Research Article

Mixed Linear Programming for Charging Vehicle Scheduling in Large-Scale Rechargeable WSNs

P. Suman Prakash, M. Janardhan, K. Sreenivasulu, Shaik Imam Saheb, Shaik Neeha, and M. Bhavsingh

1Department of Internet of Things, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India
2Department of Computer Science & Engineering, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India
3Department of Computer Science & Engineering, Lords Institute of Engineering and Technology, Hyderabad, Telangana, India
4Department Computer Science Engineering (Artificial Intelligence and Machine Learning), Lords Institute of Engineering and Technology, Hyderabad, Telangana, India

Correspondence should be addressed to P. Suman Prakash; sumancse@gpcet.ac.in

Received 18 August 2022; Accepted 25 August 2022; Published 9 September 2022

Academic Editor: Praveen Kumar Donta

Copyright © 2022 P. Suman Prakash et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Because wireless sensor networks (WSNs) have low-constrained batteries, optimizing the network lifetime is a primary challenge. Rechargeable batteries are a solution to prolong the lifetime of a sensor node instead of restricting their functionalities to save energy. Wireless energy transmitters have the added benefit of providing a charger for the batteries of the sensor nodes in the WSN. However, scheduling one or more charging vehicles efficiently to recharge multiple sensor nodes is challenging. In this context, this paper provides a solution to recharge the sensor nodes using charging vehicle scheduling in WSNs through a mixed linear programming approach. Initially, we identify a heuristic value of each sensor node based on their residual energy, distance from a charging vehicle, available data packets, and other metrics. Further, a set of nodes is recharged by identifying the best charging vehicle to prolong their lifetimes, as well as the lifetime of the network as a whole. We simulated the proposed approach using a Python simulator, tested using different performance metrics, and compared using the recently published works. We notice the superior performance of the proposed work under various metrics in time and query-driven WSNs.

1. Introduction

The sensor nodes (SNs) of wireless rechargeable sensor networks (WRSNs) have rechargeable batteries, which can be charged to extend their lifespan [1]. There are several ways in the literature to recharge the sensor nodes, such as using energy harvesting techniques, battery-equipped unmanned aerial vehicles (UAVs) [2], solar panel-equipped SNs, and charging stations in between the network [3, 4]. Through recent advancements, one of the most promising approaches is wireless energy transmitters (WET). In general, a WET is equipped with a vehicle called a charging vehicle (CV), which is scheduled in the network to recharge the SNs without affecting their topological disorders [5]. A CV is equipped with a battery of unlimited energy (due to energy harvesting), and it recharges the nodes while traveling in the network. Mainly, the CV visits a set of points called anchor points (APs) which is one hop distance from a group of nodes to recharge parallelly. There are several approaches in the literature to identify an optimal set of APs, with their benefits and limitations [6–8]. These approaches also specify a path for the CV to traverse in the network.

In large-scale WSNs, it is not easy to schedule a CV to recharge the whole network because recharging a node takes a long time than the data collection process [9]. So, it is required to introduce more than one CV to recharge a demanded or group of nodes in the network. Increasing
the number of charging will also increase the deployment cost, but a lower number of CVs cannot solve the issue [10, 11]. It is challenging to identify the best set of CVs, increasing the charger efficiency by prolonging the network lifetime. Hence, it is necessary to provide an efficient algorithm to choose the best quantity of CVs to recharge the energy demanded SNs because it is lacking in the literature. When more than one CV is in the network, scheduling them to solve the recharging purpose is difficult. So, it is further challenging to assign a group of charging requests to a specific CV. There are several approaches in the literature, but they are not dynamic, and many techniques use random or heuristic strategies [12, 13].

The scheduling problem of requests and CVs is solved using a mixed-integer programming (MIP) approach in this paper. Before that, we identify the weights of each SN depending on various factors of an SN, including its residual energy, distance from the charger or base station, and available data packets. Using this weight, an MIP is applied to identify the best CV that can handle the charging request quickly and efficiently. With this motivation, we proposed a dynamic CV scheduling approach to prolong the lifetime through WET to recharge sensor nodes in WRSNs.

Overall, the contributions of the proposed MLPCV strategy are summarized as follows:

1. We propose a heuristic strategy to analyze the energy status and sustainability of each sensor node in the network and assigned a weight value to each node.
2. A mixed linear programming strategy is used to identify the suitable charging vehicle, which can be scheduled to recharge a set of sensor nodes efficiently.
3. The performance of the proposed work is evaluated under different scenarios and metrics using the existing charging vehicle scheduling approaches.

This paper is organized as follows: The recently published charger scheduling algorithms are analyzed in Section 2. The system model along with the problem statement is defined in Section 3. The proposed MLPCV algorithm is presented in Section 4 along with illustrative examples. This section also analyzes the complexity of MLPCV. The simulation results are discussed and plotted in Section 5. Finally, Section 6 concluded the paper with future scope.

2. Related Work

We find several works in the literature using WET-based mobile vehicles to recharge the sensor nodes in the WSNs. This section categorizes the previous works into two types based on the number of CVs used in their works, such as one or more CVs. We summarize recent and related works in this section with their benefits and challenges.

2.1. Single Charging Vehicles. A CV can serve one or more recharge requests where coverage is the primary objective than the longest sustainability of sensor nodes [14]. This approach provides efficient charging scheduling and increases connectivity, which further helps efficient data routing. Clustering is a common approach in WSNs, whereas this strategy is used to recharge the SNs in the WSNs using solar power. The efficiency and sustainability of the nodes are increased in this approach in terms of longer life. Similarly, Tomar et al. [15] use a clustering mechanism to identify the efficient AP to visit by the CV to recharge the needed nodes. The path of the CV is crucial in this approach, but this work is not for large-scale WSNs. The charger utility is maximized by Srinivas et al. in [16] using a heuristic strategy. This approach efficiently maximizes the CV utility while balancing the energy of the SNs among the network.

Identifying an efficient path for the CV is a challenging task in WSNs, whereas Jiang et al. [17] focused on this issue in their approach. Efficient anchor points are identified to recharge multiple nodes and serve the charging requests at a single instance. It takes time to restore an SN until the battery is full. Considering these challenges, a partial charging strategy is introduced in [18] with a genetic approach and multiattribute decision-making systems. This approach balances and simultaneously serve multiple requests to sustain the longer visibility of nodes. The charging scalability is optimized in [19] to maximize the CV’s efficiency. Multiple SNs are recharged simultaneously in [20], where the scheduling is decided based on the residual energies of the SNs. In [21], a hierarchical clustering approach is used to identify the optimal clusters to schedule the CV to serve maximum requests in the WSNs. Since the computational complexity of this approach is high, it is not feasible for large-scale networks.

Depending on the need for charging, the on-demand CV scheduling strategy is proposed in [22] for recharging more nodes to sustain the longer lifetime. In [23], multiple battery-filled sensor nodes are used to replace the SNs, which are critical with their battery. However, this is expensive to replace the number of SNs and not secure. It is also not feasible for all applications. Deep reinforcement learning is an efficient classification and learning method, and the benefits of this approach are used for WSNs to recharge the nodes in [24]. This approach is used a partial charging strategy, so the network lifetime is longer for small WSNs. This approach also takes high computational resources than other approaches. A collaborative charging strategy is presented in [25], in which the authors mainly focused on building an optimal tour for the CV. In this approach, the optimal charging points are decided to fit the large-scale WSNs.

From the above discussion, we identify several algorithms in the literature that use a single charging vehicle to recharge the sensor nodes in the network. However, these approaches are limited to small networks with limited SNs. The large-scale WSNs are deployed with unlimited SNs and receive multiple requests simultaneously. Handling such situations is not possible with a single charging vehicle.

2.2. Multiple Charging Vehicles. A new strategy for identifying the reason to charge a node along with the route from the current location of CVs to recharging nodes of WSNs
is presented in [21]. But identifying the best set of CVs required for the network to serve the requests efficiently is not determined in the paper. A traditional push-shuttle-back approach serves the charging requests in large-scale WSNs. This strategy is efficient and inexpensive but not optimal. Another charging request serving algorithm is presented in [26] using more CVs, and this approach achieves 1% efficiency over the traditional algorithms. However, this approach also failed to determine the optimal number of CVs that can serve the network requests. An electromagnetic radiation-based recharging algorithm is proposed for WSNs in [27]. In this, the TSP approach is used for the CV route to recharge the nodes, but this approach also failed to determine the optimal number of CVs that can solve the problem to minimize the CV cost. In [28], lifetime enhancement through CV is presented for large-scale WSNs. Unlike other traditional approaches, this approach determines the best number of CVs that can solve the charging requests in the network. However, this approach is computationally high and also not optimal.

With coverage intention, a CV scheduling approach is proposed for large-scale WSNs in [29]. In this strategy, identify the SNs with limited energy and give high priority to recharging them to maintain the connectivity and avoid sink-hole problems in the network. However, in [30], swarm intelligence is used to schedule multiple CVs in the network. In this, the APs are identified initially, and then CVs are scheduled based on the APs to recharge the nodes in the network. However, this approach failed to decide the best number of CVs required to restore the requested network nodes. Control of the velocity of the CVs is a critical challenge, and it is addressed in [31] using approximation algorithms. Still, this approach is not efficient and optimal in terms of choosing the APs in the network. In [32], an integer programming approach is used to schedule the CVs in large-scale WSNs. Still, this approach failed to choose the best possible CVs to cover the entire network.

A ring-wandering approach used to schedule the multiple CVs to recharge the SNs deployed in large-scale WSNs is presented in [33]. This approach is efficient in balancing the energy of the SNs and prolonging their life-span. However, the performance is improved but failed to identify the optimal number of CVs to control the cost. A clustering algorithm is also used in multiple CV scheduling in [34] using a statistical approach and heuristic formulae. This approach efficiently identifies the APs in the networks and controls the scheduling, but it does not provide the optimal number of CVs. In [35], an optimal number of CVs are determined to replenish the energy of the SNs in the network. However, this approach failed to produce the optimal route for the CVs and charging efficiency. In [36], a hybrid metaheuristic strategy is used to serve the charging requests by providing optimal scheduling. This approach efficiently finds the best set of APs and routes to recharge the nodes. Although this is efficient, it requires more computational resources to run the algorithm, whereas it is not feasible because of the constrained resources of WSNs.

As we noticed in this discussion, most of the works are using heuristic, clustering approaches to identify the APs and TSP for the traveling path for the CVs. While considering all these limitations, we proposed an efficient approach to schedule multiple CVs in the large-scale WSNs to recharge the nodes in the network. It can also decide the optimal visiting points.

3. Problem Formulation and Parameter Description

The representation of the WSNs is shown as a graph $G$ with a set of nodes and edges. The $n$ number of nodes are represented as $S = S_0 \cup S_j$, where $S_0$ is the BS and $S$ means the set of nodes. The proposed algorithm is centralized, so it is executed in $S_0$. So, the computational resources are unlimited for executing the code. The edges are identified using their communication range $\Gamma$ and denoted using $D$. Each $S$ is equipped with a battery of capacity $E^S$ and buffer of capacity $B$. The $k$ number of CVs are denoted using $C = \{c_1,c_2,\cdots,c_k\}$. The locations of the SNs are randomly decided using Sah et al. [37], and $S_0$ is located at $(0,0)$. Each node $s_i$ drains some energy during operations like acquiring data from the environment or other SNs ($s_j$), or sometimes, the nodes are idle. So, at a particular time $t$, a node $s_i$ have residual energy of $E^i_t$. Similarly, the buffer of each node is occupied with some data during the data transmissions, and at a time $t$, it is considered $B^i_t$ for a node $s_i$. At time $t = 0$, the energy of the battery is full capacity and the buffer is empty. $E^\text{min}_i$ is considered the threshold energy for any SN in $G$, and $s_i$ sends energy request to $S_0$ when it meets the condition $E^\text{min}_i \geq E^i_t$. All the charging and buffer capacities are updated through the control signal to $S_0$. The charging requests are categorized and assigned to CVs by $S_0$ in the proposed work.

The CV can move quickly with $u$ velocity, and the charging time is minimal. In the course of CV stay at the BS, it was recharged. The SN energy drain during the data transmission to the CVs is shown in

$$E^i_t = \sum_{i} v_i \times a_i + \sum_{i} \alpha_i \delta_i \forall i \in S^*, \quad (1)$$

where $\sum_i$ shows total packets transmitted by $s_j$; Equation (2) computes $\delta_{ij}$, where it is the distance from $s_i$ to $CV_j$ generated during processing a bit need $a_i$, energy, and amplification needs $\alpha_i$ energy:

$$\delta_i = \sqrt{(x_i - x_{cv_j})^2 + (y_i - y_{cv_j})^2}, \quad (2)$$

where $(x_i, y_i)$ are the coordinates of the SN $s_i$ and $(x_{cv_j}, y_{cv_j})$ is $CV_j$ location. The energy required to acquire sensed data from the field is

$$E^{tx}_i = \sum \alpha_i \forall i \in S^*, \quad (3)$$
where receiving a sensed packet from the field is required α, energy. The overall EC of SN $s_i$ is considered

$$E_r^i = E_r^0 + E_r^i \forall \ i \in S'. \quad (4)$$

The residual energy of SN at simulation time $t$ is considered

$$E_r^i = E_r^i - E_r^i \forall \ i \in S'. \quad (5)$$

The average residual energy of a cluster is computed as follows:

$$E_i = \sum_{j \in C_i} \frac{E_r^j}{|C_i|} \forall \ 1 \leq i \leq k. \quad (6)$$

In the recharging process, the energy harvest of SNs is calculated similarly to [38].

Through an optimal CV scheduling strategy with a low data loss rate, the proposed work is aimed at balancing the energy among the SNs.

4. Proposed MLPCV Algorithm

The proposed MLPCV primarily contains two modules. In the first module, different metrics are used to calculate the weight of each SN to determine the priority of charging. Based on these weights, the optimal CV is assigned to recharge the node. CVs are placed at a specific location in the network to get the highest variations in the weights of nodes instead of keeping them at BS. As a result, all the CVs are located at a particular distance from the SNs, and when they receive requests, it is easy to fulfill them quickly. The general model of the proposed work is summarized using Figure 1.

4.1. Finding CV’s Initial Location and SN’s Weight. In this section, we perform two operations including finding the initial deployment of the CVs and weight calculations.

4.1.1. CV Initial Locations. A new initial centroid selection approach is used in this phase based on enhanced k-means algorithms [39]. Aiming to select centroids that belong to different clusters, the farthest are chosen in such a way that they are different and vary from each other. As a result, more centroids will be likely to share the same cluster SNs. As a result, the chance of having data points from different centroids in the same cluster is maximized. This will result in a significant reduction in execution time. Algorithm 1 explains the detailed pseudocode of the proposed CV initial location ($L$) selection approaches.

Initially, we need to decide the number of CVs used in a particular environment. This is decided based on $k = \log_2(n)$ . Once $k$ is decided, it selects two locations ($L_1$ and $L_2$) based on the following:

$$L_1 = (\max (x_i), \max (y_j)) \forall \ 1 \leq i \leq n,$$

$$L_2 = (\min (x_i), \min (y_j)) \forall \ 1 \leq i \leq n. \quad (7)$$

This process iterates until $k$ locations are determined, and all the CVs are positioned in these locations initially. Further, we compute the weight of each SNs as known in the subsequent sections.

4.1.2. SN Weight Calculation. Weight calculation is an important metric and can be used to choose the charger that can recharge the node before it drains its entire power supply. In this context, we generate heuristic formulae to compute the weight of each SN. This uses different metrics such as distance from SNs to the CVs, heuristic information, and buffer capacity along with the energy information. Initially, we estimate a probability value as shown in

$$P_{ij} = \frac{\psi_{ij} \times d_{ij}^\beta}{\sum_{i=1}^n (\psi_{ij} \times d_{ij}^\beta)} \forall j \in k, \quad (9)$$

where $\psi_{ij} = r_j / x_j$ is the energy consumption rate of a CV. $\alpha$ and $\beta$ are the heuristic information, and $\omega$ is the wavelength, and $\eta$ is the efficiency of the rectifier. All these values are constants. The weight of SNs is shown in

$$W_{ij} = \frac{E_{max}}{E_i} \times P_{ij}, \quad (10)$$

As we know, most of the metrics used in Equations (9) and (10) are changed dynamically in the meantime. So, this process iterates after every unit time to notice the changes in the distance from nodes to CVs, residual energy, etc. So, the updated weight is supplied to the charging vehicle scheduling algorithm, to identify the best CV to recharge the SNs.

4.2. Charging Vehicle Scheduling. Once the weight calculation of all nodes ($W = \{W_i \forall i \in (1, n)\}$) is completed, we start assigning suitable CVs using the charger vehicle scheduling approach. This algorithm decides the suitable CV for each SN charging request according to their residual energy and distances, whereas a similar approach is implemented
Step 1. Arrange all the SN weights particular set or SNs or a single SN using the following steps.

1: $\mathcal{E} = \emptyset$
2: $c_1 = \max \{D(n)\}$
3: $c_2 = \min \{D(n)\}$
4: $\mathcal{E} = \{\mathcal{E} \cup c_1\}$
5: $\mathcal{E} = \{\mathcal{E} \cup c_2\}$
6: while (size($C$) < $k$)
7: $nC = DP(n, \mathcal{E})$
8: $\mathcal{E} = \{\mathcal{E} \cup nC\}$
9: end while
10: return $\mathcal{E}$

Algorithm 1: Initial CV location selection algorithm.

for the fog environment by Hazra et al. [40]. A weight matrix $E_{nk}$ is constructed for device allocation, of the size $n \times k$, in which the SNs are considered in rows and CVs considered in columns. The entry of $E_{nk}$ is the $\mathcal{W}_{nj}$ value, it is a nonnegative heuristic weight for a node $i(\in (1,n))$ to consider a CV $j(\in (1,k))$, and it is calculated as shown in equation (10). Using this information, we generate a scheduling matrix $\Psi$, where each entry of this matrix is either one or zero. The values associated with one mean an SN is assigned to a CV for recharging their battery. The primary object to generate $\Psi$ is to minimize

$$\max_{\forall i \in (1,n)} \sum_{j=1}^{k} \Psi_{(ij)} \times \mathcal{W}_{ij}$$

Subjected to $\Psi_{(ij)} \in \{0, 1\}, \forall i \in (1,n) \land j \in (1,k)$.

The primary goal of this scheduling is to prolong the lifetime of the network while balancing the energy of the SNs choosing the best CV to recharge a node. The charger vehicle scheduling uses the following steps to assign a CV to a particular set or SNs or a single SN using the following steps.

Step 1. Arrange all the SN weights ($\mathcal{W}_{nk}$) in ascending (non-decreasing) order such as $\mathcal{W}_{nk(1)} \leq \mathcal{W}_{nk(2)} \leq \mathcal{W}_{nk(3)} \leq \cdots \leq \mathcal{W}_{nk(\text{size}(\mathcal{E}))}$ where $\mathcal{R}$ indicates the rank of each element.

Step 2. Take the minimum $\mathcal{R}$ as an element ($\mathcal{W}_{nk(\mathcal{R})}$) in the $i^{th}$ row and $j^{th}$ column $E$ in ascending order while each row and column contain at least one entry.

Step 3. Change these entry $\mathcal{R}$ with $\mathcal{W}_{nk}$ of $E$ according to equation (12) shown below:

$$E_{nk} = \begin{cases} 0, & \text{if } \mathcal{W}_{nk} \leq \mathcal{R}, \\ \mathcal{W}_{nk}, & \text{otherwise}. \end{cases}$$

Step 4. From a CV column ($j$) which had the least number of zeros, schedule all the SNs ($i$) which had value one.

Step 5. Repeat Step 4 until all the sensor nodes are assigned at least one CV.

This process is repeated to recharge all the SNs in the WSNs until the data aggregation process is interrupted.

4.3. Illustration. The charging vehicle scheduling approach is illustrated using a simple example by considering ten nodes and six charging vehicles as shown in Figure 2. In general, for ten sensor nodes, one CV is sufficient, but our intention here is to show how the proposed algorithm works. Initially, all the weighted information calculated using equation (10) is stored in $E$ as shown in Figure 2(a). Next, we identify the ranks of each entry of $E$ based on nondecreases and replace the entries in $\Psi$ as shown in Figure 2(b). Our process consists of filling in rows and columns one by one until all rows and columns contain at least one entry, based on the rank of each entry. We start filling one by one rank in $\Psi$, and all the rows and columns contain at least one element after filling rank 22 as shown in Figure 2(c). Now, we use equation (12) and replace all the entries using zero for the no entries and replace the value associated with each ranking as shown in Figure 2(d). By giving high priority to $\Psi$ with the fewest zeros in a column, we can now assign each task to a resource from Step 4 of the charger vehicle scheduling approach. We need to find the rows with the highest number of zeros. As we noticed from Figure 2(d), we need to find the rows with the highest number of zeros such as $s_5$, $s_8$, and $s_{10}$. So, we assign the appropriate CV for these two SNs such as CV1 = $\{s_{10}\}$, CV4 = $\{s_8\}$, and CV2 = $\{s_5\}$ as shown in Figure 2(e). Further, all zeros are replaced using $\infty$, and assigned CVs are replaced as 1 as shown in Figure 2(f). The next highest zeros are found in $s_3$, $s_4$, and $s_9$, so we assign the appropriate CV for these nodes, and they are highlighted in Figure 2(g). If more than one charger is associated, we give high priority to the idle charger because the charging takes more time. As per this principle, $s_9$ can be assigned to CV1, and it can be represented as CV1 = $\{s_{10}, s_9\}$. Similarly, when checking $s_3$, there are two possibilities, and we can consider any one. To break the tie, we can choose the ascending order, so $s_3$ is assigned to CV5, and it becomes CV6 = $\{s_4\}$. We repeat this process, until all the nodes are visited, and the final assignment is shown in Figure 2(h), i.e., CV1 = $\{s_{10}, s_9, s_7\}$, CV2 = $\{s_5, s_6\}$, CV3 = $\{s_2, s_1\}$, CV4 = $\{s_8\}$, CV5 = $\{s_3\}$, and CV6 = $\{s_4\}$.

4.4. Complexity Study. The proposed MLPCV approach’s time complexity mainly works in two partitions: CV initial location selection along with the weight computation for each SN and the scheduling of the CVs to recharge the requested nodes. The time required to perform the initial CV locations using $k$-means required $O(n^3)$, and the weight calculation requires approximately $O(n \times k)$, where $k < n$. The time required to schedule is approximately $O(k \times n^2)$. The total complexity of MLPCV to recharge the requested nodes is $O(n^3) + O(n \times k) + O(k \times n^2)$. From this, the asymptotic complexity for MLPCV is $O(n^3)$, which is better than existing CV scheduling approaches. However, the proposed MLPCV is better than other existing approaches such as CSCT, M2C, and SPSS.
5. Simulation Studies

The simulator for experiments used Python (v3.11.0). A number of SNs are varied along with the size of the area to conduct experiments in two different scenarios. We consider 1000 to 1600 as the number of nodes, and they are deployed according to Sah et al. [37]. The CV battery capacity is 194.4 kJ, and its type is 12 V 4.5 Ah LiFePO₄ rechargeable battery. The total simulation time considered to run the experiments is 10 hours or 100 network cycles. Similarly, the packet generation also followed from this approach. The area size during experiments are changes between 600 sq.m and 1000 sq.m. $S_0$ is located at $(0, 0)$ in all the experiments and two scenarios. The packet generation and receiving and transmissions require 0.05 mJ, 0.02 mJ, and 0.02 mJ energy, respectively. The communication and transmission proximity is considered 15 m and 12 m, respectively. The TDMA protocol is used as a MAC layer protocol, and MQTT is used as the application layer protocol [41]. Each unit of time generates approximately four packets during

![Figure 2: Illustration of charging vehicle scheduling through an example of 10 nodes and six chargers.](image-url)

### Table: CV to SNs for recharging

| CV1 | CV2 | CV3 | CV4 | CV5 | CV6 |
|-----|-----|-----|-----|-----|-----|
| s1  | 0   | 0   | 38  | 39  | 44  |
| s2  | 0   | 0   | 21  | 41  | 0   |
| s3  | 0   | 0   | 22  | 29  | 0   |
| s4  | 0   | 0   | 0   | 36  | 41  |
| s5  | 0   | 45  | 0   | 0   | 0   |
| s6  | 15  | 44  | 0   | 0   | 34  |
| s7  | 28  | 0   | 33  | 0   | 23  |
| s8  | 0   | 0   | 0   | 36  | 0   |
| s9  | 25  | 0   | 32  | 0   | 0   |
| s10 | 24  | 0   | 0   | 0   | 0   |

### Table: The values associated with each rank

| CV1 | CV2 | CV3 | CV4 | CV5 | CV6 |
|-----|-----|-----|-----|-----|-----|
| s1  | 0   | 0   | 38  | 39  | 44  |
| s2  | 0   | 0   | 21  | 41  | 0   |
| s3  | 0   | 0   | 22  | 29  | 0   |
| s4  | 0   | 0   | 0   | 36  | 41  |
| s5  | 0   | 45  | 0   | 0   | 0   |
| s6  | 15  | 44  | 0   | 0   | 34  |
| s7  | 28  | 0   | 33  | 0   | 23  |
| s8  | 0   | 0   | 0   | 36  | 0   |
| s9  | 25  | 0   | 32  | 0   | 0   |
| s10 | 24  | 0   | 0   | 0   | 0   |

### Table: Final assignment of each node to a CV

| CV1 | CV2 | CV3 | CV4 | CV5 | CV6 |
|-----|-----|-----|-----|-----|-----|
| s1  | 0   | 0   | 38  | 39  | 44  |
| s2  | 0   | 0   | 21  | 41  | 0   |
| s3  | 0   | 0   | 22  | 29  | 0   |
| s4  | 0   | 0   | 0   | 36  | 41  |
| s5  | 0   | 45  | 0   | 0   | 0   |
| s6  | 15  | 44  | 0   | 0   | 34  |
| s7  | 28  | 0   | 33  | 0   | 23  |
| s8  | 0   | 0   | 0   | 36  | 0   |
| s9  | 25  | 0   | 32  | 0   | 0   |
| s10 | 24  | 0   | 0   | 0   | 0   |

The area size during experiments are changes between 600 sq.m and 1000 sq.m. $S_0$ is located at $(0, 0)$ in all the experiments and two scenarios. The packet generation and receiving and transmissions require 0.05 mJ, 0.02 mJ, and 0.02 mJ energy, respectively. The communication and transmission proximity is considered 15 m and 12 m, respectively. The TDMA protocol is used as a MAC layer protocol, and MQTT is used as the application layer protocol [41]. Each unit of time generates approximately four packets during
5.1. Average Charging Delay. An average charging delay (ACD) is the time it takes for all CVs in a WSN to charge a requested SN before it completely drains its energy. SNs are challenging to track because the CV does not reach them until they are completely drained. A travel time is included, as well as the charging of other requests waiting in line. Under two scenarios, including the size of the area and the number of SNs, we compute the average cost per node for a network. Traveling delay and queue size increase with area size and number of nodes. In order to simulate these two variables, we took into consideration both of them.

Figure 3(a) plots the results of SNs varying from 1000 to 1600 in 1000 sq.m area. We identify that the increasing number of SNs degrades the ACD, because more of the requested nodes are in queue. So, generally, it is difficult to serve all the nodes using available CVs. We notice that the ACD of the WSN is approximately 12 min after the end of the simulation time of 10 hours in the proposed MLPCV approach, which is the best compared with the existing techniques. Similarly, the existing CBCT results in a delay of approximately 18 min, M2C results in approximately 23 min, and the SPSS algorithm results in around 37 min delay on average. So, we can claim that the proposed work results in the least delay over the other techniques. In Figure 3(b), we consider the different area sizes such as 600 sq.m to 1000 sq.m while considering 1000 SNs which are deployed randomly in the network. With a constant number of sensors and a larger area, SNs have a lower communication burden than densely deployed sensor nodes. Therefore, it minimizes unnecessary data transmissions and reduces energy consumption. Consequently, there is a reduction in the number of requests for charging 1600 SNs in a 1000 sq.m area resulting in an average delay time of 12 minutes, and M2C results in a delay time of approximately 15 minutes. A delay of 18 seconds is used by the SPSS algorithm, and a delay of 21 seconds is used by the J-RCA algorithm.

5.2. Packet Reception Ratio. Based on the number of packets generated by each sensor node in a WSN, it is calculated. A total number of packets are received by the base station during the simulation time $T$ [42]. As the WSNs transmit packets, packet reception rates (PRRs) are directly proportional to throughput. There is a direct correlation between the PRR means and the performance of the system. Under two scenarios, including varying the number of SNs, the proposed MLPCV approach evaluates the PRR performance as shown in Figure 4(a) and PRR vs. the simulation time as shown in Figure 4(b). While varying the number of SNs between 1000 and 1600, the MLPCV always results in the highest performance compared to existing algorithms in Figure 4(a). Similar to the MLPCV algorithm, the PRR is detected when 1000 SNs are processed continuously, and the results are plotted in Figure 4(b). It is evident from the plots that the MLPCV performs better than existing algorithms such as M2C, SPSS, and J-RCA. The proposed work should improve on the existing algorithms in terms of PRR, so we strongly agree with that assumption.

5.3. Residual Energy. For the data collection operations in the WSNs to be sustainable, any sensor node’s remaining energy must be above zero. Increasing the RE results in a longer lifespan for the network as a result of a higher RE.

We compare the average RE between the proposed and existing works in Figures 5(a) and 5(b). Taking SNs from 1000 to 1600 in 1000 sq.m area and Figure 5(a), the average RE can be determined from the figure. Once the CV has been visited and recharged, the residual energy of any sensor node increases. Due to the fact that it can receive a much higher number of requests from a variety of sources in the network, in this scenario, the residual energy of the SNs is decreasing while increasing the number of SNs. A 100th cycle of the 1600 sensor nodes averages 4.7854 kJ for M2C and SPSS, but 2.3654 kJ for J-RCA. We propose an algorithm with an average RE of 5.96547 kJ, which is better than the existing algorithms.

Based on 1000 randomly deploying static SNs in the network, Figure 5(b) shows the average RE of the SNs. The large area and limited number of SNs result in a low data exchange rate. As a result, fewer requests are received by the CV, and delays are also reduced. As a bonus, the average RE of the sensor nodes will reduce congestion. According to the proposed work, the average RE is 7.1337 kJ for 1000 SNs in a 1000 sq.m area. There is an average RE of 5.3121 kJ for the existing M2C, SPSS, and J-RCA algorithms, respectively, 5.45 kJ, 4.5754 kJ, and 5.4454 kJ. In this case, there is a significant difference between the proposed algorithm and the three existing algorithms in terms of average RE.

5.4. Node Survival Rate. When the CV completes its cycle, the node survival rate indicates how many SNs survived. There are a large number of sensor nodes (SNs) in the network, and the area is extended, so it is not possible for all sensor nodes to be saved. As a result, we calculated the number of sustainable SNs after the mobile vehicle completed the threshold number of charging cycles and collected the data. As shown in Figure 6(a), we vary the number of SNs between 1000 and 1600 over a field of 1000 sq.m. By varying the field size between 600 sq.m and 1000 sq.m, we set the sensor nodes to 1000 in Figure 6(b).

Figure 6(a) shows the NSR of the proposed and existing algorithms by varying the 1000 to 1600 SNs in the area of 1000 sq.m. According to the proposed algorithm, the NSR is 96.79% at 1000 nodes and 90.12% at 1600 nodes after 100 cycles. As a result of the existing M2C algorithm, 95.99% of the network is deployed with 1000 SNs after 100 cycles, and 89.11% when 1600 SNs are deployed. When 1000 SNs are deployed in the network at the 100th cycle, SPSS and J-RCA result in 85.37 percent and 84.67 percent, respectively. A 100th round deployment of 1600 SNs results in 78.91% and 78.01%, respectively, for the SPSS and J-RCA algorithms. On the basis of these values, the proposed experiments. All the SN’s battery is full initially, and it is approximately 8 kJ. In case below 10% of the battery is available, it is assumed that it takes 50 min to recharge completely. The velocity of CV is 1 m/s, and approximately, it consumes 55 W/s during movement or standalone.
algorithm appears to have a higher NSR than the existing algorithms. As a result of better scheduling by the network, this achievement has been achieved.

From Figure 6(b), we observe an NSR when the network size is expanded from 600 square meters to 1000 square meters with 1000 static SNs deployed randomly. The proposed algorithm also results in better NSR than existing algorithms, but the area size also decreases the NSR. After the 100th cycle of MLPCV, the NSR is changed from 97.85% to 91.12% in the small area when the area size is doubled. According to M2C, SPSS, and J-RCA algorithms, 80.61 percent, 84.85 percent, and 82.81 percent were obtained in the small area with 100 completed cycles, respectively. M2C, SPSS, and J-RCA algorithms produce similar NSRs after 100 cycles in the 1000 sq.m. size, as well as 82.82%, 80.87%, and 77.53%.

5.5. Charger Utility Efficiency. As a result of taking into account the CV’s total energy usage during travel, data collection, and charging, the charger utility efficiency is calculated.

The CUE of the proposed and existing works is compared by varying the sensor node count and area size in Figures 7(a) and 7(b), respectively. We take the SNs in 1000 square meters based on Figure 7(a). A total of 100 cycles have been completed by both the CV and the
estimated CUE. Data and charging requests also increased as the number of nodes increased. Comparatively to the existing algorithms, the CV uses 186.23 kJ after the 100 cycles are completed for 1600 SNs. Unlike other existing algorithms, J-RCA drains completely before reaching the 100th cycle, whereas M2C uses 192.7 kJ, SPSS uses 199.31 kJ, etc. In order to recharge all the requests from SPSS and J-RCA algorithms, a large-capacity mobile charger was required. There is also the possibility of partitioning the network into multiple parts and scheduling more CVs to recharge and collect data. This task can be completed with a single CV and served in a timely manner with the proposed work.

We randomly deployed 1000 static SNs in a network with a network size varying from 600 square meters to 1000 square meters as shown in Figure 7(b). In this case, 100 cycles were completed by the CV. Figure 7(b) shows that with a larger area, the CUE is minimized as the network is extended to 1000 sq.m. As a result, the CV has to travel most of the time in order to charge. It is necessary for the CV to travel between various network parts to receive charging requests. In the existing algorithm, the 100th cycle of M2C consumes 196.16 kJ. Before assembling the 94th and 87th cycles, SPSS and J-RCA exhaust their energy completely. The proposed work consumes less energy than these...
algorithms, and when it reaches its 100th cycle, it consumes 192.22 kJ. However, it is much better than the existing algorithms but weaker than a small number of sensor nodes.

5.6. Average Service Time. In the process of charging and collecting data, the CV period from the base station is taken and returned to the station after the charging and data collection process is completed. Travel time, waiting time, and data collection time are all included in the service time. All nodes in a network are averaged to determine the AST. Indicators are more efficient when there are fewer of them, and vice versa.

In Figure 8, we compare the AST of the proposed MLPCV to the AST of existing algorithms by varying the number of sensors and the size of the field. Figure 8(a) shows the estimated AST for the proposed and existing works, ranging from 1000 to 1600 sensor nodes in 600 square meters. The proposed work involves ASTs during 1600 SNs in a 1000 sq.m area noticed 49.68 minutes after the 100th occurrence. M2C, SPSS, and J-RCA receive 57.56 minutes, 66.74 minutes, and 72.47 minutes, respectively, in the existing algorithms. Comparing the proposed MLPCV to other algorithms, we can see that it serves very quickly. Optimizing the scheduling charger vehicles efficiently for data collection and recharge is the main reason for this improvement.

In Figure 8, we compare the AST of the proposed MLPCV to the AST of existing algorithms by varying the number of sensors and the size of the field. Figure 8(a) shows the estimated AST for the proposed and existing works, ranging from 1000 to 1600 sensor nodes in 600 square meters. The proposed work involves ASTs during 1600 SNs in a 1000 sq.m area noticed 49.68 minutes after the 100th occurrence. M2C, SPSS, and J-RCA receive 57.56 minutes, 66.74 minutes, and 72.47 minutes, respectively, in the existing algorithms. Comparing the proposed MLPCV to other algorithms, we can see that it serves very quickly. Optimizing the scheduling charger vehicles efficiently for data collection and recharge is the main reason for this improvement.

As shown in Figure 8(b), we notice the AST because we randomly deploy 1600 static SNs rather than 600 sq.m. With more sensor nodes that are randomly deployed, the communication burden is reduced on the SNs than with densely deployed sensors. So by doing this, there will be a reduction in the number of charging requests, and efficiency will be improved. There will not be much difference in AST between small and large fields throughout the system. Our AST changed from 600 square meters to 1000 square meters with 1000 sensor nodes. The ASTs in the proposed work when the area is 600 square meters and 1000 square meters are 28.51 minutes and 23.43 minutes, respectively. Similarly, in 600 square meters, according to the existing algorithms M2C, SPSS, and J-RCA, 1000 SNs are processed in 28.75 minutes, 31.15 minutes, and 36.19 minutes, respectively. The M2C, SPSS, and J-RCA algorithms return AST when the area reaches 1000 square meters in 29.24, 36.54, and 39.142 minutes, respectively.

5.7. Impact on Link Failure. NSR shows that some SNs in the network are dying as a result of a lack of charging services. Additionally, some sensor nodes may be isolated as a result of the effect on the network link. To evaluate the performance of the existing and proposed work, it is necessary to evaluate this metric. For sensor nodes that fail existing links, the proposed work can identify another relay node. The reclustering mechanism is also present in M2C, which performs the mechanism by which relay nodes are assigned to sensor nodes whose links have failed due to a failure of the link.

Based on the link failure, Figure 9 shows the number of disconnected or isolated nodes as well as the number of partitions caused by the failure. When the partition number is greater than 1, it means that one network partition does not reach the base station, but another part of the network does. From Figure 9(a), we notice the minimum number of disconnected nodes in the network. The performance of the M2C algorithm is the most problematic until a few hours into the data collection, after which SPSS and J-RCA algorithms perform poorly. As a result, the J-RCA algorithm isolates more nodes, although it takes longer to simulate. As well as evaluating the partition count, we also estimate how many partitions are created when separate nodes are present. Yet, the proposed MLPCV does not divide the network into more parts, while isolated nodes form the same groups. By doing so, the base station is able to connect more
SNs. The most accurate results are often obtained through the J-RCA. The number of partitions also increases along with the simulation time, as shown in Figure 9(b). Therefore, compared with other mechanisms, the proposed work maintains a higher level of quality.

### 6. Conclusion

A mixed linear programming approach is proposed to schedule the charging vehicles in a large-scale WSN to recharge the sensor nodes. The approach is divided into two parts: weights for each SN to determine the charging device and order and scheduling strategy for the order in which charging requests will be served. Initially, all the CVs are placed in a particular location in the network, which is decided using an intelligent centroid selection approach. Calculate the weight using heuristic formulas and select a schedule based on mixed linear programming. This approach is computationally efficient compared with the existing techniques. Regarding different metrics, the MLPCV approach is superior to the recently published CV scheduling approaches. However, this work achieved outstanding performance, but much more needs to be done in the near future. This is a challenging issue that can be

---

**Figure 8:** Average service time: (a) number of sensor nodes; (b) area size.

**Figure 9:** Impact on link failure: (a) simulation time vs. isolated nodes; (b) simulation time vs. number of partitions.
addressed in the future. Specifically, we can select the optimal number of charging vehicles for a given network according to the number of nodes, battery type, and data generated.

Data Availability

All data generated or analyzed during this study are generated randomly during the simulation. The details about data generation are included in this published article.

Conflicts of Interest

There are no potential conflicts of interest.

References

[1] D. K. Sah and T. Amgoth, “Renewable energy harvesting schemes in wireless sensor networks: a survey,” Information Fusion, vol. 63, pp. 223–247, 2020.

[2] P. K. R. Maddikunta, S. Hakak, M. Alazab et al., “Unmanned aerial vehicles in smart agriculture: applications, requirements, and challenges,” IEEE Sensors Journal, vol. 21, no. 16, pp. 17608–17619, 2021.

[3] P. Joshi, A. S. Raghuvanshi, and S. Kumar, “An intelligent delay efficient data aggregation scheduling for distributed sensor networks,” Microprocessors and Microsystems, vol. 93, p. 104608, 2022.

[4] D. Praveen Kumar, A. Tarachand, and A. C. S. Rao, “Machine learning algorithms for wireless sensor networks: a survey,” Information Fusion, vol. 49, pp. 1–25, 2019.

[5] X. Lan, Y. Zhang, L. Cai, and Q. Chen, “Adaptive transmission design for rechargeable wireless sensor network with a mobile sink,” IEEE Internet of Things Journal, vol. 7, no. 9, pp. 9011–9025, 2020.

[6] E. Erdemir and T. E. Tuncer, “Path planning for mobile-anchor based wireless sensor network localization: static and dynamic schemes,” Ad Hoc Networks, vol. 77, pp. 1–10, 2018.

[7] C. Bala Subramanian, M. Maragatharajan, and S. P. Balakanan, “Retracted article: inventive approach of path planning mechanism for mobile anchors in wsn,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 3, pp. 3959–3967, 2021.

[8] P. Thilagavathi and J. Manickam, “Retracted article: Ertc: an enhanced rssi based tree climbing mechanism for well-planned path localization in wsn using the virtual force of mobile anchor node,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 6, pp. 6665–6676, 2021.

[9] H. Patel, D. Singh Rajput, G. Thippa Reddy, C. Iwendi, A. Kashif Bashir, and O. Jo, “A review on classification of imbalanced data for wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 16, no. 4, Article ID 1550147720916404, 2020.

[10] P. K. R. Maddikunta, Q.-V. Pham, D. C. Nguyen et al., “Incentive techniques for the internet of things: a survey,” Journal of Network and Computer Applications, vol. 206, article 103464, 2022.

[11] M. Zhang and W. Cai, “Data collecting and energy charging oriented mobile path design for rechargeable wireless sensor networks,” Journal of Sensors, vol. 2022, Article ID 5004507, 14 pages, 2022.

[12] M. Majid, S. Habib, A. R. Javed et al., “Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: a systematic literature review,” Sensors, vol. 22, no. 6, p. 2087, 2022.

[13] M. Numan, F. Subhan, W. Z. Khan et al., “A systematic review on clone node detection in static wireless sensor networks,” IEEE Access, vol. 8, pp. 65450–65461, 2020.

[14] H. Xue, H. Chen, Q. Dai, K. Lin, J. Li, and Z. Li, “CSCT: charging scheduling for maximizing coverage of targets in WSNs,” IEEE Transactions on Computational Social Systems, pp. 1–11, 2022.

[15] A. Tomar, K. Nitesh, and P. K. Jana, “An efficient scheme for trajectory design of mobile chargers in wireless sensor networks,” Wireless Networks, vol. 26, no. 2, pp. 897–912, 2020.

[16] M. Srinivas, P. K. Dona, and T. Amgoth, “Mobile charger utility maximization through preemptive scheduling for rechargeable wsn,” in 2021 19th OITS International Conference on Information Technology (OICT), pp. 126–131, Bhubaneswar, India, December 2021.

[17] G. Jiang, S.-K. Lam, Y. Sun, L. Tu, and J. Wu, “Joint charging tour planning and depot positioning for wireless sensor networks using mobile chargers,” IEEE/ACM Transactions on Networking, vol. 25, no. 4, pp. 2250–2266, 2017.

[18] S. Priyadarshani, A. Tomar, and P. K. Jana, “An efficient partial charging scheme using multiple mobile chargers in wireless rechargeable sensor networks,” Ad Hoc Networks, vol. 113, article 102407, 2021.

[19] W. Xu, W. Liang, X. Jia, H. Kan, Y. Xu, and X. Zhang, “Minimizing the maximum charging delay of multiple mobile chargers under the multi-node energy charging scheme,” IEEE Transactions on Mobile Computing, vol. 20, no. 5, pp. 1846–1861, 2020.

[20] C. Lee, W. Na, G. Jang, C. Lee, and S. Cho, “Energy-efficient and delay-minimizing charging method with a multiple directional mobile charger,” IEEE Internet of Things Journal, vol. 8, no. 10, pp. 8291–8303, 2020.

[21] A. Madhja, S. Nikoletseas, and T. P. Raptis, “Hierarchical, collaborative wireless energy transfer in sensor networks with multiple mobile chargers,” Computer Networks, vol. 97, pp. 98–112, 2016.

[22] C. Moraes, S. Myung, S. Lee, and D. Har, “Distributed sensor nodes charged by mobile charger with directional antenna and by energy trading for balancing,” Sensors, vol. 17, no. 12, p. 122, 2017.

[23] Y.-C. Wang and J.-W. Huang, “Efficient dispatch of mobile sensors in a wsn with wireless chargers,” Pervasive and Mobile Computing, vol. 51, pp. 104–120, 2018.

[24] S. P. R. Banoth, P. K. Donta, and T. Amgoth, “Dynamic mobile charger scheduling with partial charging strategy for wsn’s using deep-q-networks,” Neural Computing and Applications, vol. 33, no. 22, pp. 15267–15279, 2021.

[25] W. Ouyang, X. Liu, M. S. Obaidat et al., “Utility-aware charging scheduling for multiple mobile chargers in large-scale wireless rechargeable sensor networks,” IEEE Transactions on Sustainable Computing, vol. 6, no. 4, pp. 679–690, 2021.

[26] W. Na, J. Park, C. Lee, K. Park, J. Kim, and S. Cho, “Energy-efficient mobile charging for wireless power transfer in internet of things networks,” IEEE Internet of Things Journal, vol. 5, no. 1, pp. 79–92, 2017.

[27] T. Rault, “Avoiding radiation of on-demand multi-node energy charging with multiple mobile chargers,” Computer Communications, vol. 134, pp. 42–51, 2019.
Y. Feng, L. Guo, X. Fu, and N. Liu, “Efficient mobile energy replenishment scheme based on hybrid mode for wireless rechargeable sensor networks,” *IEEE Sensors Journal*, vol. 19, no. 21, pp. 10 131–10 143, 2019.

H. Yu, C.-Y. Chang, Y. Wang, D. S. Roy, and X. Bai, “Caerm: coverage aware energy replenishment mechanism using mobile charger in wireless sensor networks,” *IEEE Sensors Journal*, vol. 21, no. 20, pp. 23 682–23 697, 2021.

G. K. Ijemaru, K. L.-M. Ang, and J. K. P. Seng, "Swarm intelligence techniques for mobile wireless charging,” *Electronics*, vol. 11, no. 3, p. 371, 2022.

Y. Zhu, K. Chi, P. Hu, K. Mao, and Q. Shao, “Velocity control of multiple mobile chargers over moving trajectories in rf energy harvesting wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 314–11 318, 2018.

L. Mo, A. Kritikakou, and S. He, "Energy-aware multiple mobile chargers coordination for wireless rechargeable sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8202–8214, 2019.

Y. Hong, C. Luo, D. Li, Z. Chen, X. Wang, and X. Li, "Energy efficiency optimization for multiple chargers in wireless rechargeable sensor networks," *Theoretical Computer Science*, vol. 922, pp. 193–205, 2022.

A. M. Alabdali, N. Gharaei, and A. A. Mashat, "A framework for energy-efficient clustering with utilizing wireless energy balancer," *IEEE Access*, vol. 9, pp. 11 7823–11 7831, 2021.

T. N. Nguyen, B.-H. Liu, S.-I. Chu, D.-T. Do, and T. D. Nguyen, "Wrns: toward an efficient scheduling for mobile chargers," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6753–6761, 2020.

V. K. Chawra and G. P. Gupta, "Hybrid meta-heuristic techniques based efficient charging scheduling scheme for multiple mobile wireless chargers based wireless rechargeable sensor networks,” *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1303–1315, 2021.

D. K. Sah, K. Cengiz, P. K. Donta, V. N. Inukollu, and T. Amgoth, "EDGF: empirical dataset generation framework for wireless sensor networks,” *Computer Communications*, vol. 180, pp. 48–56, 2021.

A. Boukerche, Q. Wu, and P. Sun, "A novel joint optimization method based on mobile data collection for wireless rechargeable sensor networks,” *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1610–1622, 2021.

Y. Aamer, Y. Benkaouz, M. Ouzzif, and K. Bouragba, "Initial centroid selection method for an enhanced kmeans clustering algorithm,” in *Ubiquitous Networking*, O. Habachi, V. Meghdadi, E. Sabir, and J.-P. Cances, Eds., pp. 182–190, Springer International Publishing, Cham, 2020.

A. Hazra, P. K. Donta, T. Amgoth, and S. Dustdar, "Cooperative transmission scheduling and computation offloading with collaboration of fog and cloud for industrial iot applications,” *IEEE Internet of Things Journal*, 2022.

P. K. Donta, S. N. Srirama, T. Amgoth, and C. S. R. Annavarapu, "Survey on recent advances in iot application layer protocols and machine learning scope for research directions,” *Digital Communications and Networks*, 2022.