energies

Article

A New Task Scheduling Approach for Energy Conservation in Internet of Things

Man-Wen Tian [1], Shu-Rong Yan [1], Wei Guo [2], Ardashir Mohammadzadeh [3,*] and Ebrahim Ghaderpour [4,5,*]

1 National Key Project Laboratory, Jiangxi University of Engineering, Xinyu 338000, China
2 School of Credit Management, Guangdong University of Finance, Guangzhou 510521, China
3 Multidisciplinary Center for Infrastructure Engineering, Shenyang University of Technology, Shenyang 110870, China
4 Department of Earth Sciences and CERI Research Centre, Sapienza University of Rome, Piazzale Aldo-Moro, 5, 00185 Rome, Italy
5 Earth and Space Inc., Calgary, AB T3A 5B1, Canada
* Correspondence: a.mzadeh@sut.edu.cn (A.M.); ebrahim.ghaderpour@uniroma1.it (E.G.)

Abstract: Internet of Things (IoT) and mobile edge computing (MEC) architectures are common in real-time application scenarios for improving the reliability of service responses. Energy conservation (EC) and energy harvesting (EH) are significant concerns in such architectures due to the self-sustainable devices and resource-constraint edge nodes. The density of the users and service requirements are further reasons for energy conservation and the need for energy harvesting in these scenarios. This article proposes decisive task scheduling for energy conservation (DTS-EC). The proposed energy conservation method relies on conditional decision-making through classification disseminations and energy slots for data handling. By classifying the energy requirements and the states of the mobile edge nodes, the allocation and queuing of data are determined, preventing overloaded nodes and dissemination. This process is recurrent for varying time slots, edge nodes, and tasks. The proposed method is found to achieve a high data dissemination rate (8.16%), less energy utilization (10.65%), and reduced latency (11.44%) at different time slots.

Keywords: decision-making; edge nodes; energy harvesting; IoT; task scheduling

Citation: Tian, M.-W.; Yan, S.-R.; Guo, W.; Mohammadzadeh, A.; Ghaderpour, E. A New Task Scheduling Approach for Energy Conservation in Internet of Things. *Energies* **2023**, *16*, 2394. https://doi.org/10.3390/en16052394

Academic Editor: Álvaro Gutiérrez

Received: 16 December 2022
Revised: 24 February 2023
Accepted: 1 March 2023
Published: 2 March 2023

1. Introduction

The Internet of Things (IoT) is an interconnection of devices, machines, objects, or people with a unique identifier and can transfer several pieces of data through the network [1]. The transferred data does not require any human-to-human or human-to-computer interaction. The data from the sensor are aggregated and communicated with the other device based on the request and response manner [2]. In the modern world, the IoT Smart TV, speaker, wearable sensors, smart application, etc., act as the IoT in which the data are transferred for efficient communication [3]. It is also used to monitor the weather condition and traffic on the road. Security is an important issue during information transmission in the network. The problem is based on the safety and vulnerabilities of the device information, and, hence, different techniques are used [4]. Using IoT creates the purpose of generating real-time data that can be evaluated. The data are observed and created for business outcomes and other applications [5].

IoT enables the device to interact, collaborate, learn, and share data. It is controlled using three different types of services, such as mechanical, electrical, and electronic systems [6]. The application is created using specific methods; initially, the data are aggregated efficiently. Then the data performs high streaming created by the IoT platform; this enables effective data management in the network [7]. The application of IoT includes the smart city, industrial internet, smart farming, smart retail, etc. IoT edge is the service embedded in IoT that uses edge computing [8]. In this edge computing, the raw data are forwarded by removing, collecting, and examining the device and the results to the cloud [6,8]. The edge device is the gateway connected with the cloud environment, controller, sensor, and intelligent device. Edge computing tracks the IoT edge device in a cloud-based interface. It is also used for controlling data flow at the boundary of two network interfaces which requires more computation power [9].

Energy harvesting (EH) improves the power expenses of the devices by conserving and sharing the complex process, such as system collaborations, data transmission, and device integration in a distributed manner. Doing this, the device makes communication more effective and flexible in a reliable method. The device with sufficient energy factor can transmit the energy from one device to another effectively [10]. The energy source has a small amount of energy to release. At the time of significant data transfer, according to the range of the data, the energy is released [11]. The data transmission process consumes high and low energy according to the data usability. For the effective data transmission, the energy is derived from the external sources, such as thermal energy, gradient devices, etc., which is defined as the energy harvested [12]. The node sends the data and goes into the sleep state, conserving energy and increasing the network lifetime [13]. Many energy harvesting devices include solar, thermal, and piezoelectric sources. The data are obtained from the edge device, and sensing and aggregation are distributed based on energy availability among the devices. In addition to this, task scheduling is incorporated with the energy harvesting process to improve resource allocation. During the task scheduling resource allocation patterns are examined at a period to allocate the resource. The task scheduling process covers the planning, requirement analysis, and resource conflicts. However, the task scheduling process does not consider the edge node availability and energy factor [14].

The objective of this paper is to conserve energy, performed by scheduling the data based on energy and edge node availability and balancing the dissemination and queuing processes. The main contributions of this study are:

- To improve the data dissemination rate using the decisive task scheduling process;
- To minimize the delay and high energy consumption using the conditional decision-making and energy slots handling procedure;
- To apply the queuing process to balance the edge node availability among the device while allocating the resources.

2. Literature Survey

Wei et al. [15] proposed an IoT using dynamic edge computation offloading to conserve energy using learning methodologies. The Markov decision process is used for statistical information. The optimal offloading is used to learn the complexity of extended time data and decrease the number of iterations.

Distributed analytics is introduced for energy efficiency at the edge of the network. Here, the IoT environment is considered by Valerio et al. [16]. They proposed the fog computing hypothesis transfer learning which is used for mobile nodes processed by IoT devices to fog gateways.

Sun et al. [17] developed an IoT service composition based on concurrent request integration optimization (CRIO) for efficient energy usage. It finds the standard function components and participates in the active concurrent requests for IoT services. The shared components enhance resource utilization in the network using particle swarm optimization (PSO).

Yan et al. [18] modeled the IoT applications under unstable channel conditions. This work performs dynamic energy-efficient data offloading for regular energy consumption. Task scheduling reduces the task reliability for communication between the channels state.

Green IoT-based heterogeneous wireless nodes are used for a scalable energy efficiency scheme (SEES) implemented by Abdul-Qawy and Srinivasulu [19]. SEES consists of three types of networks: zone-based hybrid placement, multi-stage weighted election heuristic, and the minimum cost cross-layer transmission model.

Fog planning for the real-time support of IoT applications is designed by Naranjo et al. [20] for energy-efficient under resource management. A middleware layer exploits the dynamic real-time scaling for a virtualized network. It also results in low complexity by using a bin packing type heuristic.

Secure relay selection is made for energy harvesting for IoT and was introduced by Huo et al. [21]. It also performs an outage if the secondary user device is trusted or untrusted. Estimation-assisted jamming is used for communication, whereas the Vickrey auction-based EH relay is used for the secondary system.

Min et al. [22] proposed a computation of offloading for IoT devices for energy harvesting. A deep reinforcement learning-based offloading (DRLO) is introduced with EH to track the edge device, which is performed by the offloading rate by notifying the current battery status level. It is performed by using a prediction-based methodology.

Pan et al. [23] developed an energy-efficient transient computing paradigm. The edge devices have ultra-flow energy harvesting to supply power. Their model includes two lightweight modules, routine handler, and frequency modulator (FM), for efficient runtime clock frequency modulation.

IoT-edge-cloud computing systems for efficient energy and guaranteed delay workload allocation was designed by Guo et al. [24]. The delay-based issues are between the local edge server, neighbor edge servers, and cloud. It decreases the energy consumption of the delay-based workload allocation (DBWA) algorithm.

Tang et al. [25] introduced a nonorthogonal multi-access assisted based on mobile edge computing for energy consumption. In this work offloading is used for consumption; an online energy consumption minimization is implemented for latency constraints in IoT devices. The offloading is used to forward the power and CPU cycle frequencies.

Xiang et al. [26] proposed a matrix-filling theory for dynamic traffic IoT to reduce the delay and increase energy-efficient data aggregation. Delay and energy-efficient data collection (DEEDC) is performed to gather the data obtained from wireless sensor networks (WSN) and process the clustering technique. A mixed slot scheduling strategy is used for collision-free scheduling.

3. Decisive Task Scheduling for Energy Conservation (DTS-EC)

Energy conservation conserves the energy from the edge device and utilizes it for further devices and processing in IoT. The main goal of energy harvesting is to harvest and provide energy to IoT nodes. In this work, storing and utilizing energy opts for conservation. Other devices are utilized to transmit and process information in the network environment. Therefore, other devices consume energy while processing these functions. This paper aims to manage the energy in IoT to further device usage and maximize the network's lifetime. The device's obtained energy is conserved by the following equation.

$$e_n = \sum_{\alpha} \left[p_0 \left(\alpha' - \alpha_0 \right) \right] + \frac{1}{2} \left[e_n^o - re_n \right] \tag{1}$$

In Equation (1), the energy is calculated for conservation; e_n and p_0 respectively denote the energy and time of the process; a task in the network is represented as α, and the task start and end times are, respectively, denoted as α' and α_0. The energy obtained from the device is referred to as e_n^o, and re_n is the remaining energy.

In Figure 1, the IoT-Edge environment with the process of DTS-EC is illustrated. The scope of this work is to find the energy usage in the network and control it until the timely edge node data transfer has been performed. For these data, scheduling is conducted based on the obtained and received energy from the edge device.

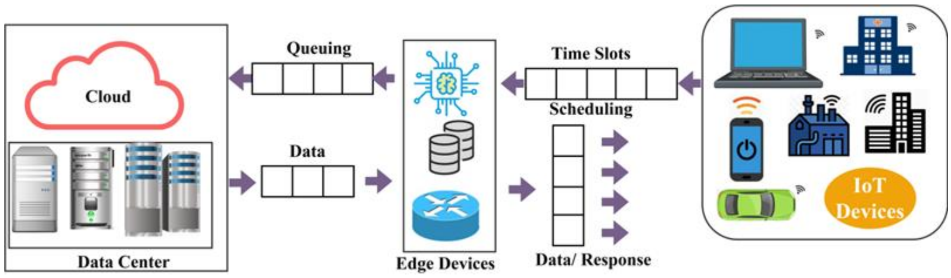


Figure 1. DTS-EC in IoT-Edge Environment.

3.1. Energy-Based Task Scheduling

The scheduling is based on energy, where the incoming data from the device are processed in the queue. The schedule considers the timely data acquired from the edge device. The data that arrive first are initially processed in the queue, and the rest are aligned based on the forthcoming time. The following equation is used to obtain the energy scheduling for the data in the queuing list.

$$s = \begin{cases} t_a + p_0 & \text{if} \quad \prod_{e_n^o}^{re_n} (\alpha' - \alpha_0) > \alpha / (me_n - le_n) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

By evaluating Equation (1), the energy conserved is equated to the scheduling required to perform to the energy needed to transfer the data. The scheduling is performed by queuing the tasks based on the energy required from minimum to maximum. The queued tasks are sent according to the requirements in the network. In Equation (2), s, t_a, and p_0 represent the scheduling, task, and process times, respectively. Here, "if the condition" is used for the obtained and remained energies. The time taken is $\alpha' - \alpha_0$, and the maximum and minimum energies required for the data transfer are denoted as me_n and le_n, respectively. In Figure 2a, the process of node state determination for the queuing data is illustrated.

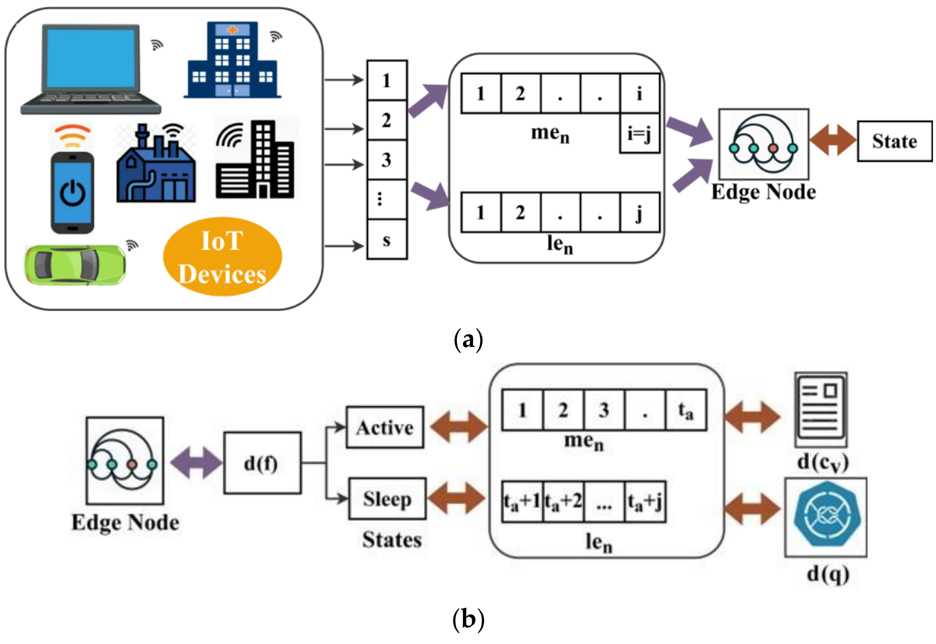


Figure 2. (**a**) Queuing data process based on node state, and (**b**) data allocation based on states.

The energy is scheduled by maximum and minimum requirements, and after that, the data transfer is performed by obtaining the edge nodes that are busy or ideal. Ideal nodes are nodes that do not participate in data transmission. Initially, if the data are to be sent, they check the edge node status. Doing this, the network lifetime increases, and energy is conserved if the edge node senses the data and sends them to rely on the edge node and goes to the sleep state. The following equation is used to evaluate the active and sleep state of the edge nodes in the network.

$$d(f) = e_n^0 + \prod_{p_0}^{\alpha} \left[t_a \to d_0 + \left(\alpha' - \alpha_0 \right) \right] \tag{3}$$

The energy conserved from Equation (1) is calculated using this. Equation (2) is used to identify the scheduling method for data transfer in the network. In Equation (3), d is represented as edge nodes if the data are sent to the other edge node d_0, and a specific process takes the required time limit. Moreover, f stands for the state of the node (active or sleep). In the fixed time, the data are transferred to the network; the edge node senses the data, transfers them, and goes to the sleep state.

The first level is when these data are transferred within the specified time limit, and so $d(f) = e_n^0$. In another level, $\prod\limits_{p_0}^{\alpha}[t_a \to d_0 + (\alpha' - \alpha_0)] \neq 0$, the data transfer is not equal to zero, which denotes the transmission failed in this case. Using the scheduling method, the data are transferred to the network. This is calculated by evaluating Equation (3). After this, the following Equations (4) and (5) are used to state the active and sleep states of the edge node. If these states are observed, then it is easy to transmit the data.

$$d(c_v) = \sum p_0 \times \begin{cases} \frac{g_0}{\sqrt{\frac{e_n^0}{re_n}}} + t_a \to i_n + f(t_a) < c_v \\ \frac{g_0}{\sqrt{\frac{e_n^0}{re_n}}} + t_a \to i_n + f(t_a) > c_v \end{cases} \tag{4}$$

Equation (4) is used to find the active state of the edge node, and it is derived in two states; the first state defines the sent data are sensed g_0 and forwarded to the initial edge node/near-edge node i_n. If the data transfer is not sensed, then there is a sleep state. The first state satisfies and is denoted as the edge node being active (c_v). It is derived if the processing is less than the active state, whereas, the second state is not satisfying the data processing.

$$d(q) = \frac{e_n}{d} + \begin{cases} \sum\limits_{g_0=0}^{g_0(d)} t_a \to i_n(p_0) \to d_0 < q \\ \sum\limits_{g_0=0}^{g_0(d)} t_a \to i_n(p_0) \to d_0 > q \end{cases} \tag{5}$$

In Equation (4), the active state of the edge node is determined; in another case, the sleep state is calculated by deriving the equation above Equation (5). In Equation (5), it is associated with two cases to identify the sleep edge node. In Figure 2b, the data allocation process is shown.

The initial case is $\sum_{g_0=0}^{g_0(d)} t_a \to i_n(p_0) \to d_0 < q$. Here, verifying data to be transferred means the neighboring edge node transfers the data if the initial edge node is busy with other processes. In this case, there is termed data forwarding if it is lesser than the sleeping state. The second case is $\sum_{g_0=0}^{g_0(d)} t_a \to i_n(p_0) \to d_0 > q$. In this case, the processing is not satisfied with data forwarding. Here, sensing itself, the process is terminated and leaves the network. By evaluating Equations (4) and (5), the following actions and sleep edge nodes are identified, and based on the requirement, the data are forwarded from the edge device to other devices in IoT. The following Equation (6) is calculated by applying scheduling for this two-state.

$$s(t_a) = (f \times d) + \sqrt{\frac{p_0(t_a)}{c_v - q}} - [me_n - le_m] \tag{6}$$

Considering Equations (4) and (5), Equation (6) is derived; here, the scheduling is performed for data sent to the network by evaluating the processing data and whether the edge node is active or sleeping. The energy is calculated by describing the maximum and minimum energy in the edge device. It is useful to describe the energy for the process, whether it needs more energy to complete the task or less energy. For transferring data, less energy is required, and rest of the energy is stored for the other process in the network. The decision-making method is used to conserve the remaining energy for the forthcoming process; in this work, the decision is made to allocate energy to the process. This gives better energy conservation and uses the remaining for the other device.

3.2. Decision Making

The decision-making is used to decide the energy required for starting and ending the process in the network. In this proposal, the decision is made for a single process to determine the amount of energy requirement. Based on the scheduling, the other process is queued in the list. The process task is to check for the network's active and sleep edge nodes, which is necessary to allocate the energy for the data.

The allocation of energy is not to be diminished in many cases; for this, initial verification is needed. The edge nodes forward the scheduled process and go to the sleep state. In this network, lifetime increases; if this is observed, then the network's energy is saved. The following equation is used to derive the process in the queue and gives the required energy for processing.

$$p_0 = (t_a + \alpha)\left[\frac{\sum_{\alpha'}^{\alpha}(g_0 + i_n)}{s/(me_n - le_n)}\right] * (c_v - q) \tag{7}$$

In Equation (6), the scheduling is carried out for the data, and from that, the processing of the data is observed by evaluating Equation (7) which processes the data based on the specific time interval. Two types of processing are involved.

The first process is $\left[\frac{\sum_{\alpha'}^{\alpha}(g_0+i_n)}{s/(me_n - le_n)}\right] * (c_v - q) = 0$, where the starting and ending times of the process are considered; if the process is sensed in the network, the edge nodes tend to forward the data. Figure 3 presents the slot allocation process based on starting and ending times. In this case, the edge node forwarding is calculated to have the maximum and minimum energy spent on the data. The active and sleep modes are determined, and the result ($= 0$) means the processing is conducted on a scheduled queue in the given time interval.

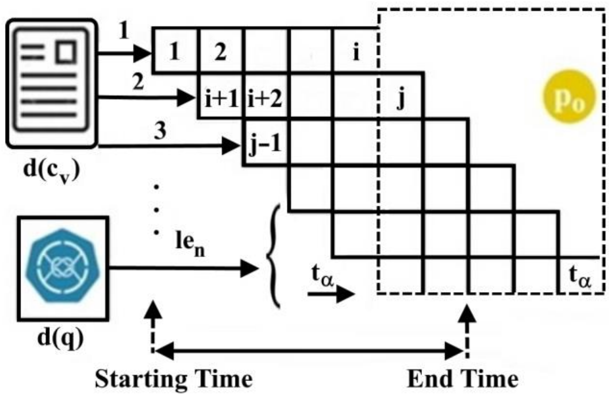


Figure 3. The slot allocation for starting and ending times.

The second process is $\left[\frac{\sum_{\alpha'}^{\alpha}(g_0+i_n)}{s/(me_n - le_n)}\right] * (c_v - q) \neq 0$, where the sensed edge node is processed in the specified time interval, denoted as not equal to zero. By evaluating this process, allocation in the queue uses the energy in the derived form. The following Equation (8) extracts the process and energy bases.

$$e_n(p_0) = \begin{cases} \left. \begin{array}{l} f \rightarrow (t_a + d_0 * e_n^o) - \prod_s^{re_n} g_o + q = le_n \\ f \rightarrow (t_a + d_0 * e_n^o) - \prod_s^{re_n} g_o - q \neq le_n \end{array} \right\} \\ \left. \begin{array}{l} f \rightarrow (t_a + d_0 * e_n^o) - \prod_s^{re_n} g_o + c_v = me_n \\ f \rightarrow (t_a + d_0 * e_n^o) - \prod_s^{re_n} g_o - c_v \neq me_n \end{array} \right\} \end{cases} \tag{8}$$

In Equation (7), the process time of the data is obtained by a scheduling approach, and Equation (8) is used to achieve the energy for the processing in four different types of transferring. The first stage is $f \rightarrow (t_a + d_0 * e_n^o) - \prod_s^{re_n} g_o - q = le_n$, which represents the data transfer in the network obtaining the energy from the device and denotes the sleeping state. This sleeping state uses only a minimum amount of energy to be transferred. The second state is $f \rightarrow (t_a + d_0 * e_n^o) - \prod_s^{re_n} g_o - q \neq le_n$, which represents that no process is carried out. The third state is $f \rightarrow (t_a + d_0 * e_n^o) - \prod_s^{re_n} g_o - c_v = me_n$, where the maximum energy is required to transfer because it denotes the active state of the edge node. If the edge node is active, it uses more energy in the network which leads to energy loss. The fourth state is $f \rightarrow (t_a + d_0 * e_n^o) - \prod_s^{re_n} g_o - c_v \neq me_n$, where no transfer is performed on the edge node because there are not equal to zero.

From the obtained equation, it is noted that the active state of the edge node consumes more energy for processing. The objective is to reduce the energy to attain energy conservation for other processing devices. The edge devices use the energy to forward the particular information to the other device; for this process, the energy is necessary for processing. The preliminary step is to track the data and how much energy is necessary to transfer the data to the destination. The following equation is used to evaluate the data verification state in the network.

$$s(u) = \sqrt{\frac{d(p_0)}{e_n}} + \prod_{\alpha_0}^{f} \left[\max_{t_a}(e_n + d_0) \right] \tag{9}$$

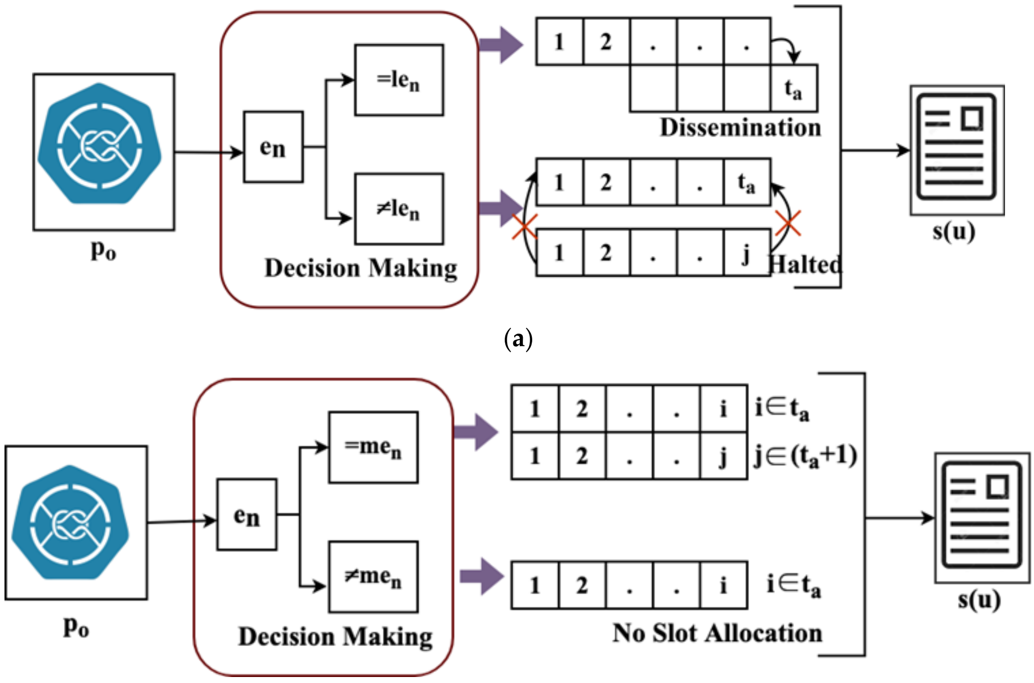
Equation (8) estimates the maximum and minimum energy where the edge nodes sleep and are active. Data verification is necessary to know the required energy used for Equation (9). In Equation (9), the verification is denoted as u, and the processing of the data to the edge node is termed as $d_0(p_0)$. Here, the maximum energy is represented using the functional value as max. Based on the time, the scheduled data from the queue are ready to forward, where the data which require maximum energy are separated.

The division is performed for maximum and minimum energy requirements. The advantage of this energy division is termed conserved energy usage. In Figure 4a,b the decision-making based on le_n and me_n is illustrated.

The scheduling already queues the data based on a timely manner; from which the data are again divided into maximum and minimum energy requirements. The following equation is used to achieve the separation process.

$$\beta = \begin{cases} \min_{t_a} \left[s + (\alpha_0 - \alpha') \right] < 1 \\ \max_{t_a} \left[s + (\alpha_0 - \alpha') \right] > 1 \end{cases} \tag{10}$$

By evaluating Equation (9), the data verification is performed on the scheduling process, and then by deriving Equation (10), the energy is divided by β. The first level of energy tends to be $\min_{t_a} \left[s + (\alpha_0 - \alpha') \right] < 1$, where the minimum energy is required for data transfer by evaluations (u). It is performed by referring to the stable state of the data in the network, which uses only less energy and has less than 1. This states that the first level is satisfied by using minimum energy.



(**a**)

(**b**)

Figure 4. Decision-making based on (**a**) le_n and (**b**) me_n.

The second is defined to have $\max_{t_a}[s + (\alpha_0 - \alpha')] > 1$, where it is necessary to acquire more energy for data transfer. Only the assigned energies are given to the data, and their allocation is greater than one. The energy processing is observed by evaluating Equation (11) as follows.

$$k = \prod_{s=0}^{t_s} \begin{cases} (d + e_n^o) - \frac{u}{\sum_{\beta}^{\alpha'} [d(q) + d(c_v)]} = \delta \\ (d - e_n^o) - \frac{u}{\sum_{\beta}^{\alpha'} [d(q) + d(c_v)]} \neq \delta \end{cases} \quad (11)$$

Following Equation (10), the energy is separated and used for storing. In Equation (11), the decision is made to process the energy allocation. Decision k is carried out in two conditions: the first condition is $(d + e_n^o) - \frac{u}{\sum_{\beta}^{\alpha'} [d(q) + d(c_v)]} = \delta$, where the obtained energy is used for dividing the energy. Symbol δ is termed energy conservation, where the first condition is satisfied. Based on the scheduling approach, the energy is conserved and stored for the forthcoming process. The second condition is $(d + e_n^o) - \frac{u}{\sum_{\beta}^{\alpha'} [d(q) + d(c_v)]} \neq \delta$, where the energy conservation process failed. The objective of this work is satisfied by using Equation (11), based on $s(u)$. The conserved energy is stored and used for the other process, which increases the network's lifetime. Herein, both c_v and q are considered for scheduling the data in the network. Finally, the decision is made for the energy allocation to the forthcoming process.

4. Results and Discussion

The performance metrics, such as energy utilization, data dissemination, active edge nodes, and latency are used to validate the efficiency of the proposed method. The experiments are performed using the Contiki Cooja simulator, in which 60 IoT-edge nodes are deployed to serve 220 users/user equipment with heterogeneous data. In this experimental setup, 50 time slots are allocated for data dissemination and queuing. The number of tasks is varied between 50 and 500, for which 60 ms is the maximum time for allocation after scheduling. The failing requests/tasks results in service halts, increasing the error rate. For data dissemination, edge nodes utilize 0.2 W of transmit power from their initial energy of

16 J. In this work, an air traffic control-related task scheduling process is performed using a different number of nodes. Using this setup, the following metrics are compared with the existing scalable energy efficiency scheme (SEES) [19], composition based on concurrent request integration optimization—particle swarm optimization (CRIO-PSO) [17], and deep reinforcement learning-based offloading (DRLO) [22] for verifying the consistency of the proposed method. This work uses the models in [17,19,22] to compare the introduced system because the existing system uses the energy consumption procedure to improve overall data analysis and resource allocation process. The effective utilization of functions maximizes overall performance. This is the main reason why references [17,19,22] are used for comparison purposes.

4.1. Energy Utilization

Figure 5 presents the comparative analysis of energy utilization for the varying IoT-Edge-Node, time slots, and tasks. Unlike the other existing methods, the proposed method does not require energy for all the data in the dissemination process. Instead, it classifies the scheduling slots based on energy levels as le_n and me_n.

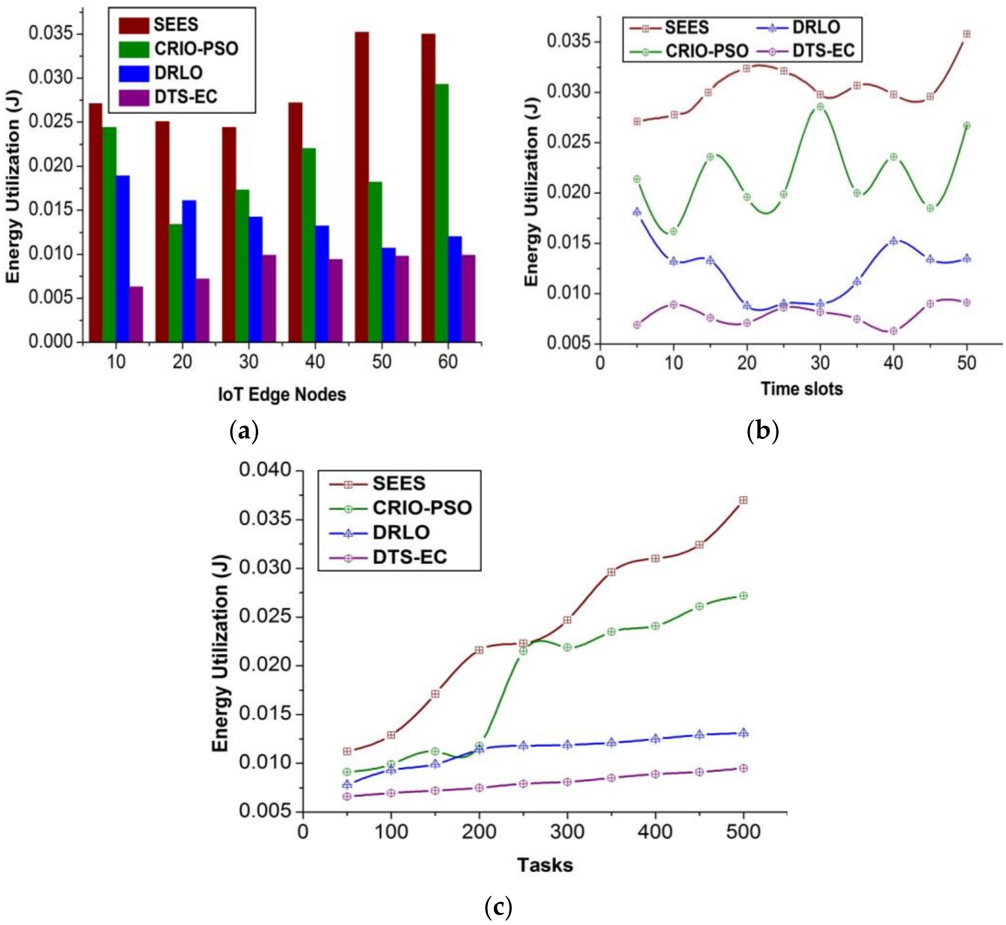


Figure 5. Energy utilization comparisons for (**a**) IoT edge nodes, (**b**) time slots, and (**c**) tasks.

This classification permits different time slot acceptance and scheduling for queuing the input requests. For the e_n observed in time slots for an active IoT edge node, further scheduling is determined on the recommendation of the decision process as represented in Figure 4a,b, respectively. The transmission and queuing slots' availability provides consistent data handling without energy drain and/or additional requirements. The initial scheduling based on s is classified by the states of the nodes permitting data transfer or halts in accessing the queues. Therefore, the excessive energy requirement is confined in this decision-making process using $s(t_a)$ and p_o for all e_n and available edge nodes.

The pre-determination of energy-based scheduling slots with appropriate decisions, as in Equations (8) and (11), helps to confine the utilization in any level of dissemination. This process is common for both the varying tasks and available slots. Here, the density and availability of the edge nodes are significant for the data handling process. However, the high data dimensionality creates computation complexity and deviations which leads to changes in energy consumption.

4.2. Data Dissemination

The comparative analysis presented in panels (a), (b), and (c) of Figure 6 validates the performance of the proposed method for varying edge nodes, time slots, and energy utilization, respectively.

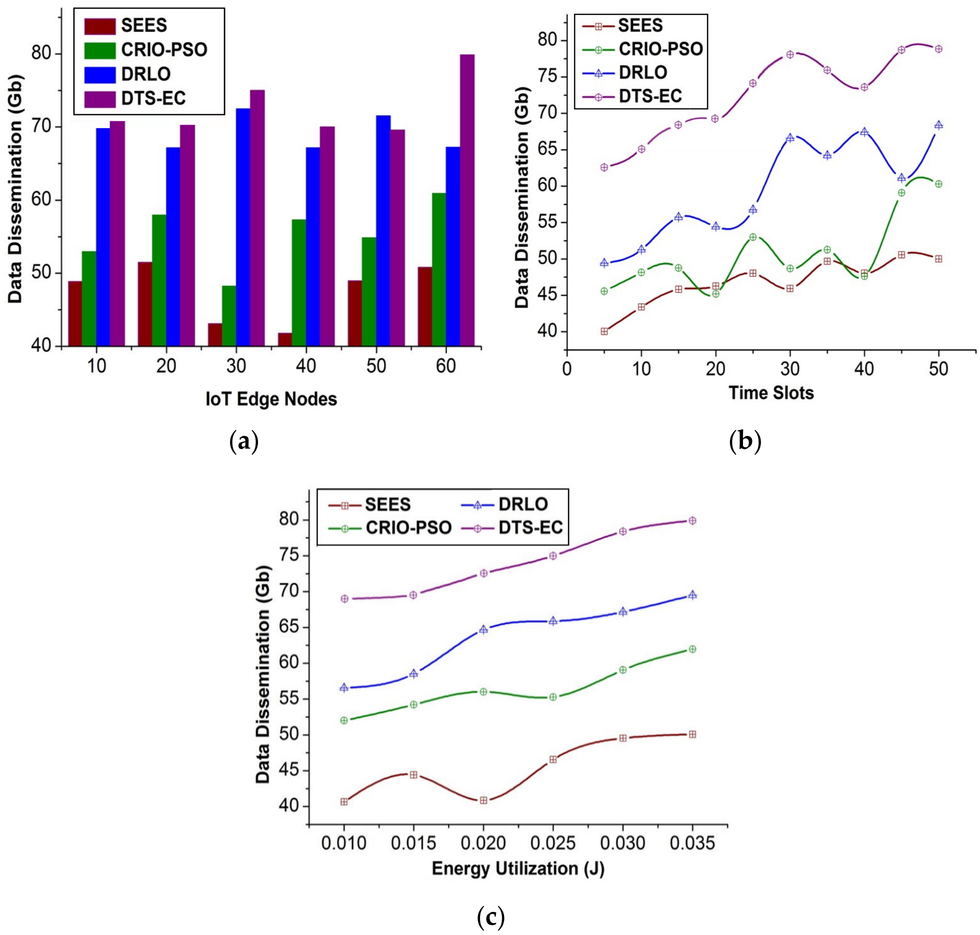


Figure 6. Data dissemination comparisons for (**a**) IoT edge nodes, (**b**) time slots, (**c**) energy utilization.

In this comparative analysis, the energy consumption observed in the previous utilization is considered as me_n (or) le_n, and prevents overloading/prolonged wait time of the cloud data. In all four conditions, as in Figure 4a,b, the classification is based on $s(t_a)\ \forall\ e_n$ and available active state nodes. For a sleep state node, the queuing is performed depending on the maximum wait time of the task allocated to the edge node. Therefore, either concurrent or sequential dissemination of data as per $d(c_v)$ is affordable in handling any amount of data in the dissemination slots. The processing of queued data follows $d(q)$ as in Equation (5) for satisfying $d(q) < q$ and $d(q)$ conditions. Similarly, for each $f \rightarrow (t_a + d_0 * e_n^o)$, $s(u)$ is verified to ensure t_a is the maximum time slot required for dissemination. The separation, as in Equation (10) and its associated δ verification using Equation (11), leverages the data dissemination rate irrespective of the energy utilization, time slots, and edge nodes. However, in all these methods, the availability of the edge nodes based on their energy state is prominent.

4.3. Active Edge Nodes

In Figure 7, the active edge nodes after multiple iterates are presented as a comparison. If an edge node is said to possess remaining energy after handling all the data in its allocated interval, it is said to be active. This data handling is monitored through the iterates to verify the devices' longevity. The remaining energy of the edge nodes is given by Equation (12) as

$$re_n = Ie_n - e_n \quad (12)$$

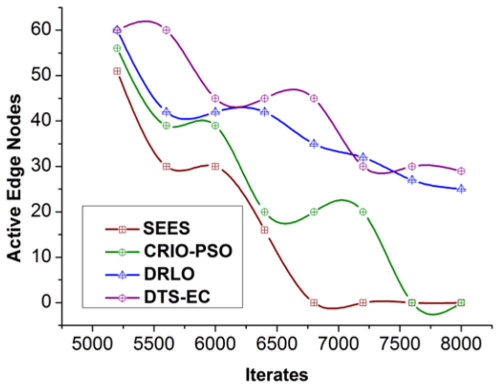


Figure 7. Comparisons between the results of active edge nodes for the methods in this study.

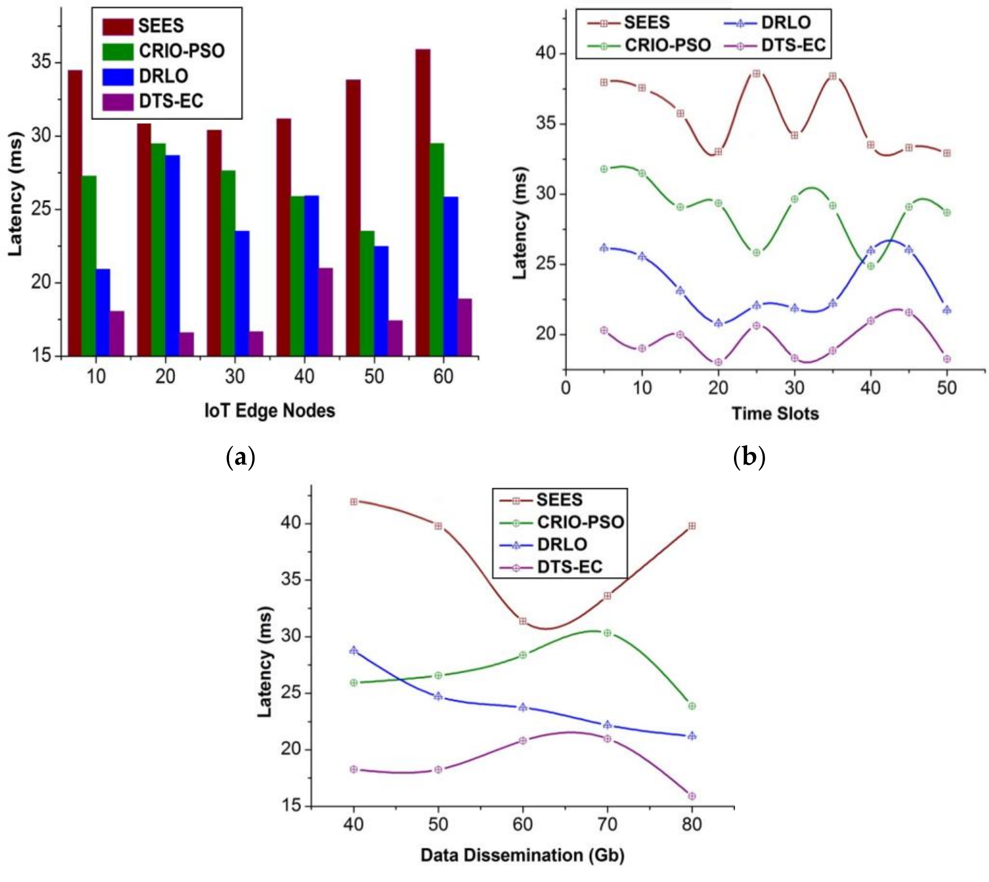
In Equation (12), Ie_n denotes the initial energy of the IoT-edge node. The node exploits energy for data dissemination and queuing processes.

The proposed method refines unnecessary overloading of nodes and, hence, allocates slots for processing tasks and their requests. The scheduling for the slots is classified for energy consumption rates, preventing the additional loss in dissemination. Therefore, assigning additional slots/edge nodes with further energy expenses is prevented in this method. The number of iterates increases the possibility of reducing excessive energy utilization by mapping $d(c_v)$ for the appropriate $d(q)$. Here, $d(q)$ is filtered based on the energy slot allocation using $s(t_a)$. The decision-making process determines p_o and $s(u)$ without overloading the active state nodes, preventing energy drain. Therefore, the energy conservation rate is high, retaining a high level of nodes in the active state in the proposed method.

4.4. Latency

For the different IoT-edge nodes, time slots, and data dissemination rate, the latency is compared in Figure 8. The slot allocation in the proposed method relies on energy variants me_n and le_n by identifying the active nodes. In the decision-making process, $e_n(p_o)$ classifies the different dissemination instances based on le_n and me_n. Following this process, the dissemination ensures $s(u)$ for all the allocated time slots. This prevents overloading of the time slots and reduces the wait time for the queuing tasks.

The queuing process prevents unnecessary drop/re-transmission of data for a new task. For the p_o, the active nodes are selected to ensure $d(c_v)$ is disseminated, whereas, the idle node is selected for queuing. The other process ensures both queuing and dissemination in a sequential/concurrent manner, preventing excessive wait time for the tasks. These flexible features help improve the data rate through all the available active nodes, with less waiting time and less latency. However, the high data dimensionality of data analysis often leads to deviations and creates computation complexity, which leads to changes in the latency.



(a) (b) (c)

Figure 8. Latency Comparisons for (**a**) IoT edge nodes, (**b**) time slots, and (**c**) data dissemination.

4.5. Summary of Findings

In Tables 1 and 2, the comparative analysis of the above is tabulated for varying IoT edge nodes and time slots. Table 1 shows that the proposed method adapts to the varying IoT edge nodes, achieving 10.78% less energy utilization, 8.44% better data dissemination, and reducing latency by 12.61%. Table 2 shows that the proposed method achieves 10.65% less energy utilization, 8.16% high data dissemination, and 11.44% less latency for the varying time slots.

Table 1. Comparative analysis for varying IoT edge nodes.

Metrics	SEES	CRIO-PSO	DRLO	Our Work
Energy Utilization (J)	0.035	0.0293	0.012	0.009
Data Dissemination (Gb)	50.84	60.93	67.22	79.88
Latency (ms)	35.9	29.49	25.83	18.9

Table 2. Comparative analysis for varying time slots.

Metrics	SEES	CRIO-PSO	DRLO	Our Work
Energy Utilization (J)	0.0358	0.0267	0.0131	0.0091
Data Dissemination (Gb)	50.01	60.3	68.36	78.85
Latency (ms)	32.94	28.68	21.74	18.25

5. Conclusions

This article introduces decision task scheduling for energy conservation in IoT-edge assisted data dissemination scenarios. The proposed method identifies the states of the edge nodes based on their energy levels and then assigns data dissemination slots. The initial differentiation of slots for queuing and dissemination improves data availability and non-overloading dissemination. Using conditional decision-making, the energy levels, the states of the nodes, and data rates are balanced for optimal dissemination. In the decision-making process, conditional analysis is performed for different data handling and slot allocation instances. This helps to perform both concurrent and sequential data dissemination. Moreover, this method permits a non-overloading node based on the remaining energy and determines the maximum limit of data forecasting through the available slot. This manifold process of decision-making improves the efficiency of the proposed method by achieving fair energy utilization (10.78%), data dissemination (8.44%), and active nodes count under controlled latency (12.61%) for different nodes.

Author Contributions: Writing—original draft, M.-W.T.; Writing—review & editing, S.-R.Y., W.G., A.M. and E.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Social Science Foundation; Grant number: 18AJY013;20BJY044;21BJY112;22BJY101. The APC was handled by E.G.

Data Availability Statement: The datasets used in this research may be provided upon request.

Acknowledgments: The authors thank the reviewers for their constructive comments that greatly helped to improve the presentation of the paper. The last author (E.G.) also thanks the CERI Research Centre at the Sapienza University of Rome for their support.

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

Abbreviations

DTS-EC	Decisive Task Scheduling for Energy Conservation
EH	Energy Harvesting
DEEDC	Delay and energy-efficient data collection
FM	Frequency Modulator
SEES	Scalable Energy Efficiency Scheme
CRIO	Composition based on concurrent Request Integration Optimization
PSO	Particle Swarm Optimization
DRLO	Deep Reinforcement Learning-Based Offloading
IoT	Internet of Things
MEC	Mobile Edge Computing

References

1. Al-Azez, Z.T.; Lawey, A.Q.; El-Gorashi, T.E.H.; Elmirghani, J.M.H. Energy Efficient IoT Virtualization Framework with Peer to Peer Networking and Processing. *IEEE Access* **2019**, *7*, 50697–50709. [CrossRef]
2. Li, Y.; Orgerie, A.-C.; Rodero, I.; Amersho, B.L.; Parashar, M.; Menaud, J.-M. End-To-End Energy Models for Edge Cloud-Based IoT Platforms: Application to Data Stream Analysis in IoT. *Future Gener. Comput. Syst.* **2018**, *87*, 667–678. [CrossRef]
3. Bushnaq, O.M.; Chaaban, A.; Chepuri, S.P.; Leus, G.; Al-Naffouri, T.Y. Sensor Placement and Resource Allocation for Energy Harvesting IoT Networks. *Digit. Signal Process.* **2020**, *105*, 102659. [CrossRef]
4. Shrivastav, K.; Kulat, K.D. Scalable Energy Efficient Hexagonal Heterogeneous Broad Transmission Distance Protocol in WSN-IoT Networks. *J. Electr. Eng. Technol.* **2019**, *15*, 95–120. [CrossRef]
5. Ashraf, N.; Hasan, A.; Qureshi, H.K.; Lestas, M. Combined Data Rate and Energy Management in Harvesting Enabled Tactile IoT Sensing Devices. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3006–3015. [CrossRef]
6. Munoz, R.; Vilalta, R.; Yoshikane, N.; Casellas, R.; Martinez, R.; Tsuritani, T.; Morita, I. Integration of IoT, Transport SDN, and Edge/Cloud Computing for Dynamic Distribution of IoT Analytics and Efficient Use of Network Resources. *J. Light. Technol.* **2018**, *36*, 1420–1428. [CrossRef]
7. Samie, F.; Tsoutsouras, V.; Masouros, D.; Bauer, L.; Soudris, D.; Henkel, J. Fast Operation Mode Selection for Highly Efficient IoT Edge Devices. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 572–584. [CrossRef]

8. Mishra, S.K.; Puthal, D.; Sahoo, B.; Sharma, S.; Xue, Z.; Zomaya, A.Y. Energy-Efficient Deployment of Edge Dataenters for Mobile Clouds in Sustainable IoT. *IEEE Access* **2018**, *6*, 56587–56597. [CrossRef]
9. Toor, A.; Islam, S.U.; Ahmed, G.; Jabbar, S.; Khalid, S.; Sharif, A.M. Energy Efficient Edge-of-Things. *Eurasip J. Wirel. Commun. Netw.* **2019**, *2019*, 82. [CrossRef]
10. Huang, J.; Li, S.; Chen, Y. Revenue-Optimal Task Scheduling and Resource Management for IoT Batch Jobs in Mobile Edge Computing. *Peer Peer-to-Peer Netw. Appl.* **2020**, *13*, 1776–1787. [CrossRef]
11. Qiu, C.; Wu, F.; Lee, C.; Yuce, M.R. Self-Powered Control Interface Based on Gray Code with Hybrid Triboelectric and Photovoltaics Energy Harvesting for IoT Smart Home and Access Control Applications. *Nano Energy* **2020**, *70*, 104456. [CrossRef]
12. Din, I.U.; Hassan, S.; Almogren, A.; Ayub, F.; Guizani, M. PUC: Packet Update Caching for Energy Efficient IoT-Based Information-Centric Networking. *Future Gener. Comput. Syst.* **2019**, *111*, 634–643. [CrossRef]
13. Ke, H.; Wang, J.; Wang, H.; Ge, Y. Joint Optimization of Data Offloading and Resource Allocation with Renewable Energy Aware for IoT Devices: A Deep Reinforcement Learning Approach. *IEEE Access* **2019**, *7*, 179349–179363. [CrossRef]
14. Jamil, B.; Ijaz, H.; Shojafar, M.; Munir, K.; Buyya, R. Resource Allocation and Task Scheduling in Fog Computing and Internet of Everything Environments: A Taxonomy, Review, and Future Directions. *ACM Comput. Surv.* **2022**, *54*, 1–38. [CrossRef]
15. Wei, Z.; Zhao, B.; Su, J.; Lu, X. Dynamic Edge Computation Offloading for Internet of Things with Energy Harvesting: A Learning Method. *IEEE Internet Things J.* **2019**, *6*, 4436–4447. [CrossRef]
16. Valerio, L.; Conti, M.; Passarella, A. Energy Efficient Distributed Analytics at the Edge of the Network for IoT Environments. *Pervasive Mob. Comput.* **2018**, *51*, 27–42. [CrossRef]
17. Sun, M.; Zhou, Z.; Wang, J.; Du, C.; Gaaloul, W. Energy-Efficient IoT Service Composition for Concurrent Timed Applications. *Future Gener. Comput. Syst.* **2019**, *100*, 1017–1030. [CrossRef]
18. Yan, H.; Zhang, X.; Chen, H.; Zhou, Y.; Bao, W.; Yang, L.T. DEED: Dynamic Energy-Efficient Data Offloading for IoT Applications under Unstable Channel Conditions. *Future Gener. Comput. Syst.* **2019**, *96*, 425–437. [CrossRef]
19. Abdul-Qawy, A.S.H.; Srinivasulu, T. SEES: A Scalable and Energy-Efficient Scheme for Green IoT-Based Heterogeneous Wireless Nodes. *J. Ambient. Intell. Humaniz. Comput.* **2018**, *10*, 1571–1596. [CrossRef]
20. Naranjo, P.G.V.; Baccarelli, E.; Scarpiniti, M. Design and Energy-Efficient Resource Management of Virtualized Networked Fog Architectures for the Real-Time Support of IoT Applications. *J. Supercomput.* **2018**, *74*, 2470–2507. [CrossRef]
21. Huo, Y.; Xu, M.; Fan, X.; Jing, T. A Novel Secure Relay Selection Strategy for Energy-Harvesting-Enabled Internet of Things. *Eurasip J. Wirel. Commun. Netw.* **2018**, *2018*, 264. [CrossRef]
22. Min, M.; Xiao, L.; Chen, Y.; Cheng, P.; Wu, D.; Zhuang, W. Learning-Based Computation Offloading for IoT Devices with Energy Harvesting. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1930–1941. [CrossRef]
23. Pan, C.; Xie, M.; Hu, J. ENZYME: An Energy-Efficient Transient Computing Paradigm for Ultralow Self-Powered IoT Edge Devices. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *37*, 2440–2450. [CrossRef]
24. Guo, M.; Li, L.; Guan, Q. Energy-Efficient and Delay-Guaranteed Workload Allocation in IoT-Edge-Cloud Computing Systems. *IEEE Access* **2019**, *7*, 78685–78697. [CrossRef]
25. Li, C.; Tang, J.; Zhang, Y.; Yan, X.; Luo, Y. Energy Efficient Computation Offloading for Nonorthogonal Multiple Access Assisted Mobile Edge Computing with Energy Harvesting Devices. *Comput. Netw.* **2019**, *164*, 106890. [CrossRef]
26. Xiang, X.; Liu, W.; Wang, T.; Xie, M.; Li, X.; Song, H.; Liu, A.; Zhang, G. Delay and Energy-Efficient Data Collection Scheme-Based Matrix Filling Theory for Dynamic Traffic IoT. *Eurasip J. Wirel. Commun. Netw.* **2019**, *2019*, 168. [CrossRef]