IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 1 May 2024, accepted 19 May 2024, date of publication 22 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3404222

RESEARCH ARTICLE

Enhancing Sustainable Edge Computing Offloading via Renewable Prediction for Energy Harvesting

MOHAMMED ALHARTOMI[1], (Member, IEEE), ADEB SALH[2], LUKMAN AUDAH[3], (Member, IEEE), SAEED ALZAHRANI[1], (Member, IEEE), AND AHMED ALZAHMI[1]

[1]Department of Electrical Engineering, University of Tabuk, Tabuk 71491, Saudi Arabia
[2]Faculty of Information and Communication Technology, University Tunku Abdul Rahman, Kampar, Perak 31900, Malaysia
[3]Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja 86400, Malaysia

Corresponding authors: Adeb Salh (adebali@utar.edu.my) and Mohammed Alhartomi (malhartomi@ut.edu.sa)

This work was supported in part by the Deputyship for Research and Innovation, Ministry of Education, Saudi Arabia, under Project S-1443-0183; and in part by the Universiti Tunku Abdul Rahman Research Fund (UTARRF) Fund through the under Grant 6557/2A02.

ABSTRACT The integration of Edge Computing (EC) and Energy Harvesting (EH) technologies has facilitated the growth of the Internet of Things (IoT), allowing for the interconnectivity of a wide range of devices. The integration of this technology has not only enhanced energy sustainability but also significantly extended the battery life of these devices. Adopting Renewable Energy (RE) sources has become more widespread in energy systems as a strategy to reduce carbon emissions. Low energy consumption and constrained battery capacity for IoT devices are concerns related to offloading. The unpredictability of RE quality makes it difficult for edge servers to maintain high quality of service in EH-EC systems, which impedes effective energy conservation for IoT. To solve an optimization problem, RE Predictions with a Deep Reinforcement Learning algorithm named (REP-DRL) are proposed. Accurately, REP-DRL used the actor-critic technique to identify the best approach for predicting RE and optimal offloading decisions. The approach improves IoT device processing and expands the system state to offload experiences per time slot. To store excess energy during periods of abundance and use it during times of higher demand, the service offloading process is modelled based on the predicted amount of RE, to find the best service offloading technique and improve energy sustainability for IoT. By determining the most efficient service offloading approach using the predicted RE amount, this solution increases the energy sustainability of the IoT ecosystem. Finally, the simulation results show that the REP-DRL system utilizes local computing to conserve power when both the battery level and projected EH are low, showcasing its capacity to adapt to varying operating conditions and optimize the utilization of resources.

INDEX TERMS Edge computing, energy harvesting, IoT, RE, DRL.

I. INTRODUCTION

The rapid development of the Internet of Things (IoT) is impeded by limited device processing. Sustainably offloading Edge Computing (EC) with Renewable Energy (RE) is necessary to solve these problems. This calls for an integrated approach that considers resource allocation strategies, hardware design, offloading strategies, and energy sources [1], [2]. Implementing such a strategy can not only have a positive impact on the environment but also enhance the efficiency and reliability of IoT applications. IoT technologies can help reduce carbon footprints, resulting in more efficient energy generation [3], [4].

The power sector is increasingly shifting towards RE sources due to their affordability and significant contribution to reducing carbon emissions. Solar, wind, hydroelectric,

The associate editor coordinating the review of this manuscript and approving it for publication was Alon Kuperman.

biomass, and other types of RE are all included in the RE sources [5], [6], [7]. Integrating IoT technology into energy systems and utilizing the data collected from interconnected devices can offer a solution to long-standing issues such as energy sustainability. By effectively managing the linked devices, the potential benefits of IoT can be harnessed to optimize energy usage and minimize waste. Energy Harvesting (EH) is a highly promising technology that has the potential to significantly extend the battery life of IoT devices while also ensuring a satisfactory quality of experience. IoT increases energy usage since it uses electricity to communicate with other IoT devices [5], [8]. Due to the insufficient processing power of IoT devices, computing services are often assigned to the cloud center [5]. This avoids significant energy consumption caused by data transmission over long distances between IoT devices and the cloud center, which can also lead to energy depletion in IoT devices with short battery lives. By harnessing energy from the environment, such as from solar panels or motion sensors, EH technology can reduce the dependence on traditional power sources and enable IoT devices to operate for longer periods without the need for frequent battery replacements or recharging [9]. The EH module allows an IoT device to collect ambient RE from diverse sources like solar radiation, wind power generation, and Radio-Frequency (RF) transmissions to supply the IoT Central Processing Unit (CPU) and radio transceiver with cleaner energy [10]. A possible paradigm to address the problem of energy shortage is EC. The EC has the potential to play a big role in fostering a more sustainable future by lowering the amount of energy used in data centers, enabling energy-efficient applications, and assisting in the management of RE sources. On low-power devices, EC encourages the real-time execution of several computing tasks [11], [12], [13], [14]. Edge Servers (ES) adjacent to IoT devices are distributed processing, storage, and bandwidth resources by EC [15]. The EC enables service delivery optimization by shifting service requests to ESs, resulting in enhanced Quality of Service (QoS) for end-users. In addition to this, it contributes to the sustainability of the energy system by reducing transmission latency, avoiding network congestion, minimizing energy consumption, and extending the battery life of IoT devices [16].

A growing number of energy systems are adopting micro-grids to reduce their carbon footprints by using RE as their primary power source, microgrids offer an environmentally friendly alternative [17]. RE generation's pivotal role in renewable-powered energy systems highlights the significance of accurately predicting its amount in improving the QoS of ESs. Moreover, this can facilitate the service offloading process [18]. New developments in Deep Learning (DL) technology are required to create accurate learning models that can be utilized in the energy analysis process, such as prediction, forecasting, and decision-making, to enhance the distribution of RE sources. Our goals are to develop a DL-based enhanced RE prediction system and to establish a service offloading system for EC IoT-enabled systems. This scheme will be built based on the predicted amount of RE generation, ensuring optimal utilization of resources and efficient energy management.

A. RELATED WORKS

Based on previous research on RE supply and environmental health by developing an energy consumption model that incorporates solar energy collection and energy storage using batteries [4]. Information sharing among all objects has been made possible by the emergence of IoT [19]. Energy sustainability can be greatly increased by integrating IoT into energy systems [20]. To employ an offloading framework to transfer computation tasks from an IoT device to an ES, with joint optimization of task allocation and CPU frequency. This optimization aims to minimize both implementation delay and energy consumption, thereby improving overall system efficiency. IoT terminals are frequently challenged when processing a large volume of data due to their typical compact size and low power consumption. Although replacing the batteries can significantly extend a device's service life, doing it at a distant location is difficult, expensive, and time-consuming [21]. To assure sustainable energy production, the authors in [22] integrated an industrial Internet of unmanned aerial systems with the smart grid. However, a recurring flaw in this research is that they never integrated EC into the energy system. A learning-based online method is being investigated for EH-powered ES so that the system operator can determine how much workload should be transferred from the ES to the central cloud and how quickly the ES should process requests [22]. This analysis is based on data regarding core network congestion, computation workload, and the state of the energy grid. A computational offloading technique that considers the low battery capacity of IoT devices is an effective way to extend the lifetime of IoT systems, given the restricted compute capacity of the mobile EC server. To determine the best offloading policy, the authors of [10] suggested using Deep Reinforcement Learning (DRL). The rationale for using DRL to estimate EH and adjust battery levels justifies its use in RE systems. For controlling the inherent variability in RE sources, DRL algorithms are a good fit. To optimize battery usage, DRL can provide adaptive control systems that use data to predict EH patterns and make decisions in real-time. This method improves the system's overall sustainability and efficiency while also maximizing the use of RE. The Markov Decision Process (MDP) was developed as a solution to the problem of large-time complexity in learning algorithms.

Enhance the smart devices' processing capability while saving battery-level power. The problem is the variable generation of RE and the energy used, which affects the battery level. To enhance battery level the [11] proposed a deep Q noisy neural network model that automatically predicts changes in the volume to adjust the level of noise in smart devices to save battery energy. The authors in [12] improve IoT QoS and prolong battery life. IoT fog systems enabled by EH bring important offloading challenges to light. IoT devices are assumed to fully grasp system states in existing works. Consequently, [12] investigates partially visible, decentralized offloading in IoT fog systems with EH enabled. By offloading decisions through a decentralized, partially observable Markov decision process, several IoT devices collaborate for the best possible network performance and quality of experience. EC has arisen as a potent paradigm for furnishing IoT devices with top-tier computing services. This approach addresses the inherent limitation of substantial computational capacity in the majority of IoT devices [23]. The authors of [24] introduced EC and edge IoT to address the scalability challenges of traditional IoT architecture in energy systems. This technique integrates EC into the IoT infrastructure, providing a viable solution for scaling IoT systems. To select whether to offload to an edge device that serves many users, the authors of [25] presented a binary EC offloading mechanism known as DRL. Compared to complete offloading and non-offloading systems, this can decrease energy consumption and average estimation delay. In addition, the dynamic mobile EC network has implemented offloading solutions using DRL algorithms. An effective DRL-based resource management strategy was recommended by [26]. With the help of this plan, real-time workload dumping rates will be improved for the ES and the centralized cloud, respectively. It also emphasizes providing services to reduce service delays and manage operational costs successfully. However, in [27], the authors propose an offloading strategy based on DRL for an IoT devices with EH capabilities. The strategy involves selecting the optimal edge device and offloading rate based on the device's current battery level, thus maximizing its energy efficiency. In [27], the authors propose a DRL-based offloading strategy that aims to accelerate the learning process, increase transmission rates to each edge device, and predict the amount of energy that can be harvested. This approach leverages DRL techniques to optimize the offloading decision-making process, resulting in faster and more efficient EC. To enhance the lifespan of IoT devices, researchers in [28] have proposed a Deep Neural Network (DNN) architecture incorporating regularization techniques and an energy-efficient computation offloading with EH-IoT. The primary goal of this design is to accelerate the convergence rate and optimize Resource Allocation (RA) by establishing a suitable offloading strategy to extend the battery life of IoT devices while maintaining an acceptable level of performance [12], [28]. In [29], the authors propose an online RA method for IoT systems that considers real-time tasks and channel states, as shown in Table 1. To improve the system's energy efficiency, an approach driven by offline optimization is employed. This approach utilizes the Lagrangian dual method combined with a sliding-window-based online RA to optimize both the offloading decision and RA. By considering both factors, this joint approach more effectively minimize the system's overall energy consumption, leading to better performance and reduced costs. This technique is well-suited for real-world applications where energy efficiency is a critical concern. In [30], the authors propose a service offloading technique that leverages RE predictions to reduce the instability and discontinuity of renewable generation. By utilizing memory cells, Long Short-Term Memory (LSTM) effectively handles the issue of exploding and vanishing gradients in DNN, facilitating long-term prediction. Prediction time series data is one of the many sequence prediction tasks where this approach has shown efficacy. By considering these predictions, this approach can optimize the offloading decision-making process and lower the ES-QoS. By reducing the reliance on non-RE sources. The proposed MDP is used by these authors [30] to improve the service of offloading procedure.

In [14], the authors develop an online CPU frequency for sustainable edge computing by minimizing the execution time of tasks offloaded. To balance energy consumption and computational latency, the authors in [14] proposed an optimal offline strategy as well as a prediction value for the energy that can be harvested by optimizing energy efficiency while fulfilling performance requirements, changing service offloading decisions based on real-time RE predictions, and adjusting the clock frequency. The asynchronous advantage actor-critic is used with the anticipated amount of RE to choose the best service offloading method. Renewable generation plays a critical role in RE systems, and predicting the amount of renewable generation can lead to improved QoS for ES. Additionally, such predictions can guide the process of service offloading [16]. Over the past few years, DL has exhibited remarkable success across various tasks, such as predicting stock market trends [31] and predicting traffic flow [32]. EC research is confronted with two primary challenges: (1) One of the major optimization challenges is lowering the energy usage of IoT devices. To formulate this problem, RA and offloading decisions must have significant computational overhead and time expenses. This strategy aims to reduce overall energy consumption by concurrently optimizing compute offloading options, modifying the time allocation ratio for EH, and utilizing the device's local CPU processing capabilities. (2) IoT terminal batteries have a limited capacity, and task processing uses energy. In [33], a novel technique for energy-efficient IoT systems with EC is leveraged to determine the optimal service offloading by proposing renewable prediction-driven service offloading. The goal is to predict the availability of RE to reduce reliance on conventional power sources. A DL-based approach for predicting RE data is used to find the best service offloading approach. This approach involves extending the system state, to include the offloading experience sequence at each time slot based on the proposed RE Predicted with a Deep Reinforcement Learning algorithm named (REP-DRL), which leverages RE predictions to optimize service offloading. Thus, the magnitude of RE generation anticipated necessitates the development of a DL-based strategy for accurate

TABLE 1. Summary of several works related to sustainable edge computing offloading.

References	Optimization Goal	RE	QoS	Latency	Task Offloading	EH	Proposed	Applications
[3]	Reducing system costs depends on designing a data offloading and RE-aware system.	√	×	×	√	√	Proposed a DRL based on considering RE and energy efficiency for IoT devices in dynamic environments to optimize data offloading and minimize latency.	Perform local processing and offload tasks.
[5]	Reduce energy consumption and latency.	×	√	×	×	×	Design a stochastic game-based resource allocation algorithm with prioritized experience replays.	Offloading policy.
[10]	Efficient computation offloading to generate high data.	√	×	√	×	√	IoT devices' battery life is limited; EH makes them last longer. Data and energy unpredictability in offloading is modeled as MDP so that optimal policies can be learned through DRL.	Optimal offloading policy.
[11]	Minimize energy consumption and latency.	√	×	×	√	√	Limit latency and energy consumption in EC networks based on the proposed novel DRL approach using noisy neural networks to save battery energy.	Execution time of computation task.
[12]	Increase the renewable battery level and better Quality of Experience (QoE).	√	√	×	×	√	The proposed offloading challenges seek an approximate optimal solution to optimize network performance while satisfying QoE s by solving a learning-based decentralized offloading method.	Optimal decision offloading on the current number tasks.
[13]	Reduce energy consumption and computing latency for the mobile IoT.	√	×	√	×	√	Limited energy storage of the cloudlet and the dynamic intensity of renewable energy, and then develop offline optimal CPU frequency scaling policies that maximize the computing power.	Execution time of tasks offloaded.
[22]	Improve execution latency and solve a problem in each time slot.	√	×	√	×	×	Compute offloading takes into account the transmission power concerning the battery energy level, the offloading decision, and the execution latency.	Performance for execution cost.
[23]	Maximum energy efficiency performance.	√	×	×	×	×	Utilize a game-theoretic method for compute offloading to minimize energy costs and smart device job execution times while simultaneously optimizing the value of the service providers.	Lower the smart devices' energy costs and execution times.
[24]	Minimize interference channel and transmission power.	×	×	√	√	√	A game theoretic analysis is conducted on compute offloading from multi-mobile devices to multi-mobile EC-s in heterogeneous mobile EC systems with energy harvesting.	Time efficiency improvement.
[27]	Minimize latency and enhance energy sustainability.	√	×	√	×	×	Provide a resource management that is effective and based on DRL that learns the best course of action for offloading dynamic workloads in mobile EC.	Optimal offloading policy.
[28]	Reduce energy consumption and the computation latency.	×	×	√	×	×	Proposed to use DRL to direct IoT EH devices in selecting edge devices and offloading rates based on battery level, historical transmission rates, and anticipated harvested energy.	Optimize the offloading policy.
[29]	Develop eco-friendly EC and increase processing energy efficiency for sustainability.	√	×	×	√	√	Proposes an IoT offloading system with harvesting that saves energy. By taking into account decisions about offloading, time allocation, and local resources, it optimizes the consumption of energy.	Offloading decision.
[31]	Improve the sustainability of energy for energy systems.	√	×	×	×	√	Proposed DL to predict RE sources with an asynchronous advantage actor-critic to determine the best offloading process.	Optimal service offloading.
[33]	Enhance the continuity of renewable generation and improve QoS.	√	×	×	√	√	Proposed a novel technique for energy-efficient to determine the optimal service offloading by proposed renewable prediction-driven service offloading.	Optimal service offloading.
This paper	Improve energy sustainability and enhance the battery life of IoT devices.	√	√	√	√	√	Proposed REP-DRL to provide the best service offloading technique and improve energy sustainability for IoT.	Provide the best service offloading.

prediction, and the creation of an IoT-enabled energy system service offloading scheme utilizing EC technology.

The instability and discontinuity of renewable generation, low energy consumption, and extending the battery life of IoT devices are concerns related to offloading. The unpredictability of RE quality makes it difficult for ES to maintain high QoS in EH-EC systems, which impedes effective energy conservation for IoT.

This work focuses on building a service offloading approach with RE prediction for IoT-enabled energy systems powered by RE sources. This study's main goal is to construct a service offloading method with RE. This article focuses on enhancing energy sustainability for edge-enabled IoT energy systems, not many of these studies have integrated EC and RE prediction as an integrated whole. This could lead to unpredictable QoS for the edge servers. This essay explores the issue of observable computation offloading in the context of the EH-enabled IoT EC system to close this gap. This depends on the energy stored in excess during times of abundance and used when demand is higher. To find the best service offloading methods and improve energy sustainability for IoT applications, the service offloading process is envisioned by utilizing the predicted RE amount. To clarify the difference between this work and previous works, this approach combines the use of the actor-critic with expected renewables, it improves IoT processing. Energy consumption is decreased, and RA is optimized REP-DRL for RE prediction. DNN is used by REP-DRL to make the best offloading decisions and improve energy sustainability.

B. MOTIVATION AND CONTRIBUTIONS

To date, there has been limited research that has effectively coordinated RE prediction-driven service offloading mechanisms. To improve the energy sustainability of IoT and enhance the extended battery life of these devices. To address this issue in the [24], [25], [26], [27], [28], [29], [30], and [31], we have developed REP-DRL, a novel approach that leverages RE predictions to optimize service offloading. Our work represents a significant advancement in this field. The main contributions of this paper are listed as follows.

- To address the challenge of reducing the energy consumption of IoT devices, an optimization problem is formulated that considers RA and offloading decisions. To enable local processing and compute offloading, IoT devices use RE sources like solar energy to generate their energy. By simultaneously maximizing the choice of compute offloading, the time allocation ratio for EH, and the device's local CPU processing capabilities, this approach aims to lower overall energy usage.
- Offloading raises concerns regarding IoT devices' low energy consumption and constrained battery capacity. Maintaining high QoS in EH-EC systems is difficult for ES due to the unpredictable nature of RE quality. To optimize the allocation of resources and lower the energy consumption. This depends on the proposed REP-DRL. The REP-DRL algorithm achieves the best offloading decision based on DNN. Where the DNN creates binary offloading decisions to faster up and stabilize convergence for LSTM to solve accelerated convergence by updating the gradient function based on time series prediction.
- To enhance the processing performance of IoT devices, we develop a service offloading approach by combining the asynchronous advantage of actor-critic with the anticipated amount of renewable generation. REP-DRL used the actor-critic technique to identify the best approach for predicting RE and optimal offloading decisions. This approach involves extending the system state, to include the offloading experience sequence at each time slot. By doing so, the optimal offloading destination is determined by allocating as many resources as possible to handle the offloading process from action to execution and improve energy sustainability.

II. SYSTEM MODEL

In this section, we proposed an EC offloading framework in IoT to enable EH-IoT as shown in Fig. 1, wherein an IoT device that has EH and electricity-storage components can offload some of its processing tasks to one of M ES that are each individually connected to Base Stations (BS). The IoT device has a battery that is powered by energy that is harvested from the environment, such as solar, wind, and RF signals. Let $\mathrm{M}\in\mathcal{M} = [1, \ldots, \mathcal{m}]$ signify the indices of ES, and each device generates a task with a time-slots of $\mathcal{t}\in j = [1, \ldots, \mathtt{T}]$, where the duration of each time-slot $\mathcal{t}$ is τ seconds. Considering the randomized time-slots for tasks and the constraints of available local computing resources, the device will determine whether to offload a given task or not. The offloading choice made by the device is represented by.

$$\delta_j = \begin{cases} 0, & \textit{Offload } j \textit{ to local computing mode} \\ 1, & \textit{Offload } j \textit{ to the ES} \end{cases} . \tag{1}$$

This system uses orthogonal frequency division multiplexing access (OFDMA), with each subchannel having a bandwidth of $\mathcal{B}$. Based on radio link transmission, the IoT device chooses an ES to offload the compute tasks during time-slot $\mathcal{t}$. The rate of transmission between the $\mathcal{m}-th$ ES and the IoT device is given by

$$\mathcal{r}_j^{\mathcal{m}} = \mathcal{B}\, log_2\left(1 + \frac{\mathcal{P}_j \hbar_j}{\sigma^2}\right), \tag{2}$$

where $\hbar_j$ represents the channel gain, $\mathcal{P}_j$ represents the transmit power of the device for offloading computations, and σ^2 is the Gaussian white noise power of the ES. Every IoT device j performs the task within the time slot $\mathcal{t}$, which is represented by C_j (in kbit) as:

$$\mathcal{t}_j^{loc} = \frac{C_j \mathbb{r}}{f_j^{loc}}, \tag{3}$$

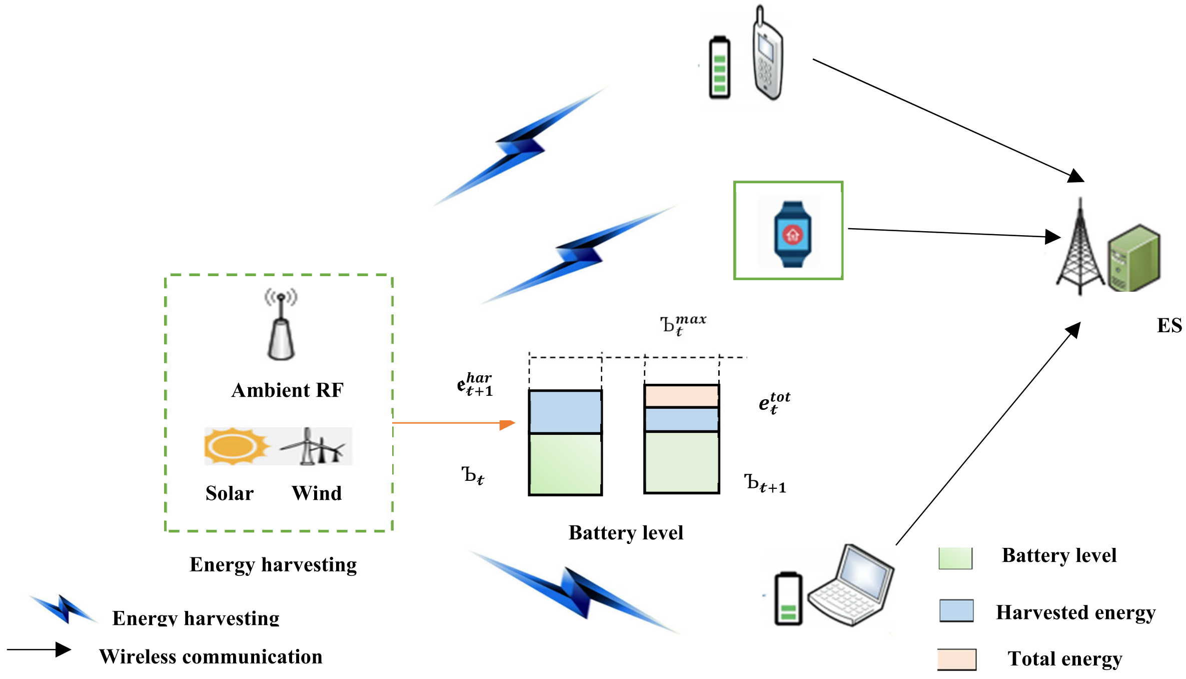

FIGURE 1. EH powered for IoT device with RE.

where $\mathbb{r}$ is the total of CPU cycles needed to complete a 1 kbit operation, and f_j^{loc} is denoted by the CPU frequency of a computing resource, the processor frequency has a super-linear relationship with processing power. The energy needed for local computation at time-slot t can be written as:

$$\mathrm{e}_j^{loc} = \varsigma C_j \mathbb{r} (f_j^{loc})^2. \tag{4}$$

According to [1] the chip architecture-dependent effective capacitance coefficient ς. The time it takes for $C_j \mathbb{r}$ tasks to be offloaded to the $m - th$ ES-QoS at time-slot t can be written as:

$$t_j^{off} = \frac{C_j}{(1 - \partial_j)\, r_j^{m}}, \tag{5}$$

where ∂ indicate the offloading rate selected by the IoT device at the time-slot t, and denoted by ∂_j and $1-\partial_j$, where If $\partial_j = 0$, the calculations are performed locally, and fully offloading to the ES if $\partial_j = 1$. The amount of energy needed for the device j to transfer the task to the ES is given by

$$\mathrm{e}_j^{off} = \mathcal{P}_j t_j^{off}. \tag{6}$$

Using an EH module and an internal battery, the IoT devices may harvest energy from renewable sources, such as solar energy. During times of favorable wind speeds and low electricity demand, the proposed algorithm can efficiently store the output power generated by the wind farm and simultaneously inject it into the grid. Hence, let's define e_j^{har} and Ъ_j as the total amount of EH at time-slot t and the battery level at the beginning of time-slot t, respectively. Based on the power harvested history and the modeling method [2], the IoT device may estimate the amount of energy collected at each time-slot. Whereas specifies the location of each service's offloading j—th the energy consumption of each service.

$$e_j = \left[(1 - \delta_j)\, \mathrm{e}_j^{loc} + \delta_j \mathrm{e}_j^{off} - \mathrm{e}_j^{har}, \text{Ъ}_j \right], \tag{7}$$

where δ_j represents the task offloading decision by device j.

A. BATTERY MODEL

The energy derived from the solar panel and stored in the battery during t are represented as $\varepsilon_{rec}(t)$ and $\varepsilon_{bat}(t)$, respectively. Energy consumption and the amount of energy available for services are correlated in two different ways.

$$\begin{aligned} \varepsilon_{bat}(t+1) = min\, (\varepsilon_{bat}(t) + \varepsilon_{rec}(t) \\ - \textstyle\sum_{i \in t} y_j e_j(t), \text{Ъ}_{bat}^{max}\Big), \end{aligned} \tag{8}$$

where Ъ_{bat}^{max} represents the battery's capacity for subchannel to evaluate the service offloading performance and t services waiting are run at a given time interval as indicated by the indicator y_j. However, services cannot be provided using stored battery energy and received solar power when the received energy is insufficient (in the evening) and can be expressed as $\varepsilon_j = \left(y_j e_j(t) + \varepsilon_{bat}(t) - \varepsilon_{rec}(t)\right)$, the battery capacity will be reduced to zero.

III. PROBLEM FORMULATION

To minimize system energy consumption through efficient offloading and enhance the QoS in EH-EC systems, we formulate an optimization problem that focuses on reducing the overall energy consumed by terminal devices.

$$\min_{\delta_j,\partial_j,f_j^{loc}} \sum\nolimits_{j\in\{1,2,\ldots.M\}} e_j, \tag{9}$$

$$0\le f_j^{loc}\le f_j^{max}, \tag{9a}$$

$$0\le \frac{C_j}{r_j^{m}}\left(1-\partial_j\right)t, \tag{9b}$$

$$0\le \mathfrak{e}_j^{loc}\le \mathfrak{e}_j^{har}+\text{Ъ}_j, \tag{9c}$$

$$0\le \mathfrak{e}_j^{off}\le \mathfrak{e}_j^{har}+\text{Ъ}_j, \tag{9d}$$

$$0\le \partial_j\le 1, \tag{9e}$$

$$1\le \text{Ъ}_j\le \text{Ъ}_{bat}^{max}, \tag{9f}$$

$$\delta_j\in\{0,1\}, j\in M. \tag{9g}$$

The CPU computing resource constraint on the device j is indicated by constraint (9a). The completion of the task offloading within the time limit is shown by constraint (9b). Per constraints (9c) and (9d), in the event of local computing and data transfer, the task's energy consumption should not be greater than the total energy gained by wireless charging and the battery's available energy. The time allocation ratio for EH cannot be more than 1, as per constraint (9e). The Ъ_j in (9f) determines the amount of energy harvested t, then check the battery level. The value of the offloading choice is shown by constraint (9g) to be either 0 or 1.

IV. PREDICTED RE AND EC OFFLOADING

In this section, the optimization problem in (9), is proposed to be solved by the REP-DRL. The algorithm uses REP-DRL to achieve the best offloading decision-based DNN. This algorithm combines this decision with momentum gradient descent to optimize the allocation of resources, further lowering the system's energy consumption, and computation offloading with RE generation prediction can be highly beneficial in the IoT. Where IoT devices can extend their battery life and use less energy by shifting computing to distant servers [27]. The complexity of the conventional approach rises as the number of devices increases:

$$\min_{\partial_j,f_j^{loc}} \sum\nolimits_{j\in\{1,2,\ldots.M\}} e_j\left(\delta_j^*\right), \tag{10}$$

where δ_j^* represents the suitable offloading decision.

A. OFFLOADING DECISION-MAKING

The weather and RE are closely correlated, and weather data and RE data are combined to create vectors. In this case, the DNN creates binary offloading decisions to faster up and stabilize convergence. Each DNN layer's output is regularized using stochastic gradient descent when the loss function is minimized to produce the best-fitting effect. This approach accelerates convergence by updating the gradient function based on a single sample can be written as:

$$\varepsilon(t)=\varepsilon_{rec}(t)\uplus\mathfrak{w}(t), \tag{11}$$

where $\mathfrak{w}(t)$ indicates the weather conditions at the time interval t, and $\uplus$ denoted as a series of interconnected. Increasing the amount of RE that is generated at various time frames will help balance generation fluctuations over a wider range of time periods, and compressed air energy storage to help balance fluctuations in RE generation [29], [31]. On account of various temporal connections, the concatenated vectors are divided into three vectors called short-time vector, period vector, and long-term vector as follows:

$$\varepsilon_{sh}(t)=\uplus_{\tau=1}^{\mathcal{T}_{sh}}\varepsilon(t-\tau), \tag{12}$$

$$\varepsilon_{p}(t)=\uplus_{\tau=1}^{\mathcal{T}_{p}}\varepsilon\left(t-\mu_p\tau\right), \tag{13}$$

$$\varepsilon_{l}(t)=\uplus_{\tau=1}^{\mathcal{T}_{l}}\varepsilon(t-\mu_l\tau), \tag{14}$$

where μ_p and μ_l represent the long-term vector and period vector of the platform for the interval between two vectors; each vector's length, or the quantity of actually renewable vectors in each temporal group, is denoted by the letters $\mathcal{T}_{sh}$, $\mathcal{T}_{p}$, and $\mathcal{T}_{l}$, which are all hyperparameters in REP-DRL respectively.

B. LONG SHORT-TERM MEMORY NETWORK

LSTM is a recurrent neural network that is especially well-suited to handle data sequences, including time series. Problems with time series prediction are developed using DNN for LSTM [3]. The two LSTM layers receive the three vectors, and the number of LSTM units per layer can be adjusted to prevent overfitting and decrease training time. The three vectors are used to extract temporal features from two LSTM layers, which are calculated as:

$$f(\tau)=f(\tau).f(\tau-1)+i(\tau).\bar{f}(\tau), \tag{15}$$

$$H(\tau)=\mathrm{O}(\tau).tanh(f(\tau)), \tag{16}$$

$$\varepsilon_{l}(t)=\uplus_{\tau=1}^{\mathcal{T}_{l}}\mathfrak{h}(\tau)\,\ell=1,2, \tag{17}$$

where $C(\tau)$ and $H(\tau)$ denote the cell status and hidden status of the input tensor's $\tau-th$ renewable vector, respectively; and (.) illustrates the multiplication of elements. To get certain attributes from a time series database that has a lot of underlying information and has the potential to improve predicting performance, the DNN component is utilized. The LSTM framework design has four essential components: output stage, forgetting entry, feed gateway, and cell condition. The updating, preservation, and removal of cell updates are governed by the inputs, forgetting, as well as outputs gateways. The update rule for LSTM cell [3] proceeds as follows:

$$C(\tau)=\varphi\left(\mathrm{W}_f.\varepsilon'(\tau)+\text{ƀ}_f\right), \tag{18}$$

$$i(\tau)=\varphi\left(\mathrm{W}_i.\varepsilon'(\tau)+\text{ƀ}_i\right), \tag{19}$$

$$\bar{f}(\tau)=tanh\left(\mathrm{W}_{\mathrm{C}}.\varepsilon'(\tau)+\text{ƀ}_{\mathrm{C}}\right), \tag{20}$$

$$\mathrm{O}(\tau)=\varphi\left(\mathrm{W}_{\mathrm{O}}.\varepsilon'(\tau)+\text{ƀ}_{\mathrm{O}}\right), \tag{21}$$

where W_f, W_i, and W_{O} are the weight matrices of the forgetting gate, input gate, and output gate, respectively; and ƀ_f, ƀ_i, ƀ_{O} are the balance items of the forgetting gate, input gate, and output gate, respectively, φ represent the nonlinear activation perceptron, and tanh represent sigmoid transfer feature. The input to the network typically consists of a vector of values denoting the characteristics of the input data, which are then concatenated with the hidden state of the $\tau - th$ vector [32]. A set of neurons make up the hidden layers $(\tau - 1)$ and apply a set of weights to the input values to generate an output.

$$\varepsilon'(\tau) = \varepsilon_{\ell-1}(t,\tau) \uplus H(\tau - 1). \tag{22}$$

Because there are two connected LSTM layers, the output is $\varepsilon_2(\tau)$. After that, each residual unit is fed into a network that is completely connected to the residual unit and can be written as:

$$\varepsilon_{\ell+2}(t) = \varepsilon_{\ell+1}(t) + r(\varepsilon_{\ell+1}(t)), 1 \le \ell \le L \tag{23}$$

where L represents the number of residual units. The residual function, denoted by the r is a stack of two fully connected layers that has been activated by the ReLU function. In particular, we utilize $\mathrm{L} = 1$ as the sole residual unit in REP-DRL to transform the output into a scalar. Subsequently, this output is passed into another completely connected layer as follows:

$$\varepsilon^{(4)}(t) = \mathrm{W}^4.\varepsilon_i^3(t) + \text{ƀ}^4. \tag{24}$$

The final output of a neural network is typically a vector of values representing the predicted output for each regression target. The W in the output layer is learned during the training process using backpropagation. The error between the predicted output and the actual target value is used to adjust the W to minimize the error. By using a fully-connected layer with a single neuron in the output layer, the REP-DRL can reshape the output of the neural network into a scalar value that can be used for regression tasks. The three distinct vectors' outputs are combined using a weighted sum that is activated by a hyperbolic tangent to provide the estimated value of the amount of renewable generation can be written as:

$$\overline{\varepsilon}(t) = tanh\left(\mathrm{W}_{\mathrm{j}}.\varepsilon_{\mathrm{j}}^4(t) + \mathrm{W}_{\mathrm{p}}.\varepsilon_{\mathrm{p}}^4(t) + \mathrm{W}_{\ell.\varepsilon_\ell^4}(t)\right), \tag{25}$$

where W_j, W_p, and W_ℓ represents the learnable weights. The Mean Squared Error (MSE) loss function is assumed to be the sample's loss function $q = \{1, 2, \ldots .m\}$, and the loss function for all m samples can be written as:

$$loss(\varphi) = \frac{1}{m}\sum\nolimits_{q=1}^{m} \|\overline{\varepsilon}(t) - \varepsilon_{rec}(t)\|_2^2. \tag{26}$$

The most suitable parameters can be established as φ^* to make the loss diminished by using gradient descent. Therefore, $(\mathcal{N}(\varepsilon_j^0(t) + \mathrm{W}_p.\varepsilon_p^0(t) + \mathrm{W}_\ell.\varepsilon_\ell^0(t); \varphi^*)$ can be utilized for predicting the amount of RE in the future.

C. PREDICTIONS FOR RE

By enabling energy to be stored during periods of abundance and utilized later on during times of higher demand. Our aim is to reduce the amount of energy that must be drawn permanently from the conventional power system. This kind of EC issue can be reduced to a MDP, which can handle successfully by DRL [32], [33], [34], [35]. An actor-critic with asynchronous advantage is used to find the best service offloading technique using the anticipated amount of RE and improve energy sustainability for IoT.

1) STATE SPACE

To mitigate potential challenges such as dimension explosion, it is advisable to restrict the agent's observation to the state of a single service at any given time t. Based on this evaluation, the device determines the proportion of data to offload to an ES. According to the EH model shown in (8), the IoT device can estimate the quantity of RE overline $(\varepsilon(t)$ in each time slot t, evaluate its battery's capacity at the moment Ъ_{bat}^{max}, and check the radio link data rates to the M edge devices in the previous t. The anticipated amount of RE is included as a state component, as it guides the management of the service's load by projected future levels of RE, which can be written as:

$$\mathcal{S}(t) = < \overline{\varepsilon}(t), \mathcal{k}, \sigma^2, \mathcal{P}(t), C(t), \text{Ъ}_{bat}^{max}(t), |\mathcal{Q}| > \tag{27}$$

where the Q-value is used to determine the length of $\mathcal{Q}$.

2) ACTION SPACE

Based on state $\mathcal{S}(t)$, the IoT device chooses ES j, and uses the greedy policy with a threshold Ƴ_j of 0 to 1 to prevent remaining in the local maximum. The offloading action can be written as:

$$\mathcal{A}_l(t) = \{0, 1, , \ldots., \text{ȩ}_j^{loc} \le (\text{ȩ}_j^{har} + \text{Ъ}_j)\}, \tag{28}$$

$$\mathcal{A}_o(t) = \{0, 1, , \ldots., \text{ȩ}_j^{off} \le (\text{ȩ}_j^{har} + \text{Ъ}_j)\}, \tag{29}$$

$$\mathcal{A}_b(t) = \left\{0, 1, , \ldots., \text{Ъ}_{bat}^{max}\right\}. \tag{30}$$

When the requirements in (9) cannot be satisfied by merely implementing the offloading action. The task's energy consumption shouldn't be higher than the sum of the energy gained by wireless charging and the battery's available energy in (30), where (28), and (29) represent local processing and data transfer. Additionally, the offloading destination δ_j must be addressed for actions to offload by allocating as many resources as possible for the service.

3) REWARD FUNCTION

Throughout the learning process, the IoT device is motivated by immediate rewards, which are determined by evaluating the benefits of sharing data, along with other factors such as the current battery level, energy consumption, and overall delay. The device receives its reward $\mathcal{R}(t)$ through equations (7) and (25) after receiving the processed service from the ES, it can be written as

$$\mathcal{R}(t) = \varepsilon_{bat}(t) + \varepsilon_{rec}(t) - e_j(t). \tag{31}$$

D. REP-DRL METHOD

REP-DRL used the actor-critic technique to identify the best approach for the MDP. The time-varying decision-making problem is initially modeled using a MDP [14], [36] and the MDP model is subsequently solved using a DRL method [37],

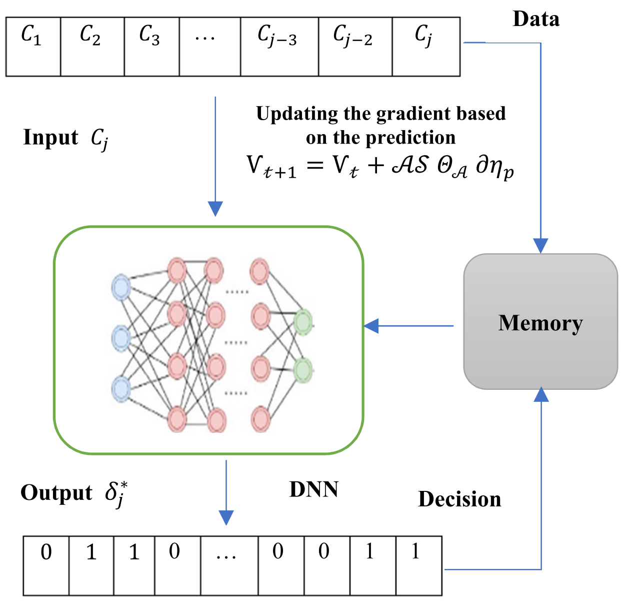
Data
C_1 C_2 C_3 $\cdots$ C_{j-3} C_{j-2} C_j
Input C_j
Updating the gradient based on the prediction
$\mathrm{V}_{t+1} = \mathrm{V}_t + \mathcal{AS}\,\Theta_{\mathcal{A}}\,\partial\eta_p$
Memory
Output δ_j^*
DNN
Decision
0 1 1 0 … 0 0 1 1

FIGURE 2. **Offloading the decision-generation process.**

[38], [39], [40]. The time slot in an MDP is determined by carefully evaluating the objectives of the decision-making process. The actor-critic is made up of an actor-network that decides what to do in a given state and a critical network that calculates the expected cumulative reward perception of a particular state [34], [38], [41].

The strategy in the system is represented by the symbols $\pi(\mathcal{A}_l(t), \mathcal{A}_o(t), \mathcal{A}_b(t)\,|\mathcal{S}(t))$ and refers to the agent's potential to select actions $\mathcal{A}_l(t), \mathcal{A}_o(t)$, and $\mathcal{A}_b(t)$ when the system state is $\mathcal{S}(t)$. The objective of REP-DRL is to identify the best strategy π^*, where the maximum accumulated reward can be expressed as:

$$\mathcal{R} = \sum\nolimits_{t}^{t^{max}} \mu^{t}\mathcal{R}(t). \tag{32}$$

The t^{max} represents the maximum time step, and $\mu \in (0, 1)$ represents the discount factor. The REP-DRL network employs a fully connected to fit the critic, denoted by $\mathrm{V}\left(\mathcal{S}(t); \eta_{\mathrm{V}}\right)$, using the temporal-difference error as a measure. The most effective service offloading technique is determined using asynchronous advantage actor-critic and the predicted amount of RE.

The actor-critic scheme, which increases the processing performance by extending the system state $\mathcal{S}(t)$ to the offloading experience sequence at t, denoted by η_{V}, speeds up learning. Gradient ascent on the actor and critic networks can be used to determine the best offloading approach. The gradient of the objective function of the critic can be written as:

$$d\eta_{\mathrm{V}} = \frac{\partial[\mathcal{R}(t) + \mu\mathrm{V}(\mathcal{S}(t+1)) - \mathrm{V}(\mathcal{S}(t))]^2}{\partial\eta_{\mathrm{V}}}. \tag{33}$$

Concerning the variable V, the actor component can perform admirably on REP-DRL. To do this, the policy is created by the parameter vector V. The actor part $\pi_{\eta_p}(\mathcal{A}_l(t), \mathcal{A}_o(t), \mathcal{A}_b(t)\,|\mathcal{S}(t)))$ is based on the-headed fully connected network, whose value gradient can be expressed as:

$$\partial\eta_p = \mathbb{A}(t)\,\nabla_{\eta_p}\int_{\mathcal{S}}\int_{\mathcal{A}}\left[\log\pi_{\eta_p}(\mathcal{A}_o|\mathcal{S}) + \log\pi_{\eta_p}(\mathcal{A}_l|\mathcal{S}) + \log\pi_{\eta_p}(\mathcal{A}_b|\mathcal{S})\right]d\mathcal{A}d\mathcal{S}. \tag{34}$$

where η_{V} and η_p represent the global actor and critic network, and $\mathbb{A}(t) = \mathcal{R}(t) + \mu\mathrm{V}(\mathcal{S}(t+1)) - \mathrm{V}(\mathcal{S}(t))$ is the advantage function. Getting the best offloading depends on adjusting the gradient for the global actor and critic by summarizing the training status and removing the problem of ongoing parameter updates as:

$$d\mathrm{V}_{t+1} = \mathrm{V}_t + \mathcal{AS}\Theta_{\mathcal{A}}\partial\eta_p, \tag{35}$$

where $\Theta_{\mathcal{A}}$ represents the learning rate of the actor.

Algorithm 1 REP-DRL Based Offloading

```
Input $\Gamma$, $C$, $f$, $\partial$, $\varepsilon(t)$, $\mu$, $\mathcal{B}$, $t$, $C^{max}$, $\text{Ъ}$
output $\pi^*(\mathcal{A}_l(t), \mathcal{A}_o(t), \mathcal{A}_b(t)\,|\mathcal{S}(t)$
initialize $\eta_p$,$\eta_{\mathrm{V}}$
$t \leftarrow 1$
for $t = 1, 2, 3, \ldots\ldots$ do
    Calculate the amount of harvested energy $e_t^{har}$
    Observe the radio transmission rate $\mathcal{B}(t)$, and battery level $\text{Ъ}_{bat}^{max}(t)$.
    $\overline{\varepsilon}(t) \leftarrow (\varepsilon_j^0(t), \varepsilon_p^0(t), \varepsilon_\ell^0(t), \eta)$
    Find the current state $\mathcal{S}(t)$ by $\overline{\varepsilon}(t)$
    Evaluate the current battery level based on calculating the reward in (31)
    Calculate the global actor and critic for $d\eta_{\mathrm{V}}$ and $d\eta_p$ on (33) and (34)
    Update the $\Delta\eta_p$, $\Delta\eta_{\mathrm{V}}$
end for
return $\pi_{\eta_p}(\mathcal{A}_l(t), \mathcal{A}_o(t), \mathcal{A}_b(t)\,|\mathcal{S}(t)))$
```

V. SIMULATION RESULTS

In the upcoming section, we assess the effectiveness of our proposed compute offloading strategies for IoT devices, which are based on the REP-DRL approach, in a dynamic network that includes edge devices. We consider a scenario where an IoT device is tasked with providing real-time emergency treatment through simulations. Two benchmark schemes were evaluated via simulations, namely the DRL scheme from [42], and the Q-learning-based offloading system described in [43]. Figure 3 depicts the impact of the average data size of computational activities on energy consumption across various battery levels and CPU frequencies. Because all activities are executed locally at the highest frequency, the NO scheme, running at maximum CPU frequency, has the highest energy usage. On the other hand, less energy is used by the REP-DRL, which reduces latency by offloading tasks to EC servers.

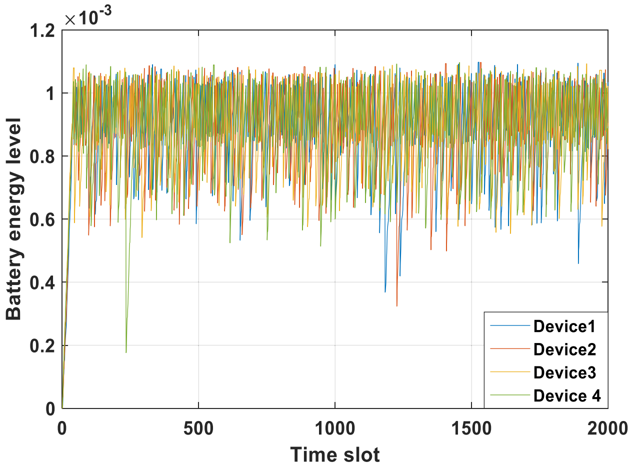

FIGURE 3. Battery energy level versus time slot.

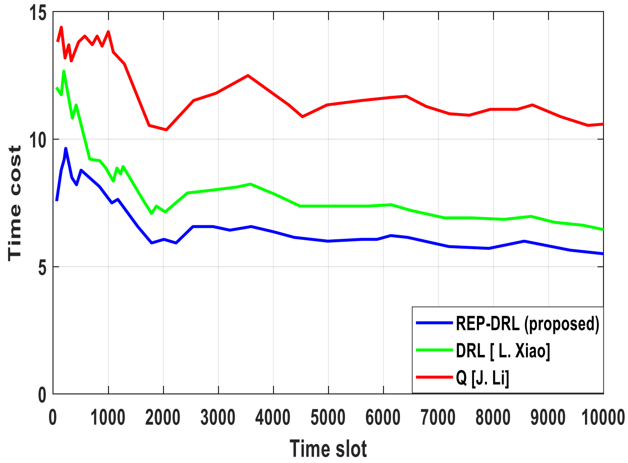

FIGURE 4. Time cost versus time slot.

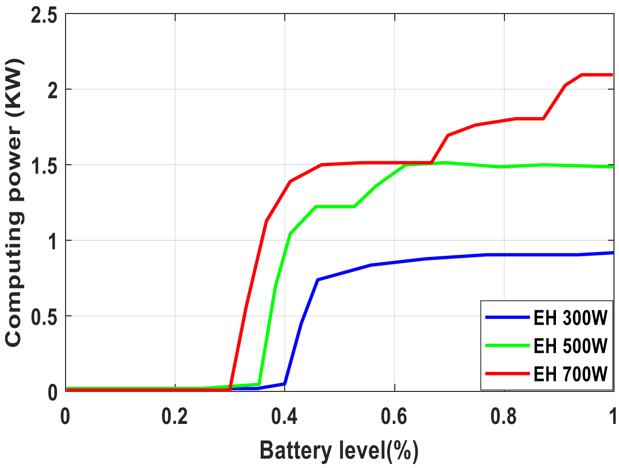

FIGURE 5. Computation power versus battery level.

The NO method shows reduced energy usage limited by latency when it operates at optimal CPU frequency after CPU frequency allocation. Additionally, the technique that has been suggested manages energy usage the least by combining computing RA with power control. Furthermore, by preserving energy, mobile devices with low battery levels can increase their operational hours thanks to the battery level-aware algorithm. The average ratio of jobs offloaded by applying the REP-DRL yields superior cell capacity than when the number of devices grows, as shown in Fig. 3. The average ratio of offloading duties gradually reduces as the maximum distance increases, which explains why. Zero offloading compute jobs will occur when the distance is arbitrarily large. Because of this, there will be longer execution delays and higher energy consumption as the distance between each mobile device and each EC server increases due to the channel power gain.

In Fig. 4, we present a comparison of our scheme's runtime efficiency with that of two benchmark methods over 10,000 time slots and convergence very quickly. As the task requests are directed towards the ES, the REP-DRL method is more appropriate for handling under sampled records. As demonstrated, the cost of the proposed REP-DRL is significantly lower compared to all benchmark methods. The suggested DRL technique [L. Xiao] leads to a high time-average cost because it fails to consider the decision-making process over time, resulting in frequent activation of backup power in later time slots.

By incorporating RE into mobile EC, the authors of [29] focus on improving delay sensitivity at the mobile network's edge. While ignoring the temporal link between system states and decisions, the authors of [29] stress the significance of maximizing the time-average cost. By using all of the battery's energy, they hope to reduce the cost function, which increases to 14 in this time frame. The suggested REP-DRL structure, however, shows that it is possible to cut the time cost to 9.8. This is accomplished by taking particular attributes out of a time series database that contains a wealth of underlying data, which improves the DNN's predicting ability. While the Q-learning method developed by Q[J. Li] is known to have a slow convergence rate due to the large state space, our proposed REP-DRL strategy outperforms it. Despite the decrease in time average cost across 10,000-time slots, our approach demonstrates a significant improvement over the Q-learning method. Figure 5 displays the optimal policy that was learned, providing further evidence for why the suggested approach outperforms the REP-DRL. The policy learned by the proposed method tends to be conservative in its use of local processing resources when the battery level is low and workload demand is modest, and when the network is not congested. This behaviour highlights the effectiveness of the proposed method in adapting to different operating conditions and optimizing resource utilization.

Compared to [29], which is based on an ideal solution, it ignores temporal correlation, which is a myopic approach. To boost the battery level to the ideal power demand of 0.98 KW with a battery level of 0.38 %, it turns on local

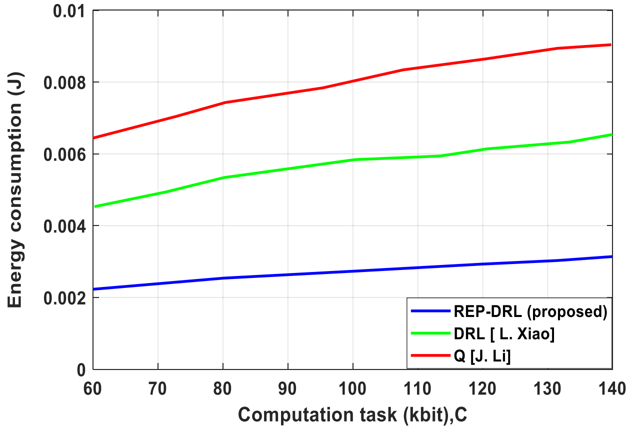

FIGURE 6. Energy consumption versus computation task.

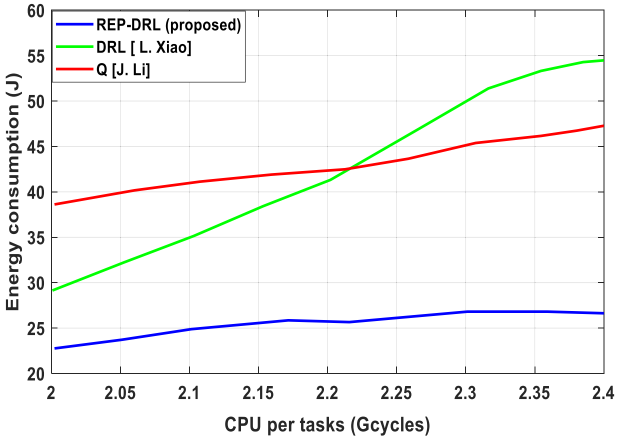

FIGURE 7. Energy consumption versus CPU.

servers to handle the load. However, this manuscript extends the system state, $\mathcal{S}(t)$, to the offloading experience sequence at t, and the REP-DRL architecture reduces processing power while improving processing performance. The LSTM prediction can completely examine the changing pattern of data transfer and has greater data prediction abilities and prediction accuracy. Reducing the processing power to less than 0.97 KW and maintaining a 0.4% battery level expedites the learning step η_V. Nevertheless, this will result in a higher prediction overhead, which will lengthen the LSTM model's training period. Figure 5 demonstrates that the computational power demand policy learned through REP-DRL can be conserved when the environment state is at a low level. The figure indicates that REP-DRL utilizes local computing power more judiciously when the battery level is low and the projected EH is low, highlighting its ability to adapt to changing operating conditions and optimize resource utilization. When the workloads are high, implementing power-saving measures during periods of improved network congestion can result in long-term cost savings for the system. Despite the existence of temporal correlations, the DRL [L. Xiao] algorithm fails to account for them and may activate local servers to handle workloads even when the battery level is low. Figure 6 demonstrates a correlation between the performance of the system and the scope of the computation task C.

The increase in the number of computing tasks in EC leads to a rise in the energy consumption of IoT devices. Migration of computation-intensive jobs to the mobile EC server is one way that mobile EC can increase the computational capability at the edge of wireless networks (J. Li). Furthermore, although there will be a cost and energy cost associated with this, the multi-user mobile EC system can execute computation offloading across wireless channels to a mobile EC server. Even with intermittent and unstable RE sources, the REP-DRL allows renewable-powered IoT energy systems to effectively choose the best offloading strategy, greatly improving EC performance. Despite the number of processing resources and subchannels implemented in the energy system, the REP-DRL exhibits remarkable proficiency in decreasing the supplementary energy demand from the power grid. With an increase in the computing power of the energy system, the REP-DRL approach demonstrates superior performance in comparison to the other two methods. This is illustrated in Fig. 6, where it can be observed that under all conditions, REP-DRL only necessitates an additional energy of 0.002 J to provide services, as opposed to 0.006 J for the benchmark technique and 0.008 J for the Q[J. Li] approaches. For instance, the REP-DRL strategy reduces the energy consumption of the IoT device more than the two techniques, when the computation task size increases from 60 kb to 140 kb. The performance of various task processing techniques in terms of the number of CPU cycles needed for each task is shown in Fig. 7. Based on the data, the CPU cycles per task range from 2 to 2.4 Gcycles, and it is evident that the average energy consumption increases as the number of CPU cycles per task increases. As the CPU cycle per task increases, it requires more energy and time to execute and transmit tasks, which can limit the number of tasks that can be processed simultaneously when there are more subchannels. To create effective mobile offloading solutions, [L. Xiao] analyzes the interconnections among mobile EC offloading. Nonetheless, the attainment of maximum expertise in mobile offloading disposed identifications is hindered by financial delays resulting from computational resources like CPUs, disks, memory, and substantial databases. Nevertheless, by utilizing RE projections to maximize service offloading, REP-DRL can increase the usefulness of IoT devices, boost detection accuracy, and shorten detection time. This methodology expedites the process of arriving at optimal offloading choices while augmenting energy sustainability. However, despite these limitations, the promising results in other situations suggest that the reduced task processing capacity is still acceptable in such scenarios.

From Fig.7, the RED-DRL proposed reduces the energy consumption of the IoT device more than the two techniques, when the CPU cycles increase from 2 to 2.4. Based on the findings presented in references (22) and (23), it can be observed that the REP-DRL proposed achieves the highest computation offloading performance once convergence is reached. When the simulation settings satisfy conditions (28) and (30), the utility of the IoT device approaches the performance bound specified by (34).

VI. CONCLUSION

In this paper, to optimize the utilization of RE, a service offloading technique called REP-DRL has been developed for energy systems enabled by IoT and EC. To forecast RE, a DL-based technique is employed, followed by the use of actor-critic in REP-DRL to determine the optimal service offloading approach for energy systems, leveraging the predicted RE generation information. The use of a REP-DRL approach allows for renewable-powered IoT energy systems to efficiently determine the optimal offloading and autoscaling policies. This is achieved through fast learning, even in the presence of unknown system parameters, by considering the specific problem formulation. From the simulation results REP-DRL optimize computation offloading and greatly increases EC performance even when it is powered by intermittent and unreliable RE. In addition, the REP-DRL saves more energy and lowers the energy consumption of the IoT device, when the CPU cycles increase. Our future efforts will be directed toward enhancing the accuracy of REP-DRL forecasts, reducing service latency, and designing a more precise service offloading mechanism. These endeavors will enable us to effectively apply the technique to real-world energy systems.

REFERENCES

[1] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 13, nos. 2–3, pp. 203–221, Aug. 1996.

[2] H. Wang and J. Huang, "Incentivizing energy trading for interconnected microgrids," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2647–2657, Jul. 2018.

[3] H. Ke, J. Wang, H. Wang, and Y. Ge, "Joint optimization of data offloading and resource allocation with renewable energy aware for IoT devices: A deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 179349–179363, 2019.

[4] S. Aslam, H. Herodotou, S. M. Mohsin, N. Javaid, N. Ashraf, and S. Aslam, "A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids," *Renew. Sustain. Energy Rev.*, vol. 144, Jul. 2021, Art. no. 110992.

[5] Y. Li, Z. Zhou, X. Xue, D. Zhao, and P. C. K. Hung, "Accurate anomaly detection with energy-efficiency in IoT-edge-cloud collaborative networks," *IEEE Internet Things J.*, vol. 10, no. 19, pp. 16959–16974, Sep. 2023.

[6] M. Guo, Q. Li, Z. Peng, X. Liu, and D. Cui, "Energy harvesting computation offloading game towards minimizing delay for mobile edge computing," *Comput. Netw.*, vol. 204, Feb. 2022, Art. no. 108678.

[7] H. Ma, P. Huang, Z. Zhou, X. Zhang, and X. Chen, "GreenEdge: Joint green energy scheduling and dynamic task offloading in multi-tier edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4322–4335, Apr. 2022.

[8] B. Bermejo and C. Juiz, "Improving cloud/edge sustainability through artificial intelligence: A systematic review," *J. Parallel Distrib. Comput.*, vol. 176, pp. 41–54, Jun. 2023.

[9] M. A. Alhartomi, A. Salh, L. Audah, S. Alzahrani, A. Alzahmi, M. R. Altimania, A. Alotaibi, R. Alsulami, and O. Al-Hartomy, "Sustainable resource allocation and reduce latency based on federated-learning-enabled digital twin in IoT devices," *Sensors*, vol. 23, no. 16, p. 7262, Aug. 2023.

[10] Z. Wei, B. Zhao, J. Su, and X. Lu, "Dynamic edge computation offloading for Internet of Things with energy harvesting: A learning method," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4436–4447, Jun. 2019.

[11] Z. Sun, M. Zhao, and M. R. Nakhai, "Computation offloading in energy harvesting powered MEC network," in *Proc. IEEE Int. Conf. Commun.*, Montreal, QC, Canada, Jun. 2021, pp. 1–6.

[12] Q. Tang, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Decentralized computation offloading in IoT fog computing system with energy harvesting: A dec-POMDP approach," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4898–4911, Jun. 2020.

[13] Y. Yang, "Multi-tier computing networks for intelligent IoT," *Nature Electron.*, vol. 2, no. 1, pp. 4–5, Jan. 2019.

[14] Y. Luo, L. Pu, and C.-H. Liu, "CPU frequency scaling optimization in sustainable edge computing," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 2, pp. 194–207, Oct. 2022.

[15] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[16] S. K. U. Zaman, A. I. Jehangiri, T. Maqsood, Z. Ahmad, A. I. Umar, J. Shuja, E. Alanazi, and W. Alasmary, "Mobility-aware computational offloading in mobile edge networks: A survey," *Cluster Comput.*, vol. 24, no. 4, pp. 2735–2756, Dec. 2021.

[17] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7919–7927, Sep. 2020.

[18] X. Zhou, W. Liang, J. She, Z. Yan, and K. I. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6G supported Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5308–5317, Jun. 2021.

[19] F. R. Badal, P. Das, S. K. Sarker, and S. K. Das, "A survey on control issues in renewable energy integration and microgrid," *Protection Control Modern Power Syst.*, vol. 4, no. 1, pp. 1–27, Dec. 2019.

[20] H. Dai, G. Chen, C. Wang, S. Wang, X. Wu, and F. Wu, "Quality of energy provisioning for wireless power transfer," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 527–537, Feb. 2015.

[21] X. Zhou, X. Xu, W. Liang, Z. Zeng, S. Shimizu, L. T. Yang, and Q. Jin, "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1377–1386, Feb. 2022.

[22] P. K. Khatua, V. K. Ramachandaramurthy, P. Kasinathan, J. Y. Yong, J. Pasupuleti, and A. Rajagopalan, "Application and assessment of Internet of Things toward the sustainability of energy systems: Challenges and issues," *Sustain. Cities Soc.*, vol. 53, Feb. 2020, Art. no. 101957.

[23] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, and T. Umer, "Energy-efficient industrial Internet of UAVs for power line inspection in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2705–2714, Jun. 2018.

[24] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[25] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Mobile edge computing and networking for green and low-latency Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 39–45, May 2018.

[26] T. Zhang and W. Chen, "Computation offloading in heterogeneous mobile edge computing with energy harvesting," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 1, pp. 552–565, Mar. 2021.

[27] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.

[28] R. Anwit, P. K. Jana, and A. Tomar, "Sustainable and optimized data collection via mobile edge computing for disjoint wireless sensor networks," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 2, pp. 471–484, Apr. 2022.

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

[30] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[31] E. C. P. Neto, S. Dadkhah, and A. A. Ghorbani, "Sustainable and secure optimization of load distribution in edge computing," in *Proc. IEEE 19th Int. Conf. Smart Communities, Improving Quality Life Using ICT, IoT AI (HONET)*, Marietta, GA, USA, Dec. 2022, pp. 40–45.

[32] A. D. A. B. A. Sofian, H. R. Lim, H. S. H. Munawaroh, Z. Ma, K. W. Chew, and P. L. Show, "Machine learning and the renewable energy revolution: Exploring solar and wind energy solutions for a sustainable future including innovations in energy storage," *Sustain. Develop.*, vol. 32, pp. 1–26, Jan. 2024.

[33] K. Mishra, G. N. V. Rajareddy, U. Ghugar, G. S. Chhabra, and A. H. Gandomi, "A collaborative computation and offloading for compute-intensive and latency-sensitive dependency-aware tasks in dew-enabled vehicular fog computing: A federated deep Q-Learning approach," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 4, pp. 4600–4614, Sep. 2023.

[34] Y. Zhang, Y. Miao, S. Pan, and S. Chen, "Efficient and energy-saving computation offloading mechanism with energy harvesting for IoT," *Secur. Commun. Netw.*, vol. 2021, pp. 1–10, Dec. 2021.

[35] F. Wang, H. Xing, and J. Xu, "Real-time resource allocation for wireless powered multiuser mobile edge computing with energy and task causality," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 7140–7155, Nov. 2020.

[36] Z. Fang, X. Xu, M. Bilal, and A. Jolfaei, "Renewable prediction-driven service offloading for IoT-enabled energy systems with edge computing," *Wireless Netw.*, vol. 27, no. 6, pp. 1–13, Aug. 2021.

[37] X. Ding, Y. Zhang, T. Liu, and J. Dua, "Deep learning for event-driven stock prediction," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2327–2333.

[38] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[39] C. Li, Y. Hu, L. Liu, J. Gu, M. Song, X. Liang, J. Yuan, and T. Li, "Towards sustainable in-situ server systems in the big data era," *ACM SIGARCH Comput. Archit. News*, vol. 43, no. 3S, pp. 14–26, Jan. 2016.

[40] G. Wu, H. Wang, H. Zhang, Y. Zhao, S. Yu, and S. Shen, "Computation offloading method using stochastic games for software defined network-based multi-agent mobile edge computing," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 17620–17634, Jun. 2023.

[41] V. Mnih, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, Feb. 2016, pp. 1928–1937.

[42] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for MEC," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Barcelona, Spain, Apr. 2018, pp. 1–6.

[43] L. Xiao, Y. Li, X. Huang, and X. Du, "Cloud-based malware detection game for mobile devices with offloading," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2742–2750, Oct. 2017.

MOHAMMED ALHARTOMI (Member, IEEE) received the B.Sc. degree from King Abdulaziz University, Jeddah, Saudi Arabia, the M.Sc. degree from Swansea University, Swansea, U.K., with a focus on wireless communications systems, and the Ph.D. degree from the University of Leeds, Leeds, U.K., in 2016. He is currently an Associate Professor with the Electrical Engineering Department, University of Tabuk, Saudi Arabia. He is also a Visiting Scholar with the University of Central Florida, working on dense VLC networks. His research interests include mobile and wireless communications, optical communications, and signal processing.

ADEB SALH received the bachelor's degree in electrical and electronic engineering from IBB University, Yemen, in 2007, and the master's and Ph.D. degrees in electrical and electronic engineering from University Tun Hussein Onn Malaysia, in 2015 and 2020, respectively. From 2007 to 2012, he was a Lecturer Assistant with the Yareem Community College. From 2020 to 2023, he was a Postdoctoral Researcher with UTHM and UTM. Currently, he is an Assistant Professor with the Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman. His research interests include 5G, 6G wireless communications, massive MIMO, artificial intelligence (AI), and the Internet of Things (IoT).

LUKMAN AUDAH (Member, IEEE) received the Bachelor of Engineering degree in telecommunications from Universiti Teknologi Malaysia, in 2005, and the M.Sc. degree in communication networks and software and the Ph.D. degree in electronic engineering from the University of Surrey, U.K. He is currently a Senior Lecturer with the Communication Engineering Department, Universiti Tun Hussein Onn Malaysia. His research interests include wireless and mobile communications, internet traffic engineering, network system management, data security, and satellite communications.

SAEED ALZAHRANI (Member, IEEE) received the B.S. degree in electrical engineering from King Abdulaziz University, Jeddah, Saudi Arabia, the M.S. degree in electrical engineering from the University of Colorado at Colorado Springs, and the Ph.D. degree in electrical engineering from The Ohio State University. He was with the Microelectronics Research Laboratory, University of Colorado at Colorado Springs, from 2012 to 2014, involved in developing tunable ferroelectric-based filters, VCOs, amplifiers, and antennas. From 2014 to 2019, he was with the Electro Science Laboratory, The Ohio State University, focusing on developing design techniques for wide TR VCOs at the mm-wave frequency band. He is currently an Assistant Professor with the Electrical Engineering Department, University of Tabuk, Saudi Arabia. His research interests include addressing design and technological challenges related to RF/millimeter-wave and mixed-signal integrated circuits and systems for emerging technologies, including communication, automotive, and biomedical applications.

AHMED ALZAHMI received the M.S. degree in computer engineering from Florida Atlantic University (FAU), Boca Raton, FL, USA, in 2010, and the Ph.D. degree from Southern Methodist University, Dallas, TX, USA, in 2018. He is currently an Assistant Professor with the Faculty of Engineering, University of Tabuk, Saudi Arabia. His current research interests include 3D/2D memory interface design, RF transceiver design, PDN, and energy harvesting circuit designs.