



# Article Energy-Efficient Offloading Based on Efficient Cognitive Energy Management Scheme in Edge Computing Device with Energy Optimization

Vishnu Kumar Kaliappan <sup>1</sup>, Aravind Babu Lalpet Ranganathan <sup>2</sup>, Selvaraju Periasamy <sup>3</sup>, Padmapriya Thirumalai <sup>4</sup>, Tuan Anh Nguyen <sup>1</sup>, Sangwoo Jeon <sup>5</sup>, Dugki Min <sup>5,\*</sup> and Enumi Choi <sup>6</sup>

- <sup>1</sup> Konkuk Aerospace Design Airworthiness Institute, Koknkuk University, Seoul 05029, Korea
- <sup>2</sup> Department of Computer and Information Science, Annamalai University, Chidambaram 608002, India
- <sup>3</sup> Department of Mathematics, Rajalakshmi Institute of Technology, Chennai 600124, India
- <sup>4</sup> Melange Academic Research Associates, Puducherry 605004, India
- <sup>5</sup> Department of Computer Science and Engineering, Konkuk University, Seoul 05029, Korea
- <sup>6</sup> Department of Computer Science and Engineering, Kookmin University, Seoul 05029, Korea
- \* Correspondence: dkmin@konkuk.ac.kr

Abstract: Edge devices and their associated computing techniques require energy efficiency to improve sustainability over time. The operating edge devices are timed to swap between different states to achieve stabilized energy efficiency. This article introduces a Cognitive Energy Management Scheme (CEMS) by considering the offloading and computational states for energy efficacy. The proposed scheme employs state learning for swapping the computing intervals for scheduling or offloading depending on the load. The edge devices are distributed at the time of scheduling and organized for first come, first serve for offloading features. In state learning, the reward is allocated for successful scheduling over offloading to prevent device exhaustion. The computation is therefore swapped for energy-reserved scheduling or offloading based on the previous computed reward. This cognitive management induces device allocation based on energy availability and computing time to prevent energy convergence. Cognitive management is limited in recent works due to non-linear swapping and missing features. The proposed CEMS addresses this issue through precise scheduling and earlier device exhaustion identification. The convergence issue is addressed using rewards assigned to post the state transitions. In the transition process, multiple device energy levels are considered. This consideration prevents early detection of exhaustive devices, unlike conventional wireless networks. The proposed scheme's performance is compared using the metrics computing rate and time, energy efficacy, offloading ratio, and scheduling failures. The experimental results show that this scheme improves the computing rate and energy efficacy by 7.2% and 9.32%, respectively, for the varying edge devices. It reduces the offloading ratio, scheduling failures, and computing time by 14.97%, 7.27%, and 14.48%, respectively.

Keywords: edge computing; energy efficiency; reward function; state learning

## 1. Introduction

An edge computing device is a type of hardware that drives edge computing applications in various industries. Edge computing devices are mainly used to accomplish specific tasks given by software applications [1]. The primary purpose of an edge computing device is to manage the application closer, which prevents unwanted data loss [2]. Edge devices are also used to control data flow that occurs among functions and operations. Energy efficiency is the level of energy that is used to perform a task. The energy efficiency level is an important variable in every application and system. Edge computing devices mostly reduce the overall energy consumption rate in a network, which enhances the efficiency of the system [3]. Edge computing devices provide better privacy-preserving policies that



Citation: Kaliappan, V.K.; Lalpet Ranganathan, A.B.; Periasamy, S.; Thirumalai, P.; Nguyen, T.A.; Jeon, S.; Min, D.; Choi, E. Energy-Efficient Offloading Based on Efficient Cognitive Energy Management Scheme in Edge Computing Device with Energy Optimization. *Energies* 2022, *15*, 8273. https://doi.org/ 10.3390/en15218273

Academic Editors: R. Maheswar, M. Kathirvelu and K. Mohanasundaram

Received: 14 September 2022 Accepted: 2 November 2022 Published: 5 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). reduce energy consumption rates in the authentication process. A key agreement scheme is mostly used in edge computing devices that identify the actual users of devices [4]. A key agreement or value reduces the time consumption rate in the identification process, which improves the effectiveness and reliability of an application. Energy efficiency levels in edge computing devices are high, which reduces the error rate and latency rate in providing services for the users [5].

An edge network is a place where local networks and devices are interfaced with a network connection. An internet connection plays a major role in edge networks. Energy efficiency analysis is performed in the edge network [6]. The efficiency analysis process analyzes the datasets that are available in the database, and provides the necessary set of data for the edge network to perform particular tasks. The energy efficiency analysis process also analyzes the content and time needed to perform a certain task in a network [7]. The energy efficiency analysis process reduces the overall energy consumption rate in the computation process, enhancing the edge network's sustainability and reliability. The energy efficiency analysis process provides an appropriate set of data to provide accurate services for the users [8]. The efficiency analysis process reduces the latency rate in the computation process and improves the speed level of the process. Energy efficiency computing creates a safe path to obtain datasets that reduce the latency rate in the searching and identification process. The computing system enhances the quality of service and information in the edge network. Energy efficiency computing is mainly used in edge networks to improve the privacy and security of users' data and protect them from attackers [9,10].

Machine learning (ML) techniques are primarily used in various fields for prediction, recognition, and detection. ML techniques improve the overall accuracy rate in the prediction process, providing an accurate dataset for multiple functions. ML techniques are used in edge devices to reduce the computation cost and latency rate in the computation process [11]. Edge networks are mostly used in industries and smart devices to provide proper services for users. Convolutional neural network (CNN)-based wireless sensors are used in edge devices [12]. CNN reduces the time consumption rate in the identification process and improves the accuracy rate in the edge node detection process [13]. CNN also increases edge devices' energy efficiency rate, enhancing device efficiency. The deep reinforcement learning (DRL) algorithm is also used for energy efficiency-based edge devices [14]. DRL predicts the exact resources required to perform tasks that reduce the energy consumption rate in the allocation and classification process. DRL improves edge devices' performance and sustainability rate, providing necessary data for further processes in an application and system [15]. This study makes the following significant contributions:

- Designing a cognitive energy management scheme for edge devices by assimilating the machine learning paradigm.
- Introducing a joint scheduling and offloading process for preventing energy convergence across distributed edge devices.
- Performing a comparative analysis study for verifying the proposed scheme's performance with precise metrics and existing methods.

The paper's concept is designed for managing energy efficacy of edge devices through precise selection of scheduling and offloading concepts. The energy efficacy is predominant over increasing the user/application density. After the energy harvesting techniques, fundamental conservation is required for balancing the edge device operations over prolonged time.

The rest of the paper is structured as follows: Section 2 expands on the findings of related work on AI-based approaches for energy management schemes. Section 3 describes the proposed machine learning-based cognitive energy management scheme. Section 4 discusses implementation and results analysis, while Section 5 delves into the conclusion and future directions.

# 2. Related Works

Ali et al. [16] introduced a deep learning (DL)-based resource allocation approach for Mobile Edge Computing (MEC) systems. The power Migration Expand (powMigExpand) algorithm is used here for the resource allocation process that identifies the critical set of data. The powMigExpand algorithm provides the necessary resources for the allocation process by analyzing the exact device requirement. The proposed method achieves a high accuracy in the allocation process that increases the quality of service (QoS) rate of the MEC system.

Ale et al. [17] proposed a deep reinforcement learning (DRL) algorithm-based energyefficiency computation offloading method for MEC systems. DRL is mainly used here to determine the exact need and requirements for the computation process. DRL identifies the exact edge nodes and servers for offloading and resource allocation. The proposed method reduces the computation process's time and energy consumption rates. The proposed approach also improves the performance and reliability of the system.

Irtija et al. [18] designed an energy-efficient edge computing system for multi-access edge computing. The proposed method is also used for a fully autonomous aerial system (FAAS) by using a deep neural network (DNN). The DNN-based approach identifies the area of interest (AOI) that provides the necessary set of data for the satisfaction process. From the AOI, the energy information is observed for leveraging deployment. These deployment issues are addressed by projecting energy-efficient devices across the AOI. Therefore, the energy utilization and improvements are linear for the available computing through multiple accesses (V1). The proposed method achieved a high accuracy rate, enhancing the efficiency and sustainability of the MEC and FASS systems.

Lu et al. [19] introduced a scheduling algorithm named the energy-aware doublefitness particle swarm optimization (EA-DFPSO) method for MEC. The PSO algorithm identifies both computation and edge nodes, providing an optimal dataset for the computation process. PSO reduces the energy consumption rate in the computation process, enhancing the MEC reliability and stability. The proposed EA-DEPSO method reduces the latency rate in the computation process and improves the energy efficiency rate of the MEC system.

Wen et al. [20] developed a cluster-based wireless sensor network (WSN) for edge computing. The main aim of the proposed method is to develop an energy-efficient task allocation process. The genetic algorithm (GA) algorithm is used here to identify the requirements necessary for the load balancing process. The proposed WSN method reduces the computation process's energy and time consumption rate, enhancing the edge network's load balancing level.

Xie et al. [21] introduced a minimal retention energy harvesting (MREH) method for edge devices. The proposed method is mainly used for Internet of Things (IoT)-based devices and applications. The MREH method focuses on swapping edge nodes that are needed for energy harvesting. MREH identifies the edge nodes and provides a possible dataset for the computation process. This supplied data equips multiple observation sequences for identifying the swapping instances. Considering the available edge nodes, IoT computations are required for enhancing the EH. The proposed MREH method improves energy efficiency.

Dai et al. [22] proposed a deep reinforcement learning (DRL) algorithm-based partition approach for an edge computing system. Game theory and Deep Neural Network (DNN) approaches are used here to identify the resource required for resource allocation. DNN improves the accuracy rate in partitioning, where it provides possible services for users. The Dai approach reduces the computation processes, time, and energy consumption. Added to it, DRL improves the scalability and efficiency of an edge computing system.

Zhang et al. [23] developed a dynamic programming-based energy-saving offloading (DPESO) for the mobile edge computing offloading (MECO) system. Identifying an offloading decision problem is a complicated task to perform in an edge computing system. The proposed method is mainly used to determine the offloading decision problem presented in a computing system. DPESO minimizes the latency rate in the computation process, improving the MECO system's effectiveness and scalability. The proposed DPESO increases the energy efficiency rate in an edge computing system.

Alsubhi et al. [24] designed a Mobile Energy Augmentation for smart devices using Cloud Computing (MEACC). MEACC is used to find the inefficient energy consumption source and produce a proper dataset for further process. Smart devices utilize more energy to perform specific tasks. MEACC reduces the communication cost and latency rate in the offloading process. The proposed MEACC method reduces the overall energy consumption rate in the computation process, providing a better load balancing level for an edge computing system.

Li et al. [25] introduced a multi-edge collaborative computation offloading strategy for MEC systems. The proposed method calculates the execution time to perform a specific task in the MEC system. A migration strategy is used here that analyzes the datasets that are required for the computation process. The mitigation strategy identifies the previous state utilization and energy drains for preventing failures. The computations are restricted to the energy availability of the devices through multiple remaining energy metrics. Due to the overheads in migration, the overloading tasks are confined using MEC offloading. The proposed method reduces the computation energy demands.

Zhou et al. [26] proposed an edge intelligent energy-efficient model (ECMS) for MEC systems. When compared with other methods, the proposed method improves the performance and feasibility of the MEC system. The Elman neural network (ENN) algorithm is used here that identifies the energy consumption rate to perform tasks in MEC. ENN predicts accurate energy consumption. This energy consumption feature is estimated across different intervals where the communication is either consistent or varying, provided the energy exhaustions are identified.

Liu et al. [27] introduced an energy-aware allocation method for MEC systems. An MEC server identifies the access points (AP) presented in MEC that provide necessary data for the offloading process. The proposed method reduces the complexity and time consumption rate in the computation process. The complexity of the varying multiple access switchovers is addressed using offloading procedures. This process reduces the computations over different intervals to prevent energy drops. The proposed method provides high-quality services for users and reduces the energy consumption rate in the computation process.

Xie et al. [28] proposed an energy-efficient collaborative computation method for MEC networks. The block coordinate descent (BCD) method is used to find the coordinates available in the MEC network. The proposed method is used primarily for reconfigurable intelligent surface (RIS)-assisted MEC networks. In the reconfiguration process, the remaining energy-based allocations are provided across multiple coordinates, preventing MEC delegations. Therefore, energy utilization is reformed in order to avoid various device utilizations. In this process, the variations are provided using available devices. The proposed method improves the energy efficiency level by reducing the energy consumption rate in the computation process.

The methods discussed above rely on assisted networks as in [20,24,28] for energy conservation by incorporating the conventional WSN strategies. Independent processes such as in [18,19,23] provide optimization alongside learning paradigms that increase the complexity during iterated training. The proposed energy management scheme steers between the offloading and scheduling decisions using different state models. In particular, the state model is incorporated into this work due to its action and reinforcement strategies. The action represents the allocation, scheduling, and offloading computed using the available energy and drain. The transitions between the states are required for limited intervals, i.e., before offloading and scheduling. Therefore, recurrent training machine learning is less required for this energy management scheme.

The aforementioned methods and techniques optimize the energy efficiency through harvesting, conservation, and device changeovers. These methods are conventional for improving the current transmissions across multiple shared devices in an edge computing scenario. However, the knowledge of service demands is unknown for the varying intervals, due to which the allocation and prolonged device operations are mandatory. As the energy drain is accepted, the convergence, device failure, or recharging intervals are frequent. Therefore, the device and task swapping instances are confined through pre-exhaustion identification and proper device selection. Considering the energy levels and tasks, the balancing takes place with considerable learning procedures in this proposed scheme. This prevents scheduling failures due to periodic device switching based on energy complexities.

#### 3. Proposed Energy Management Scheme

The edge computing device-based energy efficiency processes specific tasks that different users and software applications observe. The edge computing device model maintains the application closer and requires data storage and a computation process to perform a certain task and enhance the system's efficiency. The challenges in operating the edge devices, such as scheduling, offloading, and allocation, are considered factors for improving the energy efficacy of edge computing to satisfy user needs and demands. The edge computing devices are highly competitive in achieving stabilized energy efficiency between different states considering their energy efficiency. The edge devices and their associated computing are used in many industries for improving energy conservation and distribution, along with allocating time for scheduling and offloading based on the sustainability of the proposed scheme to improve energy availability. However, addressing energy convergence at the time of cognitive management generates the edge device allocation that relies on carbon emissions, deforestation, pollution, and enormous energy consumption. These impacts augment concerns for swapping the edge computing time intervals for offloading or scheduling depending on the load needed to reduce failures. Offloading is one of the solutions used to identify the offloading features when scheduling edge devices through state learning. Offloading is identified in edge computing and requires diverse performances to reallocate the offloading features to save the minimum energy. An enormous amount of user needs and demands are available in energy management, which is feasible in scheduling failures, and computation of device exhaustion is a paramount consideration in edge computing. Figure 1 portrays the proposed CEMS in an edge environment.



Figure 1. Proposed CEMS in edge environment.

The proposed energy management scheme mainly focuses on consideration by identifying rewards for all successful scheduling over the offloading in edge computing through Q-learning. In this scheme, edge computing is administered to control the data flow when performing functions and operations. The proposed scheme reduces the edge computing latency rate and augments the process's speed (refer to Figure 1). Energy efficiency computing generates a safe path to obtain datasets from edge devices. It prevents latency in identifying and searching processes. In this proposed scheme, the edge computing systems increase the quality of information, and services in the edge network are analyzed to determine any existing edge computing modifications to augment the privacy and security of users' datasets from attackers. Machine learning is a combination of performing the prediction analysis that requires a set of data for various operations in edge computing. State learning is used in this proposed scheme to identify the accurate edge nodes that are aided in performing tasks. Learning reduces the time consumption in the identification process and increases the overall accuracy of the edge node detection process. The energy availability is classified as conservation, and distribution depends on available datasets. In this proposed scheme, the edge computing devices reduce the energy consumption in an edge network, which enhances the sustainability and reliability of the edge network.

The edge devices and their associated techniques improve the speed level of the edge computing process depending on the user's demands through state learning. The energy efficiency analysis is based on the learning process content and computing time to perform a certain task in an edge network. The offloading and computational states for energy efficacy consideration are made. The energy efficiency analysis is based on the edge devices  $(E_d)$  in that network. Therefore, the scheduling or offloading based on the load is designed into three segments: scheduling, offloading, and allocating content and time to perform a task. The energy management scheme varies based on users' needs or demands to handle energy availability in that edge network. The initial function of edge device computing is keen on maintaining the quality of service and information, relying on the objective as in Equation (1).

$$\begin{array}{c} \underset{N \in tk}{maximize} E_{d}(c) \forall Scl = Ofl = Alloc \\ and \\ maximize D_{L} \forall (\frac{Scl}{Ofl}) \\ N \in E_{d}(c) \end{array} \right\}$$
(1)

where:

$$D_L = E_d(c)(C_T - Id_P]$$
<sup>(2)</sup>

$$D_L = \underset{n \in i}{minimize} E_d(c)_{tk} \ \forall \ N \in Scl_{tk}$$
(3)

As per Equations (1)–(3), the variables  $E_d(c)$ , Scl, Ofl, and Alloc represent the edge computing process of performing N tasks depending on scheduling, offloading, and allocation, respectively. In the following next edge device computing, the variables  $D_L$ ,  $C_T$ , and  $Id_P$  are used to denote the unwanted data loss, time consumption rate, and identification process, respectively. The third objective is to minimize the data flow that occurs in functions and operations using the  $E_d(C)_{tk} \forall N \in Scl_{tk}$  condition. If  $U^S = (1, 2, ..., u^s)$ represents the set of users in edge computing devices, the overall energy consumption in the computation process that enhances the reliability and sustainability of the edge network is based on  $E_d(c) \times C_T$ , whereas an appropriate set of data provides service for the users of  $u^s \times E_d(c)$ . The overall energy efficiency analysis is based on  $u^s \times E_d(c)$  and  $E_d(c) \times C_T$ for energy availability. The control of data flow and offloading is used to perform a certain task. The scheduling of offloading relies on the load to provide better privacy-preserving policies that reduce energy consumption in the authentication process. In this edge computing, energy efficiency analysis is essential to identify data flow, time slots, and devices in that network. The user needs or demands are based on improving the sustainability  $(st_b)$ of performing N tasks. The remaining energy consumption time for scheduling and organizing for FCFS for available offloading features relies on energy efficacy for improving device exhaustion. The cognitive management for the available N tasks is performed using machine learning. Later, the edge devices are distributed at the time of scheduling; the edge computing analysis is the improving factor to provide accurate services for the users. From this scheduling or offloading based on the edge, devices are the prevailing instance

for the required computation process. The reliability and sustainability level of the system predicting the energy consumption rate for considering the energy-aware allocation are essential in the following section. The offloading process is illustrated in Figure 2.



Figure 2. Offloading illustration.

The user tasks (requests) are streamlined from  $C_T$  for identifying scheduling probability. If the allocation does not fit the identified  $C_T$  (i.e.,)  $\frac{N}{C_T} > Alloc$ , then the offloading probability is high. Therefore, an edge device (s) is identified as satisfying energy management conditions for offloading. In the offloading process, the energy conditions defined in Equations (1) and (2) are to be satisfied to prevent  $D_L$  (refer to Figure 2). The identification process in edge device computing for performing a particular task, which relies on the state learning of  $(E_d(c) \times C_T)$  and is computed for improving energy consumption for all N tasks on the basis of sustainability overtime, is the considering factor. The probability of offloading  $(\rho_{Of})$  in an edge device computation is given as:

$$\rho_{Of} = \left(1 - \rho_{E_d(c)}\right)^{C_T - 1}, \ N \in C_T$$
(4)

where:

$$\rho_{E_d(c)} = \left(1 - \frac{E_d(c) \in N}{E_d(c) \in C_T}\right) \tag{5}$$

From Equations (4) and (5), the continuous energy consumption in edge computing relies on the offloading and computational states of *N* tasks. Therefore, the remaining tasks are performed to swap between different states; hence, the scheduling time is substituted as in Equation (1). Therefore, the offloading computation for  $\rho_{E_d(c)}$  follows:

$$Of(N(tk)] = \frac{1}{(Scl + Ofl - Alloc|} \cdot \left(\rho_{E_d(c)}\right)_{C_T}, N \in C_T$$
(6)

In Equation (6), the edge computing for the *N* load depending on the scheduling or offloading as in Equation (6) is to satisfy both the condition of  $u^s \times E_d(c)$  and  $E_d(c) \times C_T$ , improving the energy efficiency of edge devices. The offloading process in edge computing is processed using state learning to assign different states for time consumption to reduce the impact of the data flow and reallocation for generating minimum energy based on  $(u^s \times E_d(c)) > (E_d(c) \times C_T)$ , and the computational states for energy efficacy descriptively use machine learning. Therefore, the successful scheduling over the offloading, which follows  $u^s > C_T$  and  $\rho_{E_d(c)}$  for minimal energy consumption, is to satisfy Equation (1). The various states depend on  $\rho_{E_d(c)}$  and hence the edge computing, resulting in an offloading process for reallocation.

In an edge device computing scenario, the data flow and identification process are performed for the condition  $u^s \times E_d(c)$  to maximize energy availability, and the time consumption rate and scheduling failures are invariant. The maximum and minimum

scheduling in edge computing identify the offloading along with the reward allocated through Q-learning of *N* tasks, and the reward allocation is the considering factor here. The probability of reward allocation ( $\rho_{Rw}$ ) is computed as:

$$\rho_{Rw} = \frac{\rho_{Of}. D_L(N(tk)].\left((Scl - Ofl) * \rho_{E_d(c)} - \left(\frac{Scl - Ofl}{N}\right)\right)}{F(S_L).N}$$
(7)

where:

$$F(Q_L) = \sum_{N=1} \left( \frac{(Scl - Ofl) * \rho_{E_d(c)} * \rho_{D_L}}{D_L(N)} \right)$$
(8)

Based on Equations (7) and (8), the variable  $F(Q_L)$  represents the function of Q-learning at different time intervals. For all of the edge device computing processes, the sustainability over time of energy availability is analyzed for N tasks required for allocating the reward. As in Equation (1), the data flow identification requires more energy efficiency. The state models with reward allocations are illustrated in Figure 3.



Figure 3. State models with reward allocation.

The Q-learning states are defined for "Alloc", "Ofl", and "Scl" based on different conditions satisfying Equations (1) and (2). This state modeling is distinguished as devicebased and allocation-based. The *RW* is estimated for  $\rho_{Of} = 1$  and  $Id_p = True$  conditions is the device-based one; the N(tk) and  $c \neq 0$  validations are performed in the allocationbased method. The  $\rho_{RW}$  is used for computing the feasibility that prevents  $D_L$  and high  $C_T$ . This state modeling is induced for  $U^s$  until resource allocation is made (Figure 3). The continuous edge computing analysis, the energy conservation, and distribution outcomes depend on identifying the minimum or maximum time consumption for swapping the computing intervals for scheduling or offloading of  $u^s > C_T$ , and *N* task performance and computing time are the considered metrics. These metrics are addressable using Q-learning and energy management to mitigate the data flows through reward allocation. The decision to allocate rewards to scheduling relies on FCFS for offloading features. The following section represents the energy management scheme for edge computing to reduce edge devices' data flows and time consumption.

Energy Management: Energy availability is classified as conservation and distribution based on performing certain tasks. This scheme is used to control the economy evaluation time for both sequential and individual factors. The energy efficiency analysis is computed to identify the data flows and reallocates tasks in edge devices using machine learning. The edge computing process relies on energy management to reduce the data flows and time consumption when performing a task. Therefore, the condition for energy availability is different for each task in edge devices that follows individual computation processes for enhancing the sustainability of the edge device. The learning process is used for computing the time consumption rate for the N tasks and available offloading features. The first energy availability relies on maximum edge device computing  $(E_d(c)_{ea})$  and  $F(Q_L)$  is computed as:

$$F(Q_L, E_d(c)_{ea}) = \left( \left( \frac{\rho_{E_d(c)}}{\rho_{e_c} + \rho_{e_d}} \right) \times \frac{1}{n} \right] - D_L(n) + 1$$
(9)

In Equation (9), the variables ea,  $e_c$ , and  $e_d$  represent the energy availability, energy conservation, and energy distribution in edge computing, respectively, depending on energy efficiency analysis as in  $\rho_{E_d(c)}$  and  $D_L(n)$  for swapping the states between edge devices. The state learning and reward estimation for ea, ec, and ed are illustrated in Figure 4.



Figure 4. State learning and reward estimation for ea, ec, and ed.

The energy management processes for state modeling and  $\rho_{RW}$  are different from that of the device-based process. This model limits the "*Alloc*" based on the *RW* and *ec* phases. For the *ea* model,  $E_d(c)$  is alone validated for  $\rho_{ED}(c) = max$ ; this induces offloading fewer computations. In the *ed* phase, if c = 0, then *RW* for "*Ofl*" is estimated  $\forall 0 \rho_{ED}(c) < 1$ such that "*Scl*" is temporarily halted. Considering the *ec*, the ea is determined by identifying N(tk) and *RW* for "*Scl*" to "*Ofl*", preventing any new "*Alloc*" (refer to Figure 4). Here, the chances of device exhaustion through previous reward achieving continuous energy distribution is computed as:

$$\rho_{e_d} = \frac{1}{\sqrt{2N}} expension\left(\frac{Scl - \rho_{E_d(c)}}{N}\right) \tag{10}$$

In Equation (10), the probability of edge computing, the objective is to balance users and time is to minimize the data flow; hence, the actual energy distribution in that edge network is computed as:

$$Scl = max \left(\frac{\rho_{E_d(c)}}{D_L(N) - \rho_{e_c}}\right)$$
(11)

Therefore, the energy distribution in the edge device is validated as  $\left(1 - \left(\frac{\rho_{E_d(c)}}{D_L(N) - \rho_{e_c}}\right)\right)$  and the time consumption of the energy management process in the device allocation instance depends on the scheduling process. The exceeding time slot and edge devices require different states and hence the energy availability is demandingly improved. The energy availability, conservation, and distribution probability are considered in edge computing. The offloading occurred in the edge network for the condition (*Scl*, *Ofl*) is differentiated based on  $\rho_{e_d}$  for  $F(Q_L)$ , and is given as:

$$D_L(N) = \begin{cases} \frac{N - (\rho_{ESG} * E_d)}{n + (\rho_{R_f})} \forall Scl = E_d(c) \\ \frac{N - \rho_{E_d(c)}}{N + (\rho_{e_d} + \rho_{e_c} - \rho_{D_L})} \forall Scl < E_d(c) \end{cases}$$
(12)

In Equation (12), the edge computing process of  $(\rho_{e_d} + \rho_{e_c} - \rho_{D_L})$  is the idle probability for energy conservation, and energy distribution is performed based on the edge network using state learning through  $D_L(N)$  analysis. Finally, the differentiation scheduling is presented in Figure 5.



Figure 5. Differentiation scheduling.

The  $F(Q_L)$  generates *RW* for device- and energy-based allocations. These allocations are discussed in Figures 3 and 4; the energy constraints determine the *RW* for different state models post the *ec* phase. Therefore, leaving out *ec* (due to no "*Alloc*"), *Scl*  $\in$  *ea* and *Scl*  $\in$  *ed* are performed. It is to be noted that "*Ofl*" is active  $\forall$  *Scl*  $\in$  *ed* alone where deviceand energy-dependent *RW* are computed. This increases the  $\rho_{RW}$  and *ed*, augmenting  $F(Q_L, E_d(c)_{ea})$  (refer to Figure 5). Therefore, the energy availability performs the remaining user demands for energy-reserved scheduling or offloading, i.e., the remaining reward, identifying until the device exhaustion is prevented. Therefore, the remaining edge device computing is processed in this continuous manner, reducing the data flows and time consumption in the edge network.

# 4. Results and Discussion

The results and discussion section presents the self and comparative analysis of the metrics used in the CEMS. First, the scheme was experimentally verified using a Contiki Cooja environment with 110 edge devices and N = 1200. The scheduling interval was varied between 5 and 60 min, with a maximum time-out of 12 s. In this scenario, six resource servers were used for computing and allocating requests and resources for the  $U^s$ .

Figure 6 presents the self-analysis on  $D_L(\%)$  for the varying N and  $\rho_{of}$  and  $\rho_{ED}(c)$ . The proposed scheme relies on two distinct RW for device and energy for performing allocations. For confining  $D_L$ , the  $Id_p$  in  $C_T$  is performed using  $\rho_{of}$  probability. In the process,  $c \neq 0$  and  $N > (C_T * c]$  conditions are validated for concurrent allocations. The allocations are performed with the Rw maximization  $\forall (tk) = true$  and  $\rho_{of} = 1 (max)$ conditions. Therefore, the  $D_L$  facing conditions are suppressed through increasing  $\rho_{of}$  and hence allocations are maximum. The continuous computation and energy allocations are performed using the *ea* and *ed* phases. In particular, RW is estimated to *ec* for preventing failures. This means that the  $Id_p = True$  is satisfied using "Scl" or "Ofl" and hence the  $u^s > C_T$  is handled. Therefore, the continuous allocations endure the available edge devices for N tasks. The energy distribution and allocation are concurrent with preventing  $D_L$  in these state models, achieving fair consumption. The *ec* phase is instigated after this allocation and "Scl". In Figure 7, the energy and Scl analysis for the varying  $\rho_{Rw}$  are presented.



**Figure 6.** Analysis of  $D_L$  and  $\rho_{E_d(c)}$  for the varying *N*.



**Figure 7.** Analysis of energy (%) and *Scl* (min) for the varying  $\rho_{Rw}$ .

The self-analysis for energy (%) and "Scl" for the varying  $\rho_{RW}$  is presented in Figure 7. The energies (%)  $\forall ea, ed$ , and ec are analyzed in this analysis, and the ea is high for the varying rewards due to  $\rho_{ED}(c)$  and  $N \in C_T$ . In the "Alloc "-based state model, the "Scl" and "Ofl" are induced for an increasing  $(N * C_T)$  rate. Therefore, the ed increases; the ec case is different from the other two phases. Depending on the allocation, less  $C_Tec$  is performed. This computation determines the devices and remaining N for allocation. Therefore, as the need for ec arises to a mid ed and ea, it is unstable throughout  $C_T$ . The "Scl" time demands are varying based on "Alloc"; if  $\rho_{ED}(c)$  is high, then "Scl" is high. In particular, the  $D_L(N)$  is increased post the ec observed. This is intermittent for the distinguishable "Scl".

### 4.1. Comparative Analysis

Depending on the experimental information, the metrics of computing rate, energy efficacy, offloading ratio, scheduling failure, and computing time are comparatively analyzed. The methods DPESO [23], E2E\_DRL [17], and EA-DFPSO [18] are used alongside the proposed CEMS in the comparative analysis.

#### 4.2. Computing Rate

As shown in Figure 8, edge device computing performs certain tasks depending on the time consumption rate of energy efficiency and did not control data flow between the different states to swap the allocated timing. The offloading and computational state requires user needs and demands used for identifying the scheduling failure and computing time. The analysis of scheduling and offloading is performed to augment edge devices' sustainability to control functions and operations for energy efficacy using the state learning process. This offloading problem is identified for different edge devices in that network depending on the load for the condition  $u^s \times E_d(c)$  and  $E_d(c) \times C_T$  used for performing remaining tasks. The continuous edge devices distributed in the network achieve successive scheduling over the offloading. The cognitive energy management preventing device exhaustion and therefore further edge device computation for energy efficiency are not presented. The state learning satisfies maximum energy efficiency based on the time slot and devices in that network, preventing data flows. Therefore, a high computing rate is achieved due to operating different edge devices.



Figure 8. Computing rate analysis.

# 4.3. Energy Efficacy

This proposed scheme achieved high energy efficacy for operating and controlling edge devices through state learning at different time intervals for identifying the scheduling failures in edge computing (refer to Figure 9). The scheduling failures and offloading are mitigated relying on user demands and energy consumption for sustainability over time of edge device allocation based on energy availability and computing time through state learning. The successful scheduling is due to unwanted data flow in edge device computing failures compared to the other factors in this proposed scheme. The offloading feature processing relies on scheduling or offloading, which requires the output for energy efficiency of edge device computing to identify the reward through a Q-learning process. Therefore, the energy efficacy of the edge device was analyzed for increasing the allocation within the time consumption depending on other factors. Hence, the energy efficacy is high in this proposed scheme.



Figure 9. Energy efficacy analysis.

# 4.4. Offloading Ratio

In this proposed scheme the scheduling failure and offloading ratio in edge device computing between different states for time consumption and identification processes do not process three segments depending on the load. The computation time of energy management in that edge network relies on the scheduling and offloading of an appropriate set of data to provide service for the users of  $u^s \times E_d(c)$  and is computed for identifying scheduling failures in edge device computing. The offloading feature is processed for performing certain tasks for reward allocation at different time intervals for  $E_d(c) \times C_T$  the condition. The successful scheduling over the offloading depends on the edge devices through state learning for performing three segments. The reliable working of the edge devices is analyzed with energy availability through state learning, preventing scheduling failures. The computation states and offloading for improving energy efficacy in the edge network are used to control the convergence that incorporates the previous reward computed for different states for detecting offloading without increasing the reward function. The proposed scheme performs edge computing based on the process reallocation to save the minimum energy for which energy availability achieves a lower offloading ratio as presented in Figure 10.



Figure 10. Offloading ratio analysis

#### 4.5. Scheduling Failure

The probability of energy conservation and energy distribution based on the edge device computing analysis for the sustainability of that network is illustrated in Figure 11. In this proposed scheme swapping the computing intervals for scheduling or offloading requires less computing time. The time consumption rate and identification process rely on energy management at various allocated time intervals. In these scheduling failures and offloading based on previous validated rewards,  $u^{s} > C_{T}$  and N task performance and computing are the considering metrics for edge computing. The offloading mitigates the individual allocation of rewards depending on the edge device computing and features vary for device exhaustion, wherein the swapping of energy-reserved scheduling based on maximum energy efficiency is preceded using Equations (4)-(9) computations. In this proposed energy management scheme, the minimum and maximum energy distribution in edge computing depend on reward identification. These edge devices are distributed to prevent scheduling failures under independent computing (as in Equations (10)–(12)). Therefore, the scheduling failure identification used for controlling energy management is high compared to the other factors. Based on these processes in the edge network, the scheduling failure is reduced at different time intervals.



Figure 11. Scheduling failure analysis.

# 4.6. Computing Time

As shown in Figure 12, the probability of edge device computing performed for energy efficiency analysis using energy management is processed for operating edge devices in that network and does not swap different states at allocated time intervals. The scheduling failure identification is organized for performing certain tasks from the previous reward computed and the offloading ratio is considered for improving computing time. Based on the scheduling, offloading, and allocation for energy management based  $Scl-\rho_{E_d(c)}$ on the condition , the probability is analyzed in a consecutive manner of improving energy efficiency. This scheduling failure identification is computed from the edge devices using the state learning paradigm in current scheduling and relies on allocated time and analysis, preventing offloading. The energy conservation and distribution are analyzed and processed based on the scheduling over the offloading, and the reward is allocated through the Q-learning process at different intervals for maximizing energy. The device allocation relies on energy availability considering the successful scheduling in energy management for which the proposed scheme requires less computing time. The above analysis's summary is tabulated with the findings in Tables 1 and 2 for the varying edge devices and scheduling intervals.





This scheme improves the computing rate and energy efficacy by 7.2% and 9.32%, respectively. It reduces the offloading ratio, scheduling failures, and computing time by 14.97%, 7.27%, and 14.48% respectively.

This scheme improves the computing rate and energy efficacy by 7.43% and 9.36%, respectively. It reduces the offloading ratio, scheduling failures, and computing time by 15.81%, 7.81%, and 14.93% respectively.

Metrics	DPESO	E2E_DRL	EA-DFPSO	CEMS
Computing Rate (Scheduling/Devices)	22	41	51	67
Energy Efficiency	54.96	64.78	73.66	83.11
Offloading Ratio	43.25	32.36	27.52	19.406
Scheduling Failures	0.158	0.113	0.094	0.049
Computing Time (ms)	1193.1	879.2	491.2	112.21

Table 1. Analysis summary for edge devices.

Table 2. Analysis summary for scheduling interval.

Metrics	DPESO	E2E_DRL	EA-DFPSO	CEMS
Computing Rate (Scheduling/Devices)	22	39	52	68
Energy Efficiency	55.34	65.25	74.47	83.746
Offloading Ratio	42.69	33.79	27.78	18.947
Scheduling Failures	0.185	0.125	0.074	0.0499
Computing Time (ms)	1414.1	1072.8	596.4	107.29

# 5. Conclusions

This article introduces a cognitive management scheme to improve edge devices' computation and energy efficacy. This scheme performs differentiated task scheduling and cognitive offloading using state learning. The scheduling is based on a first-come, first-serve process wherein the offloading is performed using energy allocation feasibility. The continuous energy allocation, distribution, and computation factors are analyzed using independent state models. The models increase the chances for energy conservation amid the distribution and allocation phases. In the energy-conserved phase, the allocations are prevented, preventing data losses and hence early energy exhaustion. The reward function is used for identifying the offloading/scheduling-required intervals. This facilitates the decision on energy distribution or allocation for the pending and new tasks. In the concurrent allocation intervals, device availability and energy conservation features are estimated using the current states in maximizing energy efficacy. This scheme improves the computing rate and energy efficacy for varying intervals by 7.43% and 9.36%, respectively. It reduces the offloading ratio, scheduling failures, and computing time by 15.81%, 7.81%, and 14.93%, respectively.

Author Contributions: Conceptualization, V.K.K., and D.M.; methodology, D.M., E.C., V.K.K., and P.T.; software, V.K.K., A.B.L.R., and P.T.; validation, T.A.N., S.P., and S.J.; formal analysis, P.T., and T.A.N.; investigation, A.B.L.R., S.P., and D.M.; resources, A.B.L.R., and T.A.N.; data curation, E.C.; writing—original draft preparation, V.K.K., A.B.L.R., and S.P.; writing—review and editing, V.K.K, D.M., and S.P., visualization, D.M. and E.C.; supervision, D.M.; project administration, D.M.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2020R1A6A 1A03046811). This work was supported by the National Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT (MIST)) (No. 2021R1A2C209494311).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- 1. Hu, X.; Wang, L.; Wong, K.K.; Tao, M.; Zhang, Y.; Zheng, Z. Edge and central cloud computing: A perfect pairing for high energy efficiency and low-latency. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 1070–1083. [CrossRef]
- Chen, Y.; Zhang, N.; Zhang, Y.; Chen, X.; Wu, W.; Shen, X.S. TOFFEE: Task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing. *IEEE Trans. Cloud Comput.* 2019, 9, 1634–1644. [CrossRef]
- Kang, M.; Li, X.; Shi, Z.; Rahman, T. Collaboration-oriented Boundary Energy Efficiency Detection for Continuous Objects in Edge Networks. *Procedia Comput. Sci.* 2022, 202, 128–134. [CrossRef]
- 4. Zhang, J.; Zheng, R.; Zhao, X.; Zhu, J.; Xu, J.; Wu, Q. A computational resources scheduling algorithm in edge cloud computing: From the energy efficiency of users' perspective. *J. Supercomput.* **2022**, *78*, 9355–9376. [CrossRef]
- 5. Guo, M.; Li, Q.; Peng, Z.; Liu, X.; Cui, D. Energy harvesting computation offloading game towards minimizing delay for mobile edge computing. *Comput. Netw.* **2022**, 204, 108678. [CrossRef]
- Chen, Y.; Zhang, N.; Zhang, Y.; Chen, X.; Wu, W.; Shen, X. Energy efficient dynamic offloading in mobile edge computing for internet of things. *IEEE Trans. Cloud Comput.* 2019, *9*, 1050–1060. [CrossRef]
- Liu, P.; An, K.; Lei, J.; Zheng, G.; Sun, Y.; Liu, W. SCMA-Based Multiaccess Edge Computing in IoT Systems: An Energy-Efficiency and Latency Tradeoff. *IEEE Internet Things J.* 2021, 9, 4849–4862. [CrossRef]
- 8. Zhou, Z.; Li, Y.; Li, F.; Cheng, H. An Intelligence Energy Consumption Model based on BP Neural Network in Mobile Edge Computing. *J. Parallel Distrib. Comput.* **2022**, *167*, 211–220. [CrossRef]
- 9. Li, C.; Zhang, Y.; Gao, X.; Luo, Y. Energy-latency tradeoffs for edge caching and dynamic service migration based on DQN in mobile edge computing. *J. Parallel Distrib. Comput.* **2022**, *166*, 15–31. [CrossRef]
- Ren, D.; Li, X.; Zhou, Z. Energy-efficient sensory data gathering in IoT networks with mobile edge computing. *Peer-Netw. Appl.* 2021, 14, 3959–3970. [CrossRef]
- 11. Khan, I.; Tao, X.; Rahman, G.S.; Rehman, W.U.; Salam, T. Advanced energy-efficient computation offloading using deep reinforcement learning in MTC edge computing. *IEEE Access* 2020, *8*,82867–82875. [CrossRef]
- 12. Zeng, Q.; Du, Y.; Huang, K.; Leung, K.K. Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing. *IEEE Trans. Wirel. Commun.* 2021, 20, 7947–7962. [CrossRef]
- Zheng, X.; Shah, S.B.H.; Bashir, A.K.; Nawaz, R.; Rana, U. Distributed hierarchical deep optimization for federated learning in mobile edge computing. *Comput. Commun.* 2022, 194, 321–328. [CrossRef]
- 14. Hu, Y.; Huang, H.; Yu, N. Resource Optimization and Device Scheduling for Flexible Federated Edge Learning with Tradeoff Between Energy Consumption and Model Performance. *Mob. Netw. Appl.* **2022**, *27*, 2118–2137. [CrossRef]
- 15. Pérez, S.; Arroba, P.; Moya, J.M. Energy-conscious optimization of Edge Computing through Deep Reinforcement Learning and two-phase immersion cooling. *Future Gener. Comput. Syst.* **2021**, *125*, 891–907. [CrossRef]
- 16. Ali, Z.; Khaf, S.; Abbas, Z.H.; Abbas, G.; Muhammad, F.; Kim, S. A deep learning approach for mobility-aware and energy-efficient resource allocation in MEC. *IEEE Access* 2020, *8*, 179530–179546. [CrossRef]
- 17. Ale, L.; Zhang, N.; Fang, X.; Chen, X.; Wu, S.; Li, L. Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 881–892. [CrossRef]
- 18. Irtija, N.; Anagnostopoulos, I.; Zervakis, G.; Tsiropoulou, E.E.; Amrouch, H.; Henkel, J. Energy efficient edge computing enabled by satisfaction games and approximate computing. *IEEE Trans. Green Commun. Netw.* **2021**, *6*, 281–294. [CrossRef]
- 19. Lu, Y.; Liu, L.; Gu, J.; Panneerselvam, J.; Yuan, B. EA-DFPSO: An intelligent energy-efficient scheduling algorithm for mobile edge networks. *Digit. Commun. Netw.* **2022**, *8*, 237–246. [CrossRef]
- 20. Wen, J.; Yang, J.; Wang, T.; Li, Y.; Lv, Z. Energy-efficient task allocation for reliable parallel computation of cluster-based wireless sensor network in edge computing. *Digit. Commun. Netw.* **2022**. [CrossRef]
- 21. Xie, Z.; Poovendran, P.; Premalatha, R. Retention based energy harvesting technique for efficient internet of things aided edge devices. *Sustain. Energy Technol. Assess.* **2021**, 47, 101424. [CrossRef]
- 22. Dai, H.; Wu, J.; Wang, Y.; Xu, C. Towards scalable and efficient Deep-RL in edge computing: A game-based partition approach. *J. Parallel Distrib. Comput.* **2022**, *168*, 108–119. [CrossRef]
- 23. Zhang, Y.; Fu, J. Energy-efficient computation offloading strategy with tasks scheduling in edge computing. *Wirel. Netw.* **2021**, 27, 609–620. [CrossRef]
- 24. Alsubhi, K.; Imtiaz, Z.; Raana, A.; Ashraf, M.U.; Hayat, B. MEACC: An energy-efficient framework for smart devices using cloud computing systems. *Front. Inf. Technol. Electron. Eng.* 2020, *21*, 917–930. [CrossRef]
- Li, C.; Cai, Q.; Luo, Y. Multi-edge collaborative offloading and energy threshold-based task migration in mobile edge computing environment. *Wirel. Netw.* 2021, 27, 4903–4928. [CrossRef]
- 26. Zhou, Z.; Shojafar, M.; Abawajy, J.; Yin, H.; Lu, H. ECMS: An Edge Intelligent Energy Efficient Model in Mobile Edge Computing. *IEEE Trans. Green Commun. Netw.* 2021, 6, 238–247. [CrossRef]
- Liu, X.; Liu, J.; Wu, H. Energy-aware allocation for delay-sensitive multitask in mobile edge computing. J. Supercomput. 2022, 78, 16621–16646. [CrossRef]
- Xie, W.; Li, B.; Xiong, Y.; Liu, W.; Ou, J.; Fan, D. Energy efficient collaborative computation for double-RIS assisted mobile edge networks. *Phys. Commun.* 2022, 2, 101774. [CrossRef]