus on energy management for BSs operation equipped with renewable energy generation, while the major concern about the establishment of infrastructure of power grid supported wireless networks.

Similarly, a proposal to jointly optimizing BS operation and power distribution for mobile networks powered by a smart grid has made [13]. This study considered the coupling of BS operation and the power distribution challenges, and then provides an approximate solution. However, this work did not consider the MEC infrastructure. Thus, to operate HetNets with a hybrid energy supply, user scheduling, and resource allocation scheme has proposed based on the renewable energy sources for each of the small cells. The deep reinforcement learning has used to solve this problem by introducing the actor/critic algorithm [12]. However, they did not mention the relationship between the mobile networks task requests with renewable energy generation over the time horizon.

To an extent, researches are focusing on solving energy efficiency of the task offloading problems using non-linear problem, which in some cases we scarify the optimal solution to reduce the computational complexity of the problem or proposed optimal solution with high computational complexity. In respect to low latency communication decision making, high complexity solution is not appropriate [18], [19]. However, for some cases, grid-enabled wireless networks are not considered, so the aim of MEC faces with sustainability issues. To fulfill the objective of MEC, which accomplish the smooth transition for service fulfillment for video analytics, location services, Internet-of-Things (IoT), augmented reality (AR), optimized local content distribution, data caching, and so on, near to the end device with the high reliability.

From the perspective of energy demand forecasting and demand-side management, most studies have considered the residential and industrial energy management points of view. Furthermore, to solve the unpredictable energy demand for home appliances, three layers hierarchical architecture has used to solve the multi-objective optimization problem for demand-side management [20]. Furthermore, an on-line cloud-based Non-Intrusive Load Monitoring (NILM) methodology was proposed to detect household appliance energy load in a non-intrusive way. This method has separated residential building energy consumption in near real-time for the proper demand-side management of the households appliance [21]. Additionally, DR management has already proved the potentiality for effective energy supply of residential, commercial, and cloud data center [21]–[23].

In our problem, the energy-efficient tasks assignment problem solves using the DBSCAN-based method, which is unsupervised learning in nature [24], [25]. The characteristic of DBSCAN is immune to noise and also able to handle clusters of various shapes and sizes. The number of clusters determines by its nature, which is more appropriate to solve our problem.

On the other hand, various kind of reinforcement learning (RL) has used for solving the problems regarding robotics applications, user task offloading, network management, and also for the energy market strategic plan [26]–[29]. Multi-agent reinforcement learning has used to solve the spatiotemporal resource requirements of heterogeneous mobile devices under the fog environments [27]. The smart city services are managed by ϵ-greedy based RL method for the software-defined networks (SDN), where the model is exploration and exploitation in nature [28]. The single agent-based Q-learning algorithm has deployed for analyzing the energy market and provides the market strategy [29]. However, these methods did not use the deep learning RL for the policy determination and also takes a long time to convergence. In this research, we have used model-based deep reinforcement learning (MDRL) for energy supply plan, which can overcome those challenges.

TABLE I: Summary of Notation

Notation	Description
$\mathcal{B}$	Set of BSs under the microgrid
$\mathcal{C}_i$	Set of active multi-access edge servers under the BS $i \in \mathcal{B}$
j	Active edge server under the BS i, $j \in \mathcal{C}_i$
$\mathcal{K}$	Set of tasks under the BS $i \in \mathcal{B}$ with edge server $\forall j \in \mathcal{C}_i$
$u_i(t)$	Multi-access edge servers computational capacity
$\lambda_i(t)$	Task arrival rate at BS $i \in \mathcal{B}$
$S_i(t)$	Average traffic size at BS $i \in \mathcal{B}$
$r_i(t)$	Average data rate at BS $i \in \mathcal{B}$
$\mu_i(t)$	Service rate at BS $i \in \mathcal{B}$
$\rho_i(t)$	Server utilization at BS $i \in \mathcal{B}$
$\psi_i(t)$	Latency ratio at BS $i \in \mathcal{B}$
η^{net}	Energy coefficient for network operations
η^{cpu}	Energy coefficient for CPU operations
$\mathcal{E}_i^{net}(t)$	Network operations energy consumption at BS $i, \forall j \in \mathcal{C}_i$
$\mathcal{E}_i^{cpu}(t)$	CPU operations energy consumption at BS $i, \forall j \in \mathcal{C}_i$
$\phi_{st}^{net}(t)$	Network operations static energy at BS $i, \forall j \in \mathcal{C}_i$
$\phi_{st}^{cpu}(t)$	CPU operations static energy $i, \forall j \in \mathcal{C}_i$
$\mathcal{E}_i^{dyn}(t)$	Dynamic energy consumption for $i, \forall j \in \mathcal{C}_i$
$\mathcal{E}_i^{st}(t)$	Static energy consumption for $i, \forall j \in \mathcal{C}_i$
$\mathcal{E}^{tot}(t)$	Total energy consumption for $\forall i \in \mathcal{B}$
$g^{ren}(t)$	Microgrid renewable energy generation at time slot t
$g^{non}(t)$	Microgrid non-renewable energy generation at time slot t
$g^{sto}(t)$	Microgrid storage energy at time slot t
$g^{tot}(t)$	Microgrid total energy generation at time slot t

In summary, our proposed approaches consider the microgrid energy supply plan in the domain of microgrid-enabled MEC networks, which is an evolving area for research community. To the best of our knowledge, there are no such study for the analogy of MEC energy consumption with the renewable energy generation, which resolves the sustainable issue of the multi-access edge computing (MEC). Thus, a data-driven approach is capable to provide a holistic energy supply plan for microgrid-enabled MEC networks. The proposed method considers the MEC servers energy consumption model-based on the computational requirements and tolerable computational delay requirements for the heterogeneous computational tasks of the IoT services. The proposed approaches and solution are a new contributions with respect to this field.

III. System Model And Problem Formulation

A. System Model

We consider a microgrid-enabled wireless network that consists of a set $\mathcal{B} = \{1, 2, \dots, B\}$ of BSs and each BS i has a set of active multi-access edge servers $\mathcal{C}_i = \{1, 2, \dots, C_i\}$ with the homogeneous computational capacity, as seen in Fig. 1. We also consider one time slot t, which is in a infinite

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

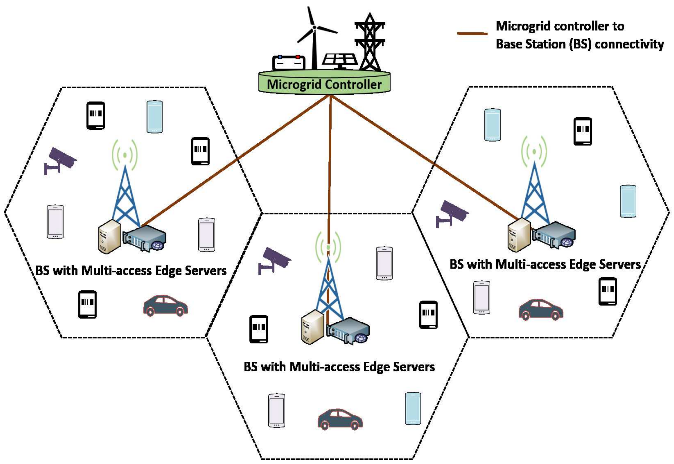


Fig. 1: System model.

time horizon $\mathcal{T} = 1, 2, \ldots, \infty$ and the duration of each time slot t is 15 minutes [30]. $u_i(t)$ represents the computational capacity of an active edge server at time slot t. There is a set of heterogeneous user tasks $\mathcal{K} = \{1, 2, \ldots, K\}$ with a user task association indicator $\omega_{ik}(t) = 1$ if task k is already assigned to the BS i at time t, 0 otherwise. Each task $k \in \mathcal{K}$ is served by a multi-access edge server $j \in \mathcal{C}_i$ at BS i. The properties of task $k \in \mathcal{K}$ represents with a tuple $\langle \beta_k, \gamma_k \rangle$, where β_k, and γ_k denote number of expected computational units (CPU cycles), and maximum tolerable computational delay, respectively. Additionally, in this model the microgrid controller is able to control the energy supply based on the energy demands from the networks. Table I represents the summary of notation.

1) Communication and Computation Model: We consider the task arrival rate at time slot t under the BS $i \in \mathcal{B}$ is $\lambda_i(t)$, which follows the Poisson process with average traffic size $S_i(t)$ and the average traffic load denoted by $\lambda_i(t)S_i(t)$. The transmission data rate for an BS $i \in \mathcal{B}$ is as follows:

$$r_i(t) = w_i \log_2 \left(1 + \frac{p_i G_i(t)}{\sigma^2 + \sum_{i' \in \mathcal{B}, i' \neq i} I_{i'}(t)}\right), \quad (1)$$

where w_i is the channel bandwidth, the transmission power is denoted with p_i, $G_i(t)$ is the channel gain, the channel noise is σ^2, and the interference of the channel is $I_{i'}(t)$.

Thus, the user task execution service rate under the BS i at time slot t is as follows:

$$\mu_i(t) = \frac{r_i(t)}{\lambda_i(t) S_i(t)}. \quad (2)$$

Hence, the service time under the BS i at time slot t is,

$$T_i(t) = \frac{1}{\mu_i(t)}. \quad (3)$$

Therefore, we assume that a set of tasks $\forall k \in \mathcal{K}$ uniformly distributed at time slot duration t under the BS i and the overall utilization rate is as follows [31]:

$$\rho_i(t) = \int_{t \in \mathcal{T}} \omega_{ik}(t) \frac{\lambda_i(t)}{\mu_i(t)} dt, \quad (4)$$

where $\int_{t \in \mathcal{T}} \omega_{ik}(t)\lambda_i(t)dt$ is the total amount of tasks served by the BS i at time slot t.

The total time that includes the execution (service) time and the waiting time in the queue for BS i is as follows [32]:

$$\tau_i(t) = \frac{1}{\mu_i(t)(1 - \rho_i(t))}. \quad (5)$$

The latency ratio at BS $i \in \mathcal{B}$ is given below,

$$\psi_i(t) = \frac{\tau_i(t) - T_i(t)}{T_i(t)} = \frac{\rho_i(t)}{1 - \rho_i(t)}. \quad (6)$$

A smaller value of $\psi_i(t)$ determines that BS i has less latency communication for its associated tasks.

Since, in this scenario[1], user task requests behave according to a Poisson process, which acts as stochastic (random over time) in nature. On the other hand, the service rate is exponentially distributed as the traffic size is already known, where the data rate is considered as a constant. Therefore, M/M/1 queuing can be considered to be an appropriate choice [31], [33], [34].

The amount of CPU resources required for a single task k execution at MEC is

$$\kappa_i(t) = \sum_{n \in N} \left(\sum_{m \in M} \delta_{mn}(t) \varpi \right), \quad (7)$$

where N is the number of cores, M is the number of CPU components (CPU cycles), ϖ represents the weight for a single core, and $\delta_{mn}(t)$ is the active ratio [35].

2) Energy Consumption Model:

Base Station Operation Energy Consumption: In BS i, the energy consumption for network communication and data transfer through the networks with transmission power p_i and transmission rate $r_i(t)$ is as follows:

$$\mathcal{E}_i^{net}(t) = \eta^{net}(t) \frac{p_i S_i(t)}{r_i(t)} + \phi_{st}^{net}(t), \quad (8)$$

where η^{net} is the energy coefficient for transferring data with energy consumption $\frac{p_i S_i(t)}{r_i(t)}$ through the networks, the value of η^{net} is depending on the type of network devices [36]. $\phi_{st}^{net}(t)$ determines the static energy consumption for the BS operations [37].

CPU of MEC server Energy Consumption: CPU of MEC server energy consumption depends on the processor architecture, number of CPU cores and number of CPU cycle usages. The total energy consumption is defined as follows [35]:

$$\mathcal{E}_i^{cpu}(t) = \sum_{n=1}^{N} \left(\sum_{m=1}^{M} \eta^{cpu} \delta_{mn}(t) \varpi \right) + \phi_{st}^{cpu}(t), \quad (9)$$

where $\delta_{mn}(t)\varpi$ represents the dynamic power consumption for core n with component m of the CPU, and η^{cpu} is the energy coefficient for CPU usage. The static energy consumption is represented by $\phi_{st}^{cpu}(t)$.

[1]Additionally, the proposed system model is capable of adopting a more general queuing model such as, M/G/1. Unlike the M/M/1 model, the service time in M/G/1 follows an arbitrary distribution that is the general [13], [38]. Therefore, in such case, the total time $\tau_i(t)$ can be defined by Pollaczek-Khinchin formula [39], $\tau_i(t) = \frac{\lambda_i(t)\mathbb{E}[T_i(t)^2]}{2(1-\rho_i(t))} + \mathbb{E}[T_i(t)]$, where $\mathbb{E}[T_i(t)]$ is expectation of service time and $\mathbb{E}[T_i(t)^2]$ is expectation of square of the service time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

Total Energy Consumption: The multi-access edge server-enabled network system requires two types of energy consumption: the static energy $\mathcal{E}_i^{st}(t)$ and the dynamic energy $\mathcal{E}_i^{dyn}(t)$ consumption [37]. The static energy consumption has a fixed amount of energy that does not depend on the computational tasks, and this type of energy is needed for idle state operations. The dynamic energy depends on the computational task load that includes BS operation (network data transfer) and amount of CPU usages. Equations (8) and (9) represent the amount of energy consumption for data transfer through the networks and CPU usages, respectively [40]. The dynamic energy consumption at time slot t is given below as

$$\mathcal{E}_i^{dyn}(t) = \mathcal{E}_i^{cpu}(t) + \mathcal{E}_i^{net}(t). \tag{10}$$

The total energy consumption under BS $i \in \mathcal{B}$ is as follows [37]:

$$\mathcal{E}_i^{tot}(t) = \mathcal{E}_i^{dyn}(t)\eta^{dyn}(t) + \mathcal{E}_i^{st}(t), \tag{11}$$

where, $\eta^{dyn}(t)$ is energy coefficient for dynamic energy consumption at time slot t and the value of parameters $\eta^{dyn}(t)$ are known, which are also depended on types of base station (BS) [37]. Therefore, the total energy consumption under the microgrid for N base stations is defined by,

$$\mathcal{E}^{tot}(t) = \sum_{\forall i \in \mathcal{B}} (\mathcal{E}_i^{dyn}(t)\eta^{dyn}(t) + \mathcal{E}_i^{st}(t)). \tag{12}$$

3) Microgrid Energy Generation: In the microgrid, there are two type of energy sources. The first type is the renewable energy sources (e.g., solar, wind, biofuels, etc.), where the amount of renewable energy generation at time slot t is denoted by $g^{ren}(t)$. The second type of energy is non-renewable energy sources (e.g., diesel generator, coal power, and so on) where the amount of non-renewable energy at time slot t is defined as $g^{non}(t)$. Additionally, the non-renewable energy sources are connected with the main grid, the microgrid can buy additional energy from the main grid if the microgrid unable to fulfill the energy demand using its own energy sources [30]. The total energy generation $g^{tot}(t)$ at time slot t is as follows:

$$g^{tot}(t) = g^{ren}(t) + g^{non}(t). \tag{13}$$

Therefore, the additional amount of buying energy $g^{buy}(t)$ from the main grid is defined by,

$$g^{buy}(t) = \mathcal{E}^{tot}(t) - g^{tot}(t). \tag{14}$$

Additionally, microgrid is capable of storing the surplus amount of energy $g^{sto}(t)$ in storage medium for future usages and the amount of stored energy at time slot t is as follows [38]:

$$g^{sto}(t) = g^{tot}(t) - \mathcal{E}^{tot}(t). \tag{15}$$

B. Problem Formulation

The energy consumption ξ_{ijk} for a single user task $k \in \mathcal{K}$ at server $j \in \mathcal{C}_i$ is as follows:

$$\xi_{ijk} = \mathcal{E}_{ijk}^{dyn}(t)\eta^{dyn}(t) + \mathcal{E}_{ijk}^{st}(t). \tag{16}$$

The first part of equation (16) determines for the dynamic energy consumption to fulfill task $k \in \mathcal{K}$ at edge server $j \in \mathcal{C}_i$ from equation (10). The dynamic energy consumption depends on equations (8) and (9) regarding the BS operation and CPU usage, respectively, for the computational task. The second part of equation (16) represents the static energy consumption which represents idle state operation energy consumption and does not depend on the computational task $k \in \mathcal{K}$. Furthermore, the amount of energy required for the server $j \in \mathcal{C}_i$ to complete one task is $\sum_{k \in \mathcal{K}} \xi_{ijk} x_{ijk}$, where x_{ijk} is a binary decision variable with $x_{ijk} = 1$ if the task $k \in \mathcal{K}$ is assigned to server $j \in \mathcal{C}_i$, and 0 otherwise:

$$x_{ijk} = \begin{cases} 1, & \text{if } k \in \mathcal{K} \text{ assigned to } j \in \mathcal{C}_i \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The objective is to minimize the total amount of energy consumption ξ_{ijk} needed for executing the heterogeneous tasks under the constraints of computational capacity and maximum tolerable delay. Additionally, based on the energy demand, the microgrid takes a necessary decision about the energy supply plan using a binary decision variable $\zeta_t \in \{0, 1\}$,

$$\zeta_t = \begin{cases} 1, & \text{if } g^{sto}(t) \geq 0,\ t \in \mathcal{T} \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

where $\zeta_t = 1$ if microgrid is able to fulfill the energy demand from its own energy generation sources at time slot t, and 0 otherwise.

The problem formulation is as follows:

$$\min_{x,\zeta} \sum_{\forall t \in \mathcal{T}} \sum_{\forall i \in \mathcal{B}} \sum_{\forall j \in \mathcal{C}_i} \sum_{\forall k \in \mathcal{K}} \omega_{ik}(t) x_{ijk} \zeta_t \xi_{ijk}, \tag{19}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{C}_i} x_{ijk} = 1, \forall k \in \mathcal{K}, \tag{19a}$$

$$\sum_{k \in \mathcal{K}} x_{ijk} \leq K, \forall j \in \mathcal{C}_i, \tag{19b}$$

$$\zeta_t g^{sto}(t) + (1 - \zeta_t) g^{buy}(t) \geq 0, t \in \mathcal{T}, \tag{19c}$$

$$0 \leq g^{tot}(t) \leq g^{ren}(t) + g^{non}(t) + g^{sto}(t), t \in \mathcal{T}, \tag{19d}$$

$$\mathcal{H}_t(.) \in \{H_t(.), \ldots, H_T(.)\}, \forall t \in \mathcal{T}, \tag{19e}$$

$$0 \leq \beta_k \leq \kappa_i(t), k \in \mathcal{K}, j \in \mathcal{C}_i, \tag{19f}$$

$$0 \leq \gamma_k \leq \tau_i(t), k \in \mathcal{K}, j \in \mathcal{C}_i, \tag{19g}$$

$$\zeta_t \in \{0, 1\}, t \in \mathcal{T}, \tag{19h}$$

$$x_{ijk} \in \{0, 1\}, \forall j \in \mathcal{C}_i, k \in \mathcal{K}. \tag{19i}$$

In problem (19), constraint (19a) ensures that one task will be assigned to the only one multi-access edge server and constraint (19b) determines the maximum number of tasks is not greater than the total number of associated tasks K with the MEC servers $\mathcal{C}_i$ in BS i. Constraint (19c) represents the coupling between energy demand of MEC networks and energy supply plan for the microgrid side. Here, the variable ζ_t takes decision regarding the amount of energy store $g^{sto}(t)$ or buy $g^{buy}(t)$ at time t. Moreover, this decision depends on the amount of total energy consumption $\mathcal{E}^{tot}(t)$ of the MEC networks at time t, which completely relies on variable x_{ijk} of constraint (19a). Therefore, (19d) is a constraint that ensures the total energy generation is not bigger than the sum of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

renewable, non-renewable, and storage energies at time slot t. Hence, the constraint (19e) is the coupling between the time horizon and the historical data. The set of functions $\mathcal{H}_t(.)$ preserves all the energy consumption and generation information for $\forall t \in T$. Constraint (19f) fulfills the computational requirement of each task with the computational capacity of MEC server (7). Therefore, constraint (19g) ensures maximum tolerable latency of a task must be satisfied by the total (service and waiting) delay (5) of the MEC networks. Finally, the (19h) and (19i) defines the decision variables as a binary variable.

The optimization problem (19) is a Mixed Integer Non-Linear Programming (MINLP) problem, where the combinatorial properties make this problem NP-hard. Hence, it is extremely hard to solve but not impossible, to find the global optimal result. Therefore, to obtain the low complexity solution, we decompose the formulated problem into two subproblems, energy-efficient tasks assignment for the MEC servers under the BS, and energy supply plan for the microgrid controller.

IV. Problem Decomposition

We decompose our original problem (19) into two subproblems. Firstly, we reformulate the *energy-efficient tasks assignment* problem for each BS in a constraint programming model, which provides a community discovery problem [41] and this is similar to the label propagation method [42]. In our reformulation, the constraints are able to generate the interesting variants, which are very convenient to solve data-driven based problem, more likely our first subproblem. Secondly, we model the *microgrid energy supply plan* using the model-based deep reinforcement learning (MDRL), while the second subproblem is stochastic in nature. Furthermore, the MDRL provides guarantee to faster convergence to the optimal solution for using the prior knowledge of transitions from the model [26].

A. Energy-efficient Tasks Assignment

We have defined our first subproblem as a constraint programming problem for solving in a data-driven approach. We recall the set of multi-access edge servers $\mathcal{C}_i$ at BS $i \in \mathcal{B}$, where these servers execute a set of computational tasks $\mathcal{K}$ and the tasks are already associated with BS $i \in \mathcal{B}$ using the task association indicator $\omega_{ik}(t) = 1, \forall k \in \mathcal{K}$. Moreover, the task $k \in \mathcal{K}$ is executed by server $j \in \mathcal{C}_i$ and this single task is performed by only one server. Hence, from the task execution point of view, the set of multi-access edge server $\mathcal{C}_i$ creates a partition of the set of tasks $\mathcal{K}$ incurred by the optimization objective, and mathematically we can represent this as $\mathcal{C}_{ij} \subseteq \mathcal{K}, \forall C_{ij} \in \mathcal{C}_i$. In addition, we have two domains: the first domain is the set of tasks $\mathcal{K}$ and the second is the set of multi-access edge serves $\mathcal{C}_i$.

We have considered a two-dimensional space with computational requirements β_k and the maximum tolerable delay γ_k for every task $k \in \mathcal{K}$. The decision variable $x_{kj} = 0$ denotes that server j is assigned for task k, and is 0 otherwise. Therefore, we have rewritten the decision variable equation (17) as follows:

$$x_{kj} = \begin{cases} 1, & \text{if server } j \in \mathcal{C}_i \text{ assigned for task } k \in \mathcal{K} \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

Additionally, we have added two more binary decision variables for fulfilling the computational capacity and tolerable latency. $y_{kj} = 1$ if server $j \in \mathcal{C}_i$ satisfy the computational capacity to execute task $k \in \mathcal{K}$, and 0 otherwise

$$y_{kj} = \begin{cases} 1, & 0 \leq \beta_k \leq \kappa_i(t), k \in \mathcal{K} \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Similarly, $z_{kj} = 1$ if server $j \in \mathcal{C}_i$ has capacity to execute task $k \in \mathcal{K}$ in tolerable delay, and 0 otherwise

$$z_{kj} = \begin{cases} 1, & 0 \leq \gamma_k \leq \tau_i(t), k \in \mathcal{K} \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

Moreover, the variable Υ determines length between the inter-class distance where this variable determines well separated and dense task cluster. The following is the constraint programming model for multi-access edge server energy consumption:

$$\max_{\Upsilon} \ \Big(\sum_{j \in \mathcal{C}_i} \min_{x} \big(\sum_{k \in \mathcal{K}} \xi_{kj} x_{kj}\big)\Big), \tag{23}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{C}_i} x_{kj} = 1, \forall k \in \mathcal{K}, \tag{23a}$$

$$\sum_{k \in \mathcal{K}} x_{kj} \leq K, \forall j \in \mathcal{C}_i, \tag{23b}$$

$$0 \leq \beta_k \leq \kappa_i(t), k \in \mathcal{K}, \tag{23c}$$

$$0 \leq \gamma_k \leq \tau_i(t), k \in \mathcal{K}, \tag{23d}$$

$$0 \leq y_{kj} + z_{kj} \leq 1, j \in \mathcal{C}_i, \tag{23e}$$

$$\sum_{\forall k \in \mathcal{K} j} \beta_{jk} \geq u_i(t), j+1 \in \mathcal{C}_i, \tag{23f}$$

$$x_{kj} \in \{0,1\}, k \in \mathcal{K}, j \in \mathcal{C}_i, \tag{23g}$$

$$y_{kj} \in \{0,1\}, k \in \mathcal{K}, j \in \mathcal{C}_i, \tag{23h}$$

$$z_{kj} \in \{0,1\}, k \in \mathcal{K}, j \in \mathcal{C}_i. \tag{23i}$$

The objective of the problem (23) is to serve all tasks $\forall k \in \mathcal{K}$ with energy-efficient while maximizing number of server usages under the BS $i \in \mathcal{B}$. Constraints (23a), (23b), (23c), (23d), and (23g) are similar to constraints (19a), (19b), (19f), (19g), and (19i) in problem (19). The constraint (23e) is used to handle the unpredictable task assignment at multi-access edge sever $C_i \in \mathcal{C}_i$. If any server $j \in \mathcal{C}_i$ exceeds the maximum computational capacity or the tolerable delay, then the task is assigned to another server $j+1 \in \mathcal{C}_i$, as determined by constraint (23f), which ensures server load balancing. Finally, the constraints (23h) and (23i) restrict variables y and z as binary variables.

B. Microgrid Energy Supply Plan

In order to solve the *microgrid energy supply plan* problem, we use a model-based deep reinforcement learning (MDRL) for the second subproblem of the proposed problem (19). The goal of this subproblem is to provide high accuracy energy supply plan of the MEC networks by dealing with uncertainties of both energy consumption and generation, where the energy consumption is the aggregated output from the first subproblem (23). On the other hand, the energy generation depends on various types of the energy source of microgrid such as, renewable (e.g., solar, wind, biofuels, etc.), non-renewable (e.g., diesel generator, coal power, and so on), and stored energy.

In the model-based deep reinforcement learning (MDRL) settings, we consider a set of states $\mathcal{S} = \{1, 2, \dots, S\}$, where a state $s_t \in \mathcal{S}$ consists of a four-elements tuple $\langle \epsilon_t^{dem}, \epsilon_t^{ren}, \epsilon_t^{non}, \epsilon_t^{sto} \rangle$ for a single state space at time slot t. Therefore, ϵ_t^{dem}, ϵ_t^{ren}, ϵ_t^{non}, and ϵ_t^{sto} determine the amount of energy demand, renewable energy generation, non-renewable energy generation, and store energy of the microgrid, respectively. We also consider a set of actions $\mathcal{A} = \{1, 2, \dots, A\}$, where a action $a_t \in \mathcal{A}$ consists of two-actions tuple $\langle \zeta_t^1, \zeta_t^0 \rangle$, in which ζ_t^1 represents energy store decision and action ζ_t^0 presents buying decision for time slot t. A set of observations $\mathcal{O} = \{1, 2, \dots, O\}$, where $o_t \in \mathcal{O}$ represents a single observation for time slot's t and this consists of a three-elements tuple $\langle s_t, a_t, s_{t'} \rangle$ with current state s_t, current action a_t, and next state $s_{t'}$.

For a given action $a_t \in \mathcal{A}$, we consider a parameter θ, which determines a stochastic policy π_θ with a state transition probability $P_\theta(s_{t'}|s_t, a_t)$. In MDRL model, for a given state-action pair (s_t, a_t), an action a_t selection decision for the next state $s_{t'}$ is determined by a reward function $R(s_t, a_t)$, where the probability distribution of that reward function depends on the state transition probability distribution $P_\theta(o_t) = P_\theta(s_1) \prod_{i \in \mathcal{T}} P_\theta(a_t|s_t) P_\theta(s_{t'}|a_t, s_t)$ from the observation o_t at time slot t. Therefore, for both state transition and reward function work in a sequential manner. Hence, the objective is to choose a probabilistic policy π_θ over the actions from the observation o_t in order to maximize the value function $V^{\pi_\theta}(s_t)$ [26],

$$V^{\pi_\theta}(s_t) = \max_{\theta} \mathbb{E}_{o_t \sim P_\theta(o_t)} [\sum_{l=0}^{\infty} \alpha^l R_{t+l}(s_t, a_t) | s_t], \quad (24)$$

where the value function $V^{\pi_\theta}(s_t)$ determines the expected cumulative discounted rewards with a discount factor α.

Therefore, we redefine the value function (24) as a sum of the immediate reward $R(s_t, a_t)$ with current policy $\pi_\theta(a_t|s_t)$, where the state transition probability $P_\theta(s_{t'}|s_t, a_t)$ determines from policy $\pi_\theta(a_t|s_t)$,

$$\begin{aligned} V^{\pi_\theta}(s_t) = \sum_{a_t \in \mathcal{A}} \pi_\theta(a_t|s_t) [R(s_t, a_t) + \\ \sum_{s_{t'} \in \mathcal{S}} \alpha P_\theta(s_{t'}|s_t, a_t) V^{\pi_\theta}(s_{t'})]. \end{aligned} \quad (25)$$

Hence, the value function with optimal policy $\pi_\theta^*(a_t|s_t)$ is as follows:

$$V^{\pi_\theta^*}(s_t) = \max_{a \in \mathcal{A}} \mathbb{E}[\sum_{l=0}^{\infty} \alpha^l R_{t+l}(s_t, a_t) | s_t, a_t]. \quad (26)$$

Now, consider a new state space $s_{t'}$ with optimal value $V^{\pi_\theta^*}(s_{t'})$ and the optimal value function at time t is determined by,

$$\begin{aligned} V^{\pi_\theta^*}(s_t) = \max_{a_t \in \mathcal{A}} \mathbb{E}[R(s_t, a_t) + \\ \sum_{s_{t'} \in \mathcal{S}} \alpha P_\theta(s_{t'}|s_t, a_t) V^{\pi_\theta^*}(s_{t'})]. \end{aligned} \quad (27)$$

However, a state-action value function $Q^{\pi_\theta}(s_t, a_t)$ can help to choose the action a_t using the policy $\pi_\theta(a_t|s_t)$. The state-action value function is defined by,

$$Q^{\pi_\theta}(s_t, a_t) = \mathbb{E}_{\pi_\theta} [\sum_{l=0}^{\infty} \alpha^l R_{t+l}(s_t, a_t) | s_t, a_t]. \quad (28)$$

Additionally, equation (28) satisfies the coupling constraint (19e) of the problem (19) between the time slots, where the decision made based on the history of previous state information. Moreover, the action selection decision fully depends on the state transition probability $P_\theta(s_{t'}|s_t, a_t)$ of the state space, which reflects on the dynamics of Markovian. Therefore the state-action function is redefined by,

$$Q^{\pi_\theta}(s_t, a_t) = R(s_t, a_t) + \sum_{s_{t'}} \alpha P_\theta(s_{t'}|s_t, a_t) V^{\pi_\theta}(s_{t'}). \quad (29)$$

Equation (29) defines the maximum value of the state-action value function $Q^{\pi_\theta}(s_t, a_t)$ for any observation (state) that chooses a specific action. Equation (27) provides the optimal policy for the state-action value function and the state-action value function $Q^{\pi_\theta}(s_t, a_t)$ is used for the reconstruction of the optimal policy $\pi_\theta^*(a_t|s_t)$.

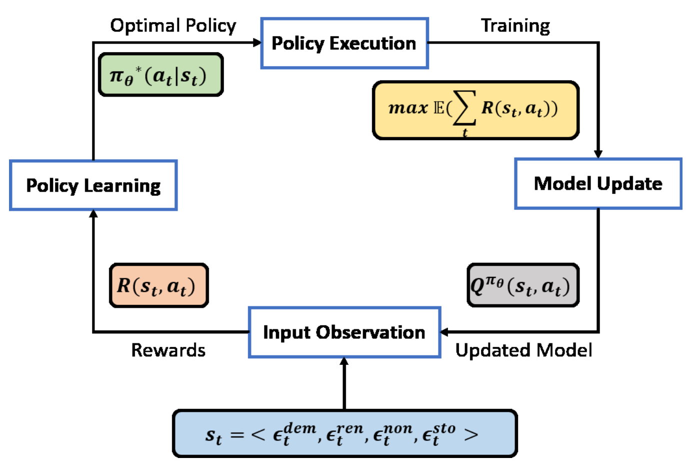


Fig. 2: Model-based deep reinforcement learning.

Fig. 2, depicts the working procedure of model-based reinforcement learning, where the model calculates rewards $R(s_t, a_t)$ from the observation (state information) $\langle \epsilon_t^{dem}, \epsilon_t^{ren}, \epsilon_t^{non}, \epsilon_t^{sto} \rangle$ and determines the optimal policy $\pi_\theta^*(a_t|s_t)$ using the current policy $\pi_\theta(a_t|s_t)$. A supervised learning technique is used, where backpropagation method updates the model for deep Q-networks (DQN) [43]. The loss function of model-based deep reinforcement learning (MDRL) is defined by,

$$L(t) = \min_{\theta} \frac{1}{|\mathcal{O}|} \sum_{o_t \in \mathcal{O}} \frac{1}{2} ||Q^{\pi_\theta}(s_{o_t}, a_{o_t}) - s_{o_{t'}}||^2, \quad (30)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

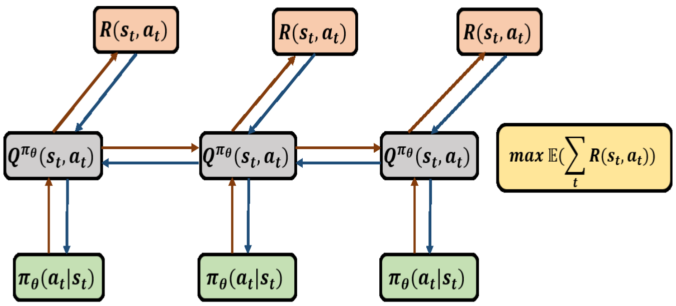


Fig. 3: Model-based deep reinforcement learning backpropagation.

where, $o_t \in \mathcal{O}$ and $o_{t'} \in \mathcal{O}$ represent current and next observation, respectively. The backpropagation for loss function (30) shows in Fig. 3, where the objective is to maximize the reward $R(s_t, a_t)$,

$$\max_{a_t} \sum_{t \in \mathcal{T}} \mathbb{E}_{\pi_\theta}(R(s_t, a_t)). \tag{31}$$

In the proposed MDRL model, a Softmax function determines the probability distribution for choose an action a_t with a maximum reward $R(s_t, a_t)$ and the probability $P(a_t)$ for action a_t is defined by, [44],

$$P(a_t) = \frac{\mathbb{E}(R(s_t, a_t)/\Gamma)}{\sum_{i=1}^{n} \mathbb{E}(R(s_t, a_t)/\Gamma)}, \tag{32}$$

where $R(s_t, a_t)$ is the reward for an action a_t and a temperature parameter is Γ. If the value of $\Gamma \to \infty$, then $P(a_t)$ is almost the same. Thus, a lower value of Γ provides the highest expected reward, where the action probability $P(a_t)$ tends to 1.

In this model, we have two actions ζ_t^1 and ζ_t^0, where action ζ_t^1 determines the energy storing decision and energy buying decision is determined by, ζ_t^0. Here, a rectified linear unit (ReLU) activation function is capable to activate a fully-connected hidden neural networks of the DQN (deep Q-networks) for this two actions [45]. The ReLU activation function is defined by,

$$f(a_t) = \max_{a_t}(a_t, 0). \tag{33}$$

A sigmoid activation function returns monotonically increasing values in the range of -1 to 1 [46]. The sigmoid activation function for output layer is as follows:

$$g(a_t) = \frac{e^{a_t}}{e^{a_t} + 1}. \tag{34}$$

In order to train the proposed MDRL model, we use Adam (short for Adaptive Moment Estimation) stochastic gradient descent optimizer for approximations [47]. This approximation algorithm estimates the first and second moments of the gradient and also computes individual adaptive learning rates for a different batches of data with different parameters. The first and second moments of the loss function (30) are defined by,

$$\Theta_{t+1}^{w} = \vartheta_1 \Theta_t^{w} + (1 - \vartheta_1) \nabla^{w} L(t), \tag{35}$$

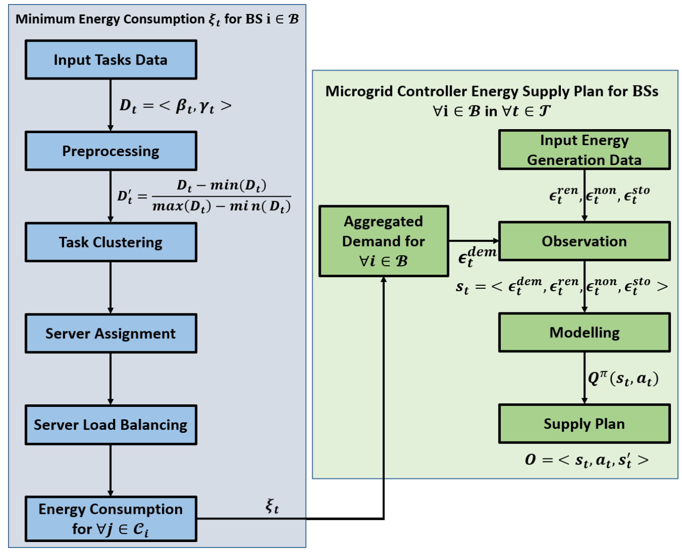


Fig. 4: Data-driven energy supply plan procedure.

$$\nu_{t+1}^{w} = \vartheta_2 \nu_t^{w} + (1 - \vartheta_2)(\nabla^{w} L(t))^2, \tag{36}$$

where ϑ_1 and ϑ_2 are the decay rates. Since, this approximation algorithm usages the adaptive learning rates, where the step size is important for the first few iterations of the training process. Therefore, this algorithm performs a bias correction before the weight update estimation and the bias correction functions for the first and second moments are,

$$\hat{\Theta^{w}} = \frac{\Theta_{t+1}^{w}}{1 - (\vartheta_1)_{t+1}}, \tag{37}$$

$$\hat{\nu^{w}} = \frac{\nu_{t+1}^{w}}{1 - (\vartheta_2)_{t+1}}. \tag{38}$$

Hence, the function of weight updates Δw_t with corrected bias is defined by,

$$\Delta w_t = -r \frac{\hat{\Theta^{w}}}{\sqrt{\hat{\nu^{w}}} + \varepsilon}, \tag{39}$$

where r is a learning rate and ε is a very small value, which protects from division by zero. Therefore, the updated weight for next time slot $t+1$ is as follows:

$$w_{t+1} = w_t + \Delta w_t. \tag{40}$$

V. Data-driven Energy Supply Plan Solution

We solve the *multi-access edge server energy supply plan* problem (19) in a distributed approach. The data-driven energy supply plan procedure is presented in Fig. 4. In the first part, the objective is to find the energy-efficient heterogeneous task execution for all BSs, and each BS solves this problem independently. The second part provides the energy supply plan, and the microgrid controller is responsible for solving this challenge.

- First, we solve the energy consumption problem (23) for each BS $i \in \mathcal{B}$ individually. To solve this problem we use unsupervised learning to determine the energy-efficiency while satisfying the computational and the maximum tolerable latency requirements for each task. We have chosen

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

Algorithm 1 Energy-efficient Tasks Assignment for BS $i \in \mathcal{B}$ Based on DBSCAN

Input: $D_t = \langle \beta_k, \gamma_k \rangle, \forall k \in \mathcal{K}, t \in \mathcal{T}$
Output: ξ_t, $t \in \mathcal{T}$
Initialization: Υ
1: **for** $\forall k \in \mathcal{K}, i \in \mathcal{B}$ **do**
2: **if** (k is visited) **then**
3: Continue to $k + 1 \in \mathcal{K}$
4: **else**
5: **for** $\forall j \in \mathcal{C}_i, i \in \mathcal{C}$ **do**
6: Constraints: (23c), (23h)
7: **if** $(0 \leq \beta_k \leq \kappa_i(t), k \in \mathcal{K}$ && $\Upsilon < \beta_k)$ **then**
8: $y_{kj} = 1, k \in \mathcal{K}, j \in \mathcal{C}_i$
9: **else**
10: $y_{kj} = 0, k \in \mathcal{K}, j \in \mathcal{C}_i$
11: **end if**
12: Constraints: (23d), (23i)
13: **if** $(0 \leq \gamma_k \leq \tau_i(t), k \in \mathcal{K}$ && $\Upsilon < \gamma_k)$ **then**
14: $z_{kj} = 1, k \in \mathcal{K}, j \in \mathcal{C}_i$
15: **else**
16: $z_{kj} = 0, k \in \mathcal{K}, j \in \mathcal{C}_i$
17: **end if**
18: Constraints: (23a), (23b), (23e), (23f), (23g)
19: **if** $(\sum_{k \in \mathcal{K}} x_{kj} \leq K, \forall j \in \mathcal{C}_i)$ **then**
20: $x_{kj} = 1, C_i \in \mathcal{C}_i$
21: **else if** $(0 \leq y_{kj} + z_{kj} \leq 1)$ **then**
22: $x_{kC_i} = 1, C_i \in \mathcal{C}_i$
23: **else if** $(\sum_{\forall k \in \mathcal{K}j} \beta_{jk} \geq u_i(t))$ **then**
24: $x_{k(j+1)} = 1, j + 1 \in \mathcal{C}_i$
25: **else**
26: $x_{kj} = 0, C_i \in \mathcal{C}_i$
27: **end if**
28: **end for**
29: **end if**
30: Calculate : $\xi_{kj}, \forall k \in \mathcal{K}, \forall j \in \mathcal{C}_i$, ξ_{kj} using eq. (16)
31: **end for**
32: Update the clustering model using ξ_t
33: **return** ξ_t

the density-based spatial clustering of applications with noise (DBSCAN) technique for the unsupervised learning [25]. The reason for this is that heterogeneous tasks have different computational requirements and data features (maximum tolerable delay) that are nondeterministic and further the tasks requested are not predictable. Moreover, to handle the most unpredictable tasks, we can use the outlier properties. Additionally, the worst case complexity of DBSCAN is $O(n^2)$ and the average case is $O(n \log n)$, which are more convenient considering the computational capacities of the MEC servers. Additionally, we have combined this with a control flow algorithm for the server task load balancing to achieve the energy-efficiency.

- Second, a model-based deep reinforcement learning (MDRL) technique is used to solve the energy supply plan for the microgrid. The reasons behind using MDRL are as follows. First, the model learns the dynamic behavior of the environment in a supervised learning manner from the historical data using deep neural networks. Second, this model finds the optimal policy by observing the current environment and the optimal policy used before updating the model. Finally, MDRL provides faster convergence to the optimal solution using a small number of interactions with the real-time observation from the environment [26].

From the implementation point of view, the *energy-efficient tasks assignment* method for BS $i \in \mathcal{B}$ is deployed under BS i. This method runs every 15 minutes (time slot t) to serve the associated tasks with BS $i \in \mathcal{B}$. Also, after each time slot t, the DBSCAN model will be updated and it sends the amount of energy consumption ξ_i to the microgrid controller. A similar strategy is performed by the BSs $\forall i \in \mathcal{B}$ under the microgrid coverage area. The microgrid controller is responsible for execution of the MDRL method.

A. Energy-efficient Tasks Assignment for BS Based on DBSCAN

To solve the problem (23), we have used the unsupervised learning (DBSCAN) method for task categorization and server assignment based on the computational demand and maximum tolerable delay features. DBSCAN-based method provides the global optimal result in respect to task categorization [48]. The flow control algorithm concept has used for server load balancing so as to minimize the energy consumption of the multi-access edge servers to fulfill the task requests.

The proposed DBSCAN-based *energy-efficient tasks assignment* algorithm 1 verifies the computational capacity β_k and the maximum tolerable delay γ_k for task $k \in \mathcal{K}$ in lines 7 through 17, and calculates the similarity of the tasks using variable Υ. Lines 7 to 11 label the decision variable y_{kj} to ensure the computational capacity satisfies constraints (23c) and (23h). Lines 12 to 17 are responsible for the delay tolerance using the decision variable z_{kj}, which fulfills constraints (23d) and (23i). The server assignment with the load balancing decision is made via lines 18 to 27, which reflects constraints (23a), (23b), (23e), (23f), and (23g) of the problem (23). Algorithm 1 aggregates the results of energy consumption during the 15 minutes of time slot t and sends the result ξ_t to the microgrid controller before starting the time slot $t + 1$.

B. Microgrid Energy Supply Plan Based on MDRL

Our proposed *MDRL-based microgrid energy supply plan* in algorithm 2 runs at the microgrid controller. This initializes the base policy $\pi_\theta(s_t, a_t)$, which is generated from the historical data to collect the observation output $o_t = \langle s_t, a_t, s_{t'} \rangle$ for each time slot t. The microgrid controller collects the energy demands from all associated BSs $\forall i \in \mathcal{B}$ under the microgrid for at each time slot t. This algorithm learns the model $Q^{\pi_\theta}(s_t, a_t)$ to minimize the residual $\min_{\theta} \frac{1}{|\mathcal{O}|} \sum_{o_t \in \mathcal{O}} \frac{1}{2} ||Q^{\pi_\theta}(s_{o_t}, a_{o_t}) - s_{o_{t'}}||^2$. Therefore, the action selection process executes backpropagation algorithm through the model $Q^{\pi_\theta}(s_t, a_t)$ with fully-connected neural networks and accomplishes the first action a_t for observing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

Algorithm 2 Microgrid Energy Supply Plan Based on MDRL

Input: Observation $s_t = \langle \epsilon_t^{dem}, \epsilon_t^{ren}, \epsilon_t^{non}, \epsilon_t^{sto} \rangle$
Output: Observation output $o_t = \langle s_t, a_t, s_{t'} \rangle$
Initialization: Base policy $\pi_\theta(s_t, a_t)$
for $\forall t \in \mathcal{T}$ **do**
  Constraints: (19c), (19d),(19e), & (19h)
  for Until: $\min\limits_{\theta} \frac{1}{|\mathcal{O}|} \sum_{o_t \in \mathcal{O}} \frac{1}{2} ||Q^{\pi_\theta}(s_{o_t}, a_{o_t}) - s_{o_{t'}}||^2$ **do**
    Learn model: $Q^{\pi_\theta}(s_t, a_t)$
    for Until: $max \sum_t R(s_t, a_t)$ **do**
      Choose model: $Q^{\pi_\theta^*}(s_t, a_t)$
      Using: Equations (32), (33), (34), and (40)
      Execute : Action $a_t \in \mathcal{A}$ at $t \in \mathcal{T}$
      Observe : State $s_{t'} = \langle \epsilon_t^{dem}, \epsilon_t^{ren}, \epsilon_t^{non}, \epsilon_t^{sto} \rangle$
      Update : State Action $(s_t, a_t, s_{t'})$ at $t \in \mathcal{T}$
      Append : Observation output o_t at $t \in \mathcal{T}$
    end for
  end for
end for
return $o_t \in \mathcal{O}, \forall t \in \mathcal{T}$

TABLE II: Experimental Environment Summary

Description	Value
Number of microgrids	1
Number of solar units	4
Number of base stations	10
Number of servers in each BS	5
Average tasks request each BS	10000
Number of CPU cores in one server	4 with 1.2 GHz
RAM for one MEC server	1 GB
Disk memory for one MEC server	32 GB
One time slot t	15 minutes
Total number of time slots t	96 (1 days)
Both model runs	Core i7 2.6 GHz with 8 GB RAM
Energy unit	KW/h

the new state (observation) $s_{t'}$. Then, this algorithm appends the observation output $o_t = \langle s_t, a_t, s_{t'} \rangle$ and updates the model. Finally, the energy supply plan will be executed based on the prediction results.

VI. Experimental Result and Discussion

The proposed *multi-access edge server energy supply plan* model is implemented on the Python platform. To evaluate this model, we have used three well-known datasets [49], [50], and [51], where the first two datasets are for computational task requests and the third one for solar energy generation data. Moreover, in our energy consumption model, we have used the energy consumption parameters according to Raspberry Pi 3 Model B as a baseline for each multi-access edge server [52], and also considered edge servers. As a result, we do not need to include a cooling system in our experiment. Table II represents the overall experimental environment.

A. MEC Servers Energy Consumption Numerical Analysis

The energy consumption model has implemented according to the algorithm 1 and compared with the random greedy and greedy methods for the task assignment problem [53], [54]. We preprocessed and combined the CRAWDAD nyupoly/video [49] and CRAWDAD due/packet-delivery [50] datasets regarding heterogeneous task requests, considering the payload size, computational demand and maximum tolerable delay. Our proposed algorithm 1 is executed in real time for cluster-based tasks assignment for a single BS and after one time slot (15 minutes), the total energy consumption results are sent to the microgrid controller to execute the microgrid controller energy supply plan method. Further, all BSs under the microgrid execute this procedure independently.

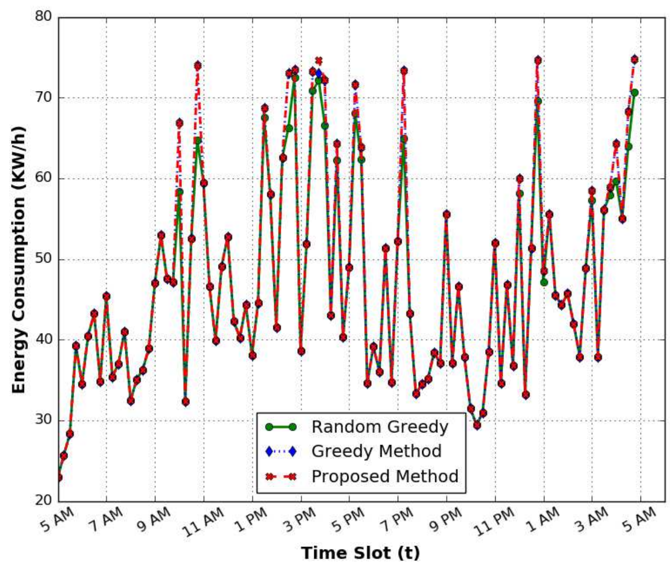


Fig. 5: Computational task execution energy consumption after 24 hours (15 minutes time slots).

Fig. 5 describes the energy consumption result for all three methods. The dashed line with crossed marks (red) shows the DBSCAN-based task execution energy consumption results for 24 hours with 15 minutes time slots, whereas the lines with solid circle marks (green) and dotted diamond marks (blue) represent the random greedy and greedy methods of energy consumption, respectively. We observe more clearly from the empirical distribution function (ECDF) in Fig. 6 that the proposed solution for energy consumption is almost the same as for the greedy method (the probability of having less than 60 KW/h energy consumption is around 0.8). The proposed DBSCAN-based unsupervised learning (algorithm 1) is more robust, less complex, and more reliable for handling the unpredictable task requests. On the other hand, the random greedy method with a uniform policy for energy consumption has a probability of 0.82 to have an energy consumption less than 60 KW/h.

The trade-off between the random greedy and proposed methods is clearly demonstrated in Fig. 7, where the proposed model and greedy methods provide around 97% successful task execution. However, the random greedy task execution success rate is not more than 78%. Additionally, the proposed method on average saves 5.7% and 7.3% energy compared with the random greedy and greedy methods, respectively.

Fig. 8 presents the average server utilization for task execution. Our proposed method assigned tasks to each multi-access

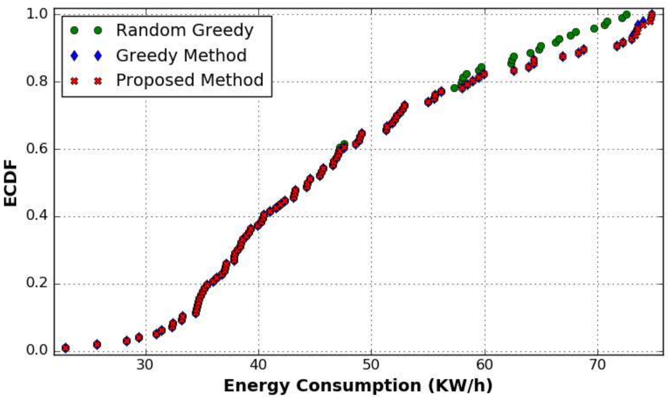


Fig. 6: Energy consumption ECDF for computational task execution.

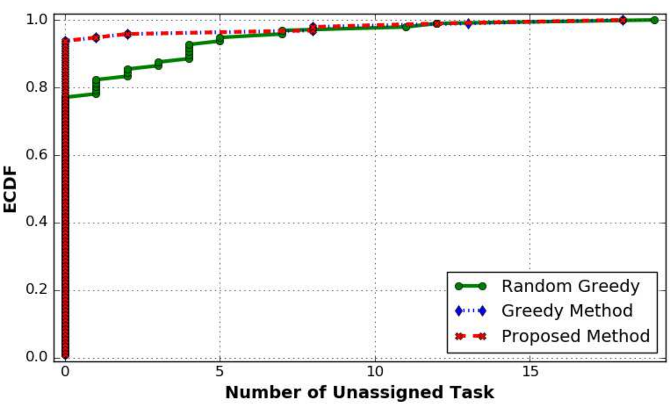


Fig. 7: ECDF for unassigned computational task.

edge server according to the task category, and this method handles unpredictable tasks using the load balancing policy while satisfying the computational and maximum tolerable delay constraints. The proposed method's average percentage of server utilization for 5 multi-access edge servers in one BS is indicated with star marks bar (red color) in Fig. 8. Additionally, the first three severs utilization is higher than the last two servers. However, the proposed method is applicable to any number of task categories because this method uses the concept of density-based spatial clustering of applications with noise (DBSCAN). The uniformly random policy-based, random greedy method server utilization percentage is almost the same for all the servers (crossed marks with green bar), but this method is not capable of fulfilling the task requests due to the computational capacity constraint, which is demonstrated in Fig. 7. Moreover, the dotted yellow bar in Fig. 8 shows that most of the tasks are assigned to the first server, and then proceeds in a sequential manner. Additionally, if the server utilization is higher then the energy consumption is increased. Fig. 9 provides a clear view of how the proposed method has less server utilization compared with the other two methods, and how our proposed method performance is better than that of the greedy method.

We have verified our proposed DBSCAN (unsupervised learning) method using the Silhouette Coefficient score and Calinski-Harabaz index metrics [55], [56]. For unsupervised learning, it is very difficult to evaluate the performance re-

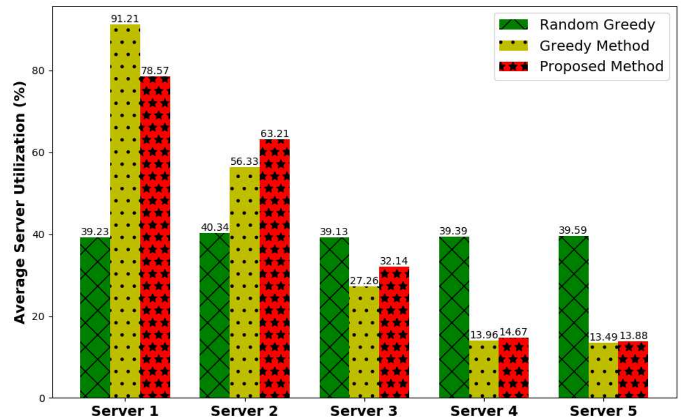


Fig. 8: Average multi-access server utilization for task fulfillment.

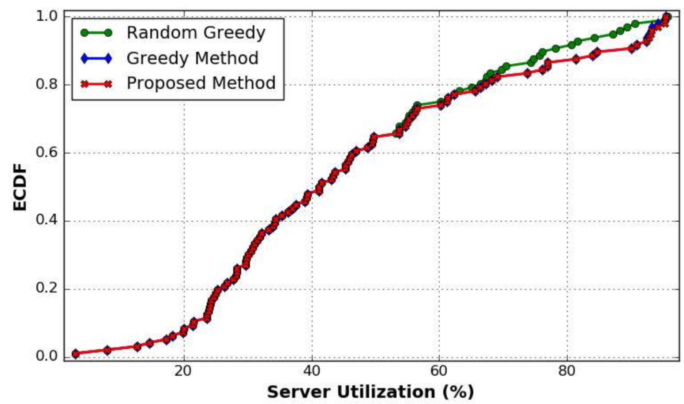


Fig. 9: ECDF for average server utilization.

garding the cluster since the ground truth labels are unknown. However, the evaluation can be performed using only developed model, and the Silhouette Coefficient score and Calinski-Harabaz index metrics are appropriate.

The Silhouette Coefficient score is one of the most well-used technique for such evaluation [55]. A higher Silhouette Coefficient score is associated with better performance. The Silhouette Coefficient score ranges from -1 to 1 and a higher score indicates the clusters are dense and well separated, thus satisfying the standard concept of a cluster formation. The Silhouette Coefficient for a single sample d is given below:

$$d = \frac{b-a}{max(a,b)}, \tag{41}$$

where a denotes the distance between a sample and all other points in the same class, and the mean distance between a sample and all other points in the next nearest cluster is denoted bt b. In addition, the following is the Silhouette Coefficient score for all data in a time slot t:

$$d_t = \sum_{\forall d \in D_t} \frac{b-a}{max(a,b)}. \tag{42}$$

Fig. 10 shows the ECDF (for 96 time slots) of the Silhouette Coefficient scores for task clustering, where the average score is 0.6963, that demonstrating the task clustering of the proposed method performs better. Even though, we have used DBSCAN, this metrics indicates that our proposed method performs better for task categorization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

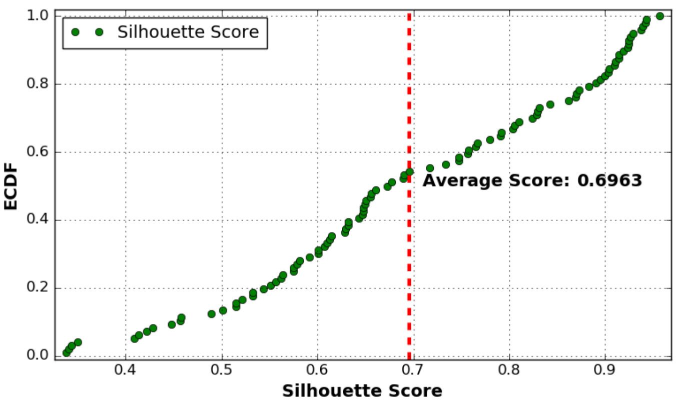


Fig. 10: Silhouette scores for the task cluster performance analysis (96 time slots).

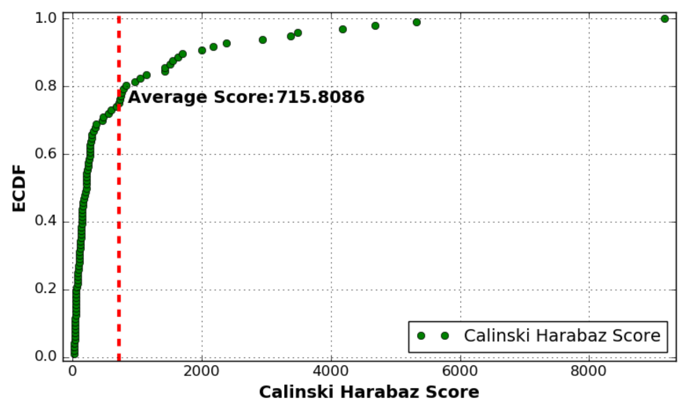


Fig. 11: Calinski-Harabaz score for task cluster performance analysis (96 time slots).

The Calinski-Harabaz index performance metric is the second metric used for further verifying our task clustering performance [56]. This procedure also does not require ground truth labels, so this is appropriate for evaluating DBSCAN (unsupervised learning). The following is the Calinski-Harabaz index score $I(p)$ for the p clusters:

$$I(p) = (\frac{\Xi_p}{\Lambda_p})(\frac{K-p}{p-1}), \tag{43}$$

where K is the total number of task, and Ξ_m and Λ_m are the between group dispersion matrix and dispersion matrix within-cluster, respectively. The values of Ξ_m and Λ_m are measured using the following equations (44) and (45):

$$\Xi_p = \sum_{q=1}^{p} \sum_{a \in \mathcal{V}_q} (a - v_q)(a - v_q)^T, \tag{44}$$

and

$$\Lambda_p = \sum_{q} d_q (v_q - v)(v_q - v)^T, \tag{45}$$

where $\mathcal{V}_q$ is the set of points in cluster q with center v_q, and d_q is the number of points in cluster q.

Fig. 11 presents the ECDF for the Calinski-Harabaz score. For around 80% of the 96 time slots (for 1 day with a 15-minute duration for each time slot), the scores are more than 715, which demonstrates our proposed method performs better for task categorization.

B. Microgrid Energy Supply Plan Numerical Analysis

Model-based deep reinforcement learning (MDRL) for the microgrid energy supply plan has also implemented on the python platform, along with TensorFlow APIs [57]. Therefore, we used the energy consumption output from algorithm 1 as the input for energy demand for each time slot t. We used the UMass solar panel dataset for estimating the renewable energy generation for each day with 96 time slots at 15-minute durations for each [51]. Based on the number of solar panel units for each day, we have divided this dataset into 70% and 30% as training and testing datasets, respectively [24]. Table III describes the parameter and co-efficient values for the proposed microgrid energy supply model. This model usages the first 100 episodes for learning the model from the historical data, and the the policy and model are updated using real observations and the learned model.

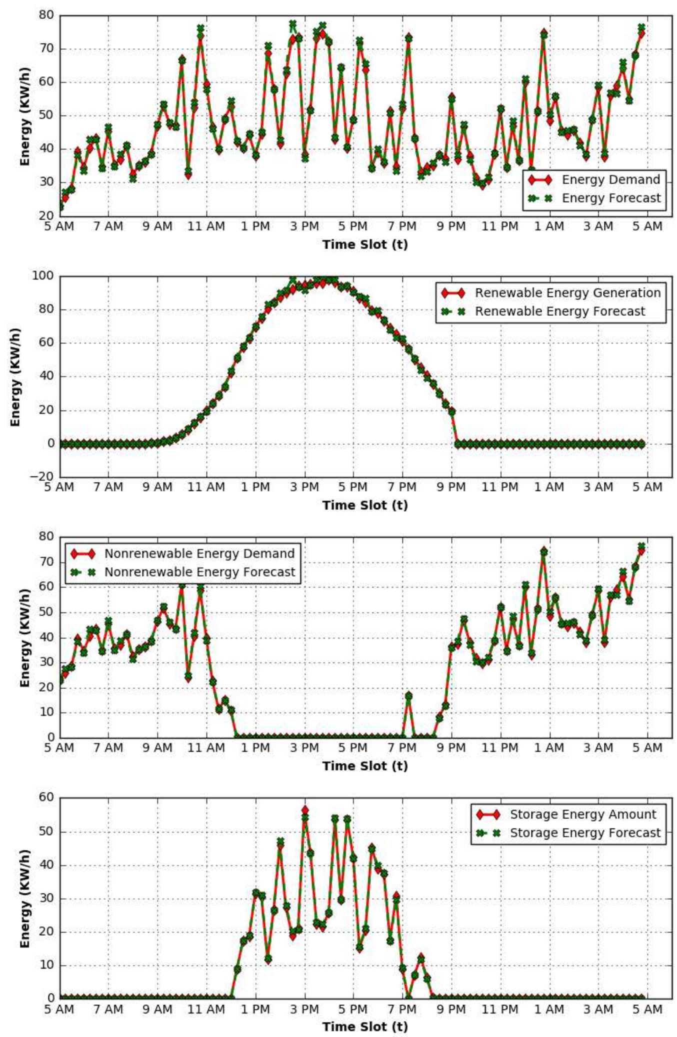


Fig. 12: Energy supply plan for 24 hours using training set.

Fig. 12 and Fig. 13 show the energy supply plan for 24 hours with 15-minute durations using training set and test set, respectively. In Fig. 12, the top sub-figure describes the overall energy demand using red diamond marks with a solid line, and the green cross marks with a dashed line represent the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

TABLE III: MDRL Parameters

Description	Value
Maximum number of episodes	1000
Learning rate, r	10^{-2}
Discount factor, α	0.99
Bias for first moment, ϑ_1	0.90
Bias for second moment, ϑ_2	0.99
Constant value, ε	10^{-8}
Model networks layer size	256
Number of hidden layer neurons	8
Input dimension	4
Batch size from model	3
Batch size from real observation	3
Model learning number of episodes	first 100

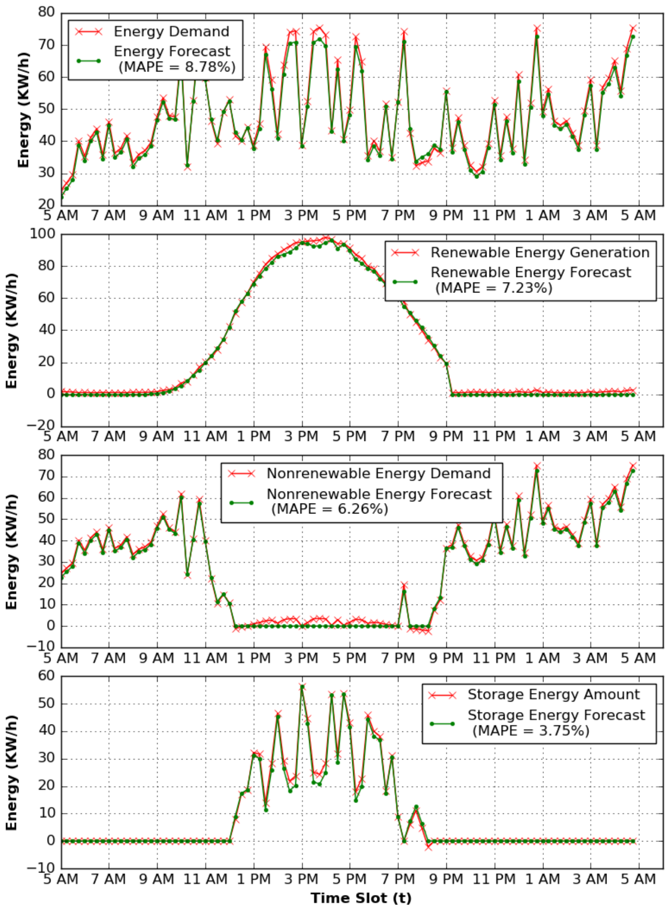


Fig. 13: Energy supply plan for 24 hours using test set (2015-01-11).

energy supply plan using the training model. Similarly, the second, third, and fourth sub-figures present the renewable energy generation, non-renewable energy demand, and the storage amount of the energy forecast, respectively. On the other hand, Fig. 13 shows energy supply plan for 24 hours using the test set, where MAPE (Mean absolute percentage error) for energy demand, renewable, nonrenewable, storage energy are 8.78%, 7.23%, 6.26%, and 3.75%, respectively. Furthermore, Fig. 14 represents the mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and root mean absolute value (RMAE) for the MDRL learning model using training and test sets. In case of the training set, the RMAE error is 1.41%. The reason behind the small amount

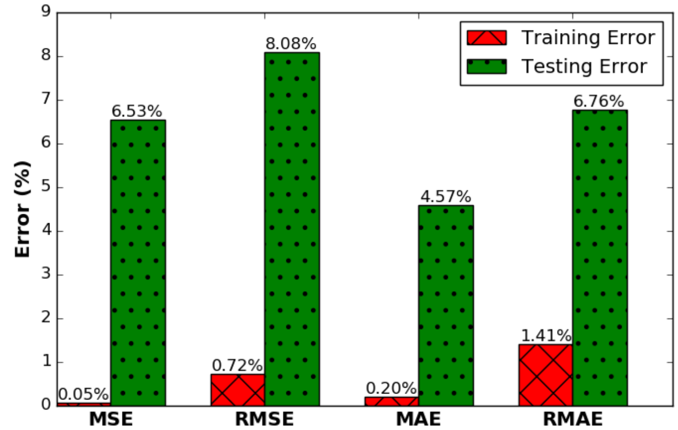


Fig. 14: Training and testing mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and root mean absolute value (RMAE) for the MDRL learning model.

error is, the proposed MDRL learns from the historical data first, where the MDEL model is updated on real environmental observations. However, the RMAE for the test set is around 6.76%, which ensures that the proposed model works well under the uncertainties for both energy demand and generation.

The convergence and reward values are given in Fig. 15, where the first sub-plot shows the convergence of the proposed model using the loss function and the second sub-plot describes the reward value. The reward function is a monotonically increasing function and reaches convergence after the around 160 episodes. We have verified the performance

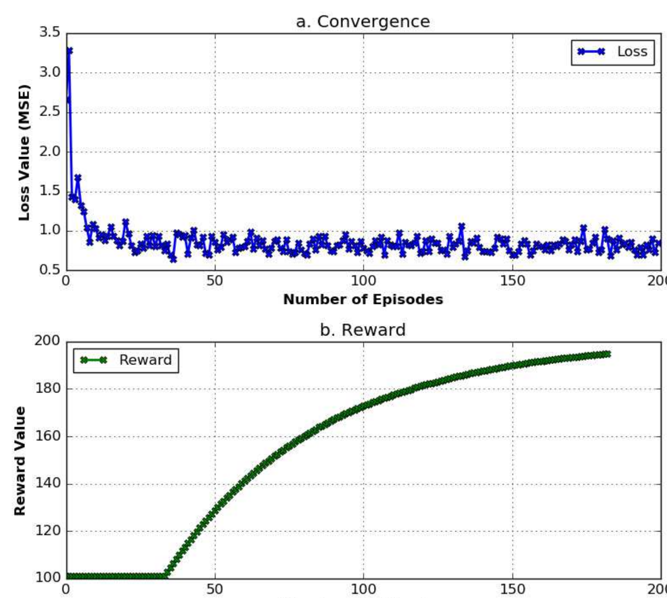


Fig. 15: MDRL convergence and reward value.

of our model using precision recall curves and receiver operating characteristic (ROC) curves, which are commonly used performance metrics for supervised learning model verification [58]. In Fig. 16, the top sub-plot presents the precision recall curve and second sub-plot shows the ROC curve, which verify the training result of the proposed MDRL method.

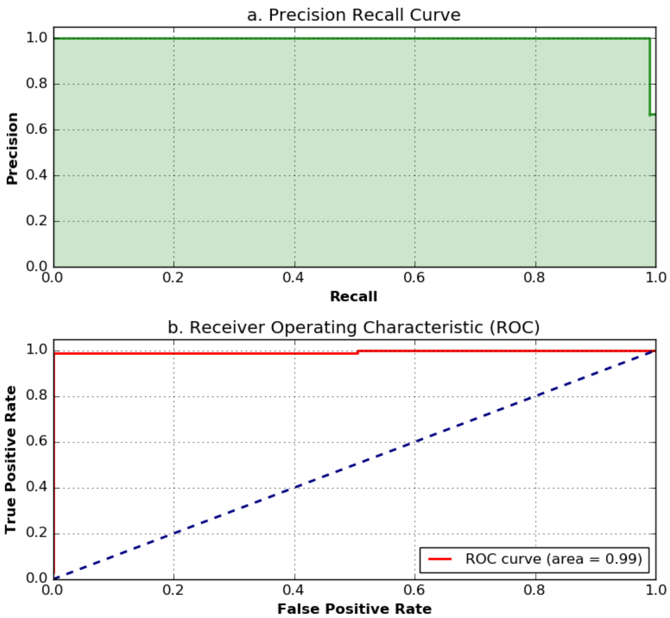


Fig. 16: Precision recall curve and receiver operating characteristic (ROC) curve for the MDRL training model.

VII. Conclusion

In this research, we investigated the MEC network energy consumption while considering computational task loads for the MEC servers, the energy generation characteristics from microgrid energy sources (i.e., renewable, non-renewable, and storages), and the energy supply plan for microgrid-enabled MEC networks. In order to solve the proposed multi-access edge server energy supply plan problem distributedly, first we decompose the proposed problem into two subproblems. Second, we have proposed DBSCAN-based approach for measuring the energy consumption of the network while performing the task assignment at MEC. Consequently, we have applied the MDRL-based mechanism to derive the solution for microgrid energy supply plan, according to the energy demand from the MEC network with the energy generation through microgrid. The simulation results establish that the proposed approaches significantly reduce the energy consumption than that of the greedy and random greedy methods without degrading the performance of the tasks computation at MEC. Additionally, overall energy supply plan exhibits a high accuracy prediction, as a result the proposed approaches mitigate uncertainty for both MEC tasks load and microgrid energy generation. Finally, the proposed approaches ensure the sustainability of the MEC networks by significantly reducing the risk of energy outage due to the nondeterministic task loads.

References

[1] F. Giust et al., "MEC Deployments in 4G and Evolution Towards 5G ,"*ETSI White Papers*, https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp24_MEC_deployment_in_4G_5G_FINAL.pdf, February 2018 (Visited on 3 December, 2018).

[2] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila and T. Taleb, "Survey on Multi-Access Edge Computing for Internet of Things Realization,"*in IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961-2991, Fourthquarter 2018.

[3] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update,"*2016–2021 White Paper*, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, February 2017 (Visited on 3 July, 2018).

[4] K. Zhang et al., "Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks,"*in IEEE Access*, vol. 4, pp. 5896-5907, 2016.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective,"*in IEEE Communications Surveys & Tutorials*,vol. 19, no. 4, pp. 2322-2358, Fourthquarter 2017.

[6] S. Kekki et al., "MEC in 5G networks,"*ETSI White Papers*, https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf, June 2018 (Visited on 4 December, 2018).

[7] C. F. Calvillo, A. Sánchez-Miralles, and J. Villar, "Energy management and planning in smart cities," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 273-287, March 2016.

[8] Ericsson, "Sustainable Energy Use in Mobile Communications," *White paper*, August 2007.

[9] Y. Mao, Y. Luo, J. Zhang and K. B. Letaief, "Energy harvesting small cell networks: feasibility, deployment, and operation," *in IEEE Communications Magazine*, vol. 53, no. 6, pp. 94-101, June 2015.

[10] X. Huang and N. Ansari, "Energy sharing within EH-enabled wireless communication networks," *in IEEE Wireless Communications*, vol. 22, no. 3, pp. 144-149, June 2015.

[11] M. Hong and H. Zhu, "Power-efficient operation of wireless heterogeneous networks using Smart Grids," *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Venice, 2014, pp. 236-241.

[12] 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," *in IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680-692, Jan. 2018.

[13] X. Huang, T. Han and N. Ansari, "Smart Grid Enabled Mobile Networks: Jointly Optimizing BS Operation and Power Distribution," *in IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1832-1845, June 2017.

[14] T. Han and N. Ansari, "A Traffic Load Balancing Framework for Software-Defined Radio Access Networks Powered by Hybrid Energy Sources,"*in IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 1038-1051, April 2016.

[15] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," *in IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016.

[16] S. Shahzadi, M. Iqbal, T. Dagiuklas, "Multi-access edge computing: open issues, challenges and future perspectives,"*Journal of Cloud Computing*, vol. , no. , pp. , December 2017.

[17] C. Li et al., "Towards sustainable in-situ server systems in the big data era,"*2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*, Portland, OR, 2015, pp. 14-26.

[18] D. Huang, P. Wang and D. Niyato, "A Dynamic Offloading Algorithm for Mobile Computing," *in IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991-1995, June 2012.

[19] L. Chen, S. Zhou and J. Xu, "Energy efficient mobile edge computing in dense cellular networks," *2017 IEEE International Conference on Communications (ICC)*, Paris, 2017, pp. 1-6.

[20] D. Li, W. Chiu, H. Sun and H. V. Poor, "Multiobjective Optimization for Demand Side Management Program in Smart Grid," *in IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1482-1490, April 2018.

[21] M. A. Mengistu, A. A. Girmay, C. Camarda, A. Acquaviva and E. Patti, "A Cloud-based On-line Disaggregation Algorithm for Home Appliance Loads," *in IEEE Transactions on Smart Grid*, Early Access, 2018.

[22] N. H. Tran, C. Pham, M. N.H. Nguyen, S. Ren, C. S. Hong, "Incentivizing Energy Reduction for Emergency Demand Response in Multi-Tenant Mixed-Use Buildings," *in IEEE Transactions on Smart Grid*, Vol.9, No.4, pp.3701-3715, July 2018.

[23] C. Pham, N. H. Tran, S. Ren, C. S. Hong, K. K. Nguyen, M. Cherieta, "A Distributed Approach to Emergency Demand Response in Geo-distributed Mixed-use," *Journal of Building Engineering*, Vol.19, pp.506-518, July 2018.

[24] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu and Y. Zhang, "Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network," *in IEEE Transactions on Smart Grid*, Early Access, 2017.

[25] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, ”A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise,”*Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, August 02-04, 1996, Portland, Oregon.

[26] A. S. Polydoros, L. Nalpantidis, ”Survey of model-based reinforcement learning: Applications on robotics,”*Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153-173, May 2017.

[27] M. G. R. Alam, Y. K. Tun and C. S. Hong, ”Multi-agent and reinforcement learning based code offloading in mobile fog,”*2016 International Conference on Information Networking (ICOIN)*, Kota Kinabalu, 2016, pp. 285-290.

[28] M. S. Munir, S. F. Abedin, M. G. R. Alam, N. H. Tran and C. S. Hong, ”Intelligent service fulfillment for software defined networks in smart city,”*2018 International Conference on Information Networking (ICOIN)*, Chiang Mai, 2018, pp. 516-521.

[29] M. Rahimiyan and H. R. Mashhadi, ”An Adaptive Q-Learning Algorithm Developed for Agent-Based Computational Modeling of Electricity Market,” *in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 547-556, Sept. 2010.

[30] Y. Zhang, M. H. Hajiesmaili, S. Cai, M. Chen and Q. Zhu, ”Peak-Aware Online Economic Dispatching for Microgrids,”*in IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 323-335, Jan. 2018.

[31] T. Han and N. Ansari, ”Network Utility Aware Traffic Load Balancing in Backhaul-Constrained Cache-Enabled Small Cell Networks with Hybrid Power Supplies,” *in IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2819-2832, 1 Oct. 2017.

[32] L. Kleinrock, ”Queueing Systems,” *New York:Wiley*, vol. I, 1975.

[33] L. Chen, S. Zhou and J. Xu, ”Computation Peer Offloading for Energy-Constrained Mobile Edge Computing in Small-Cell Networks,”*in IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1619-1632, Aug. 2018.

[34] 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,” *in IEEE Transactions on Communications*, (Early Access), 18 Sep. 2018.

[35] R. Bertran, M. Gonzalez, X. Martorell, N. Navarro and E. Ayguade, ”A Systematic Methodology to Generate Decomposable and Responsive Power Models for CMPs,” *in IEEE Transactions on Computers*, vol. 62, no. 7, pp. 1289-1302, July 2013. Firstquarter 2016.

[36] Y. Sun, S. Zhou and J. Xu, ”EMM: Energy-Aware Mobility Management for Mobile Edge Computing in Ultra Dense Networks,”*in IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637-2646, Nov. 2017.

[37] G. Auer et al., ”How much energy is needed to run a wireless network?,” *in IEEE Wireless Communications*,vol. 18, no. 5, pp. 40-49, October 2011.

[38] J. Xu, L. Chen and S. Ren, ”Online Learning for Offloading and Autoscaling in Energy Harvesting Mobile Edge Computing,”*in IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361-373, Sept. 2017.

[39] W. C. Chan, T. C. Lu and R. J. Chen, ”Pollaczek-Khinchin formula for the M/G/1 queue in discrete time with vacations,”*in IEE Proceedings - Computers and Digital Techniques*, vol. 144, no. 4, pp. 222-226, July 1997.

[40] S. L. Song, K. Barker, D. Kerbyson, ”Unified performance and power modeling of scientific workloads”, *Proc. 1st Int. Workshop E2SC*, pp. 4:1-4:8, 2013. 56-75, 2013.

[41] M. Sozio, and A. Gionis, ”The community-search problem and how to plan a successful cocktail party,” *In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD ’10)*, pp. 939-948, 2010, ACM, New York, NY, USA.

[42] C. Ruiz, M. Spiliopoulou, and E. Menasalvas, ”C-dbscan: Density-based clustering with constraints,” *In Proceedings of the 11th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, pp. 216-223, 2007.

[43] J. Suh and D. F. Hougen, ”The context-aware learning model: Reward-based and experience-based logistic regression backpropagation,”*2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, 2017, pp. 1-8.

[44] R. S. Sutton, and, A. G. Barto, ”Reinforcement Learning: An Introduction,”*The MIT Press*, Cambridge, MA, 1998.

[45] V. Nair , G. E. Hinton, ”Rectified linear units improve restricted boltzmann machines,”*Proceedings of the 27th International Conference on International Conference on Machine Learning*, p.807-814, June 21-24, 2010, Haifa, Israel.

[46] M. N. Gibbs and D. J. C. Mackay, ”Variational Gaussian process classifiers,”*in IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1458-1464, Nov 2000.

[47] D.P. Kingma, and J. Ba, ”Adam: A Method for Stochastic Optimization,”*in Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, Dec 2014.

[48] R. J. G. B. Campello, D. Moulavi, and J. Sander, ”Density-Based Clustering Based on Hierarchical Density Estimates,”*Advances in Knowledge Discovery and Data Mining*, pp 160-172, 2013.

[49] F. Fund, C. Wang, Y. Liu, T. Korakis, M. Zink, and S. Panwar, ”CRAWDAD dataset nyupoly/video (v. 2014-05-09)”, *downloaded from https://crawdad.org/nyupoly/video/20140509*, May 2014 (Visited on 3 July, 2018).

[50] S. Fu, and Y. Zhang, ”CRAWDAD dataset due/packet-delivery (v. 2015-04-01)”, *downloaded from https://crawdad.org/due/packet-delivery/20150401*, Apr 2015 (Visited on 3 July, 2018).

[51] Online:”Solar panel dataset”, *UMassTraceRepository http://traces.cs.umass.edu/index.php/Smart/Smart*, (Visited on 3 July, 2018).

[52] Online: ”Power Consumption Benchmarks”, *Raspberry Pi Dramble*, https://www.pidramble.com/wiki/benchmarks/power-consumption (Visited on 3 June, 2018).

[53] T.A. Feo, and M.G.C. Resende, ”Greedy Randomized Adaptive Search Procedures,”*Journal of Global Optimization*, vol. 6, no. 2, pp. 109-134, 1995.

[54] R. Cohen, L. Katzir, and D. Raze, ”An Efficient Approximation for the Generalized Assignment Problem,”*Information Processing Letters*, vol. 100, no. 4, pp. 162-166, November 2006.

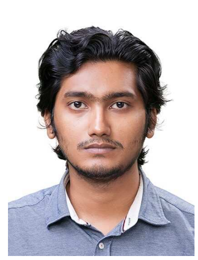
[55] P. J. Rousseeuw, ”Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,”*Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65, November 1987.

[56] T. Calinski, and J. Harabasz, ”A dendrite method for cluster analysis,”*Communications in Statistics* , vol. 3, no.1, pp. 1-27, Sep 1972, Published online: 27 Jun 2007.

[57] Online: All symbols in TensorFlow, *TensorFlow*, https://www.tensorflow.org/api_docs/python/ (Visited on 3 July, 2018).

[58] Online: ”Model evaluation: quantifying the quality of predictions”, *scikit-learn*, http://scikit-learn.org/stable/modules/model_evaluation.html (Visited on 3 August, 2018).

Md. Shirajum Munir received a B.S. degree in Computer Science and Engineering from Khulna University, Bangladesh, in 2010. He served as a Lead Engineer at the solution laboratory, Samsung R&D Institute, Bangladesh, from 2010 to 2016. Since March 2017, he has been working toward his Ph.D. in Computer Science and Engineering at Kyung Hee University, South Korea. His research interests include the IoT network management, fog computing, mobile edge computing, software-defined networking, smart grid, and machine learning.



Sarder Fakhrul Abedin (S’18) received his B.S. degree in Computer Science from Kristianstad University, Kristianstad, Sweden, in 2013. Since March 2014, he has been working toward a Ph.D. degree in Computer Science and Engineering at Kyung Hee University, South Korea. He was the recipient of a scholarship for his graduate study at Kyung Hee University, in 2014. His research interests include Internet of Things (IoT) network management, Cloud computing, Fog computing, and Wireless sensor networks. Mr. Abedin is a Member of the KIISE.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2899673, IEEE Internet of Things Journal

Nguyen H. Tran (S'10-M'11-SM'18) received the BS degree from Hochiminh City University of Technology and Ph.D. degree from Kyung Hee University in electrical and computer engineering in 2005 and 2011, respectively. Since 2018, he has been with the School of Computer Science, The University of Sydney, where he is currently a Senior Lecturer. He was an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University, Korea from 2012 to 2017. His research interests include applying analytic techniques of optimization, game theory, and stochastic modeling to cutting-edge applications such as cloud and mobileedge computing, data centers, heterogeneous wireless networks, and big data for networks. He received the best KHU thesis award in engineering in 2011 and best paper award at IEEE ICC 2016. He has been an Editor of 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.

Choong Seon Hong (S'95-M'97-SM'11) 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, Minato, Japan, in 1997. In 1988, he joined Korea Telecom, where he worked on broadband networks as a Member of the Technical Staff. In September 1993, he joined Keio University. He worked for the Telecommunications Network Laboratory, Korea Telecom, as a Senior Member of the Technical Staff and the Director of the Networking Research Team until August 1999. Since September 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 is a member of ACM, IEICE, IPSJ, KIISE, KICS, KIPS, and OSIA. He has served as the General Chair, a TPC Chair/Member, or an Organizing Committee Member for international conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, TINA, SAINT, and ICOIN. In addition, he is currently an Associate Editor of the IEEE Transactions on Network and Service Management, International Journal of Network Management, and Journal of Communications and Networks as well as an Associate Technical Editor of the IEEE Communications Magazine.