EAI Endorsed Transactions

on Scalable Information Systems Research Article **EAI.EU**

Node-Alive Index Driven Redundancy Elimination for Energy-Efficient Wireless Sensor Networks

Archana Bomnale[1,*] and Avinash More[1]

[1]Department of Electronics and Telecommunication Engineering, MPSTME, NMIMS University, Mumbai, 400056, India

Abstract

Wireless Sensor Networks (WSNs) generate correlated and redundant data. This redundancy increases energy consumption during transmission and aggregation, which reduces the network lifespan. Eliminating data redundancy using appropriate data aggregation mechanisms in the dynamic environment is challenging. To address these issues, we designed the Data Aggregation with Redundancy Removal (DARR) protocol and implemented it in two phases. In Phase I, the DARR protocol identifies redundant nodes by calculating the spatial distance between the adjacent nodes. Over time, nodes may run out of energy and stop working after continuously sensing, aggregating, and transmitting the data. The dead nodes can obstruct data forwarding to intermediate nodes, so it is important to check periodically whether the nodes are alive or dead. The periodic time check identifies the status of each node, allowing the protocol to focus only on active nodes. It sets redundant nodes to sleep, which conserves network energy. In Phase II, the protocol reduces data redundancy at the source nodes using temporal correlation between data measurements. We enhanced the DARR protocol by incorporating a High Compression Temporal (HCT) mechanism, which further reduces data redundancy. Simulations show that the DARR protocol reduces data transmissions by 24% and lowers network energy consumption by up to 31% by eliminating redundant data at both the network and node levels.

Keywords: Node-Alive Index, Data Aggregation, Energy Efficiency, Redundant Data, Temporal Correlation, Wireless Sensor Network.

Received on 26 09 2024, accepted on 05 12 2024, published on 06 12 2024

doi: 10.4108/eetsis.7397

1. Introduction

The wireless sensor network (WSN) has become more popular to facilitate real-time applications by interacting sensor nodes with physical objects via networked technology, similar to the Internet of Things (IoT). Wireless Sensor Networks (WSNs) face numerous challenges which affect their performance, reliability, and efficiency. Battery power is a common source of power for sensor nodes, however it is limited. Energy limits can significantly impact the sensor network lifetime. Hence, energy-efficient protocols and algorithms are crucial [1]. Batteries often power the sensor nodes (SNs) in a harsh environment with a non-replaceable power supply, and they have limited sensing, transmission, and computing capabilities, so optimal node scheduling plays an important role in energy-efficient network [2,3]. Also, the sensor network is more vulnerable to failure since sensing, communication, and data processing consume the most energy. Many long-term continuous monitoring applications, such as tracking forest fires, habitats, automobile automation, military surveillance, industrial control systems, traffic monitoring, etc., await the sensor nodes' information while running unsupervised environments. Therefore, energy

[*]Corresponding author. Email: archana.bomnale@nmims.edu

management is essential for extending a wireless sensor network lifetime.

In the WSN protocol stack, data aggregation plays a crucial role in saving the network energy. When aggregating the data, all network information must be efficiently transmitted to the sink node [4, 5, 6]. WSNs can enable effective data transmission, lower energy consumption, and extend the network lifetime by utilizing spatial and temporal correlation among the data measurements. The accuracy of representative data achieved through spatial correlation-based methods is more significant compared to other correlation models [6]. An Efficient Data Collection Aware of Spatio-Temporal Correlation (EAST) protocol [7] dynamically modifies the size of the correlation region. It also modifies the value of coherent threshold based on the event characteristics. This helps to reduce the amount of energy utilized in data aggregation as it maintains both data accuracy and real-time reporting. To reduce the network energy consumption and increase the network lifetime further, we have proposed a data aggregation protocol.

The proposed protocol uses an algorithm to identify redundant nodes before data transmission and puts them into Sleep mode. As a result, only a few non-redundant nodes transmit data. In contrast, the existing methods in the literature [11, 12] allow all source nodes, both redundant and non-redundant, to transmit data to the sink node. By reducing the number of data transfers within the network, our protocol significantly conserves energy and prevents network depletion.

1.1. Motivation

Data aggregation in wireless sensor networks (WSNs) is a key concept that aims to improve the network efficiency, longevity, and data accuracy. The motivation for data aggregation in Wireless Sensor Networks (WSNs) stems largely from the need to increase energy efficiency. Since sensor nodes in WSNs are usually battery-operated and placed in inaccessible or remote areas, it might be challenging to repair or recharge their power supplies. Large amounts of raw data transmission can quickly deplete a sensor node's battery since data transmission uses more energy than local data processing. By enabling nodes to analyse data locally, data aggregation lessens the volume of data transmitted over the network. The longevity of sensor nodes is greatly increased by this energy conservation through reduced data transmission, which also prolongs the network overall operating life.

Furthermore, as WSNs frequently have limited bandwidth resources, data aggregation aids in lowering bandwidth consumption. By sending only the most important, aggregated data, network congestion is decreased and the network can function more effectively. In addition to saving bandwidth, this decrease in network load also reduces interference between sensor nodes. A more reliable communication process results from fewer transmissions since there is a lower chance of data packet collisions and subsequent retransmissions. As a result, the network is more capable of managing high data volumes and sending the most pertinent information to the user.

1.2. Preliminaries

Over time, nodes may exhaust their energy due to continuous sensing, aggregating, and transmitting of data, eventually leading to their failure. These dead nodes can disrupt data transmission by hindering communication with intermediate nodes. To prevent this, we periodically check the status of each node to determine whether it is alive or dead. The terms used in our research to describe node status are defined as follows.

- **Node-Alive Index:** The Node-Alive Index represents the percentage ratio of the number of nodes alive at present to the number of nodes alive at the previous network reconfiguration instant.
- **Node-Alive Check Time:** The interval at which the node's status (alive or dead) is monitored.
- **Node-Alive Threshold:** It is a constant value set by the user. If the Node-Alive threshold is low, the network is re-configured fewer times.

1.3. Contribution

We have developed a novel data aggregation protocol that identifies redundant nodes at the network level and minimizes redundant data at the node level. The protocol then transmits the aggregated data efficiently to the sink node. The key contributions of this research are highlighted as follows.

- We have identified the adjacent nodes and determined the redundant nodes based on the spatial distance technique and redundancy threshold.
- The computed redundant nodes enter into a sleep state, and only active nodes are allowed to transmit the data to the sink node to enhance the network lifetime.
- We computed the Node-Alive Index and optimized the Node-Alive threshold to minimize transmission overhead and data loss. If the Node-Alive Index falls below the defined threshold, our protocol stops data transmission, reconfigures the network, and then resumes data transmission to avoid energy wastage.
- We have developed an aggregation method to remove redundant data from source nodes based on temporal correlation. We computed data redundancy using a similarity threshold, which depends on the difference between consecutive data measurements. Further, we assign a weight to the non-redundant data measurement.

- We have modified the DARR protocol and proposed the HCT (High Compression Temporal) mechanism to reduce redundant data measurements further.

The remaining sections of our paper are organized as follows. In Section 2, we have elaborated on the related work in the current literature. We discuss the design and implementation of the proposed protocol in Section 3. Section 4 presents the evaluation and simulation results. Section 5 provides the concluding remarks and future research direction.

2. Related Works

Advancements in computer technology have made it possible to develop WSNs that can continuously monitor important parameters. Cloud computing can be quite helpful for web applications with particular processing and storage needs. Wireless sensor networks are connected to the cloud, which has a scalable and flexible architecture based on IoT(Internet of Things) [8]. During end-to-end transmission, these systems experience limited bandwidth, power, and resource consumption. Data aggregation [9] is one of the methods discussed for alleviating the above-mentioned problems.

We have referred to the literature survey to discuss various spatial, temporal, scheduling, and routing-based data aggregation algorithms. As the nodes are randomly deployed and dense, they transmit more redundant data. In [10], data redundancy is removed based on the semi-variance-based compressive sensing method. In the data aggregation process, the aim is to eliminate redundant data by removing relevant information from the collected data and sending it to the sink node. The most commonly used data aggregation functions are maximum (MAX) and minimum (MIN). Also functions such as MEAN, MEDIAN, Average (Avg), Difference (Diff), Count, and other math functions like MATCH, and Correlation-Coefficient techniques [11, 12, 13] are often used. Three major categories, centralized, distributed, and hybrid algorithms, are used to group data aggregation methods. As part of the centralized algorithm, a central node known as a cluster head collects information on each node in the network and keeps track of every other node [14, 15]. Additionally, it keeps the same information in its database. Any node must ask the central node for permission to interact with any other node. Whereas in the distributed algorithm [16], the central node need not be identified and any node may interact with another node. In this case, each node contributes equally to the packet routing process. Data aggregation systems must use big data handling techniques when dealing with massive volumes of data in IoT-based applications using a distributed approach [17].

In the modified I-LEACH (internet-based low-energy adaptive clustering hierarchy) protocol [18], a distributed cluster formation approach is used. The existing clustering method, LEACH, is adjusted by introducing a threshold value for selecting the cluster head (CH). Additionally, the power levels of nodes are adjusted at the same time. The amount of sensed data conveyed from the source nodes to the sink node is substantially reduced using a hybrid data aggregation strategy. The hybrid approach integrates the features of both the cluster and tree-based approaches, as mentioned in [19].

The energy efficiency of WSNs can be further improved by utilizing the spatial correlation among nodes. This is achieved by implementing a data collection strategy proposed in [20]. The strategy uses length-compressed coding to eliminate the unnecessary transmission of redundant data.

In the existing algorithms [11, 12, 21], the data from all the nodes in a cluster is sent to the cluster head (CH). At the CH, redundant data is eliminated using spatial correlation. There is a significant limitation in the above algorithms, as all the nodes participate in data transmission to the CH or aggregator node. Additionally, the redundancy elimination techniques may cause the system to become more complex. The algorithms must be practical, and implementing them could require additional memory and computing power. In WSN, nodes with limited resources, this additional complexity may pose difficulties and affect the network performance.

For various reasons, the collected data comprises large redundant data. Since redundant data is utilized to increase sensed data accuracy and reliability and build a fault-tolerant network, it is desirable due to its favourable effects. However, it causes various problems, such as reducing the network lifetime, data processing costs, bottlenecks, throughput delays, network congestion, and contention. Aggressive redundancy removal approaches may sacrifice data accuracy for efficiency. In [21], a spatial redundancy reduction approach for IoT data is presented to reduce redundant data issues without compromising the data's accuracy and reliability. The Sink Level Aggregation Algorithm (SLAA) and Sink Level Grouping Algorithm (SLGA) provided in [22] detect the deviation in data that implements the Kalman filter. The Kalman filter eliminates the spatial and temporal redundancy in the data. As the sensor networks frequently have changing topologies, the performance of the Kalman filter may decline. Utilizing support vector machines (SVM), using a supervised learning model [23] is another method to remove redundancies during data aggregation. To minimize the quantity of data transferred across the network further, temporal correlation can be used for the data aggregation process of combining or summarising redundant or linked information [24]. There are issues with collisions during data transmission as well as synchronization between the sensor nodes and their neighbours.

The specific requirements of the WSN and the application's specifications determine the choice of Medium Access Control (MAC) protocol. For example, Time Division Multiple Access (TDMA) may be suitable for certain scenarios. In other cases, Carrier Sense Multiple Access (CSMA) might be a better choice. The selected MAC protocol should work effectively with temporal correlation techniques to optimize network performance.

The lifetime of Wireless sensor networks (WSNs) depends heavily on latency, especially in applications where real-time data is critical. As the processing power of sensor nodes is often constrained, reducing the latency to reduce the node's computational power is a challenge. To aggregate data measurements from other sensor nodes (SNs), each SN must wait a predetermined amount of time, known as the waiting time (WT), which can be either fixed or variable, before executing the aggregation function [25]. This creates a delay in the data transmission in the network. To minimize this delay, the Distributed algorithm for Integrated tree Construction and data Aggregation (DICA) [26] is used. This approach combines node scheduling and tree formation to minimize delay. Another popular method for extending a sensor network lifetime is Duty cycling, however, it increases the latency in data aggregation [27]. In [28], a collision-resistant dynamic (CORD) scheduling approach is proposed. This suggested approach adapts to any starting routing structure and dynamically modifies a transmission's receiver, and whenever doing so, can shorten the aggregation time. Minimum Latency Aggregation Scheduling (MLAS) [29] computes a conflict-free schedule and constructs an aggregation tree with the chosen nodes, relying on non-predetermined structures. To reduce the latency, a contention-free approach is employed as a solution in the media access control technique, which uses TDMA to allocate time slots across SN on the assumption that another service will maintain network synchronization [30].

To prolong the network lifetime, the majority of protocols in the literature either implement routing protocol or eliminate redundancy when aggregating the data. However, in [31], a novel Q-learning-based Data-Aggregation-aware Energy-Efficient Routing (QADEER) method is implemented to achieve data aggregation with routing and redundancy elimination. The approach uses reinforcement learning to find the best path for data transmission. It considers factors like communication energy and the residual energy of each node. It also evaluates the effectiveness of data aggregation based on the type of sensor in use. In another routing technique, the authors suggested a novel routing method for WSN called Efficient Bandwidth Aware Routing Protocol (EBARP) [32] to improve sensor node energy. It involves fuzzy logic to form the cluster and select the cluster head (CH) based on residual energy, bandwidth, and delay. One of the most critical challenges in transmitting the redundant data to the sink node is the formation of energy holes near the sink node, which cause energy imbalance in the network. Using the A-star heuristic technique, the LSDAR (Lightweight structure-based data aggregation routing) [33] protocol controls the energy hole problem. The Node utilization-based data routing and aggregation (NUIDRA) protocol [34] addresses the energy hole issue by relying on each node's bandwidth utilization. To further improve communication efficiency and reliability, the O-AODV (Optimized AODV) protocol for Bluetooth Low-Energy (BLE) uses a multi-hop mesh with a flat, non-hierarchical topology. This approach offers advantages over the traditional AODV (Ad hoc On-Demand Distance Vector) and standard mesh routing protocols.

As the network expands, basic aggregation and routing methods may struggle to scale effectively, resulting in increased delays, congestion, and energy consumption. To address these issues, modern data aggregation and routing algorithms often incorporate AI and ML techniques. Fuzzy logic systems have been used to decide optimal routes in real-time while considering energy and delay trade-offs.

Real-time data collection from the surrounding environment is one of the primary characteristics of IoT-based WSNs. Sensor nodes quickly provide data across the network while continuously monitoring physical attributes like temperature, humidity, and motion. As the need for routing techniques goes beyond optimisation, routing protocols themselves must be intelligent. Intelligent routing systems employ machine learning algorithms and adaptive decision-making to effectively negotiate the complexity of dynamic WSNs [37].

When processing large amounts of data in WSNs, Artificial Neural Networks (ANNs) offer a more efficient solution. The ANN has been applied in WSNs to solve various issues like routing, node localization, data aggregation, congestion control, etc. [38]. Also, [39, 40, 41] mentioned about the latest trends in AI, ML, Deep Learning, big data and Cognitive Learning in data aggregation mechanism. The knowledge tracing model (XKT) could be adapted to enhance the data aggregation in wireless sensor networks (WSNs) [42]. The XKT can apply its principles of multi-feature embedding, cognitive processing, and predictive modelling to manage and interpret the data aggregated from various sensor nodes.

The convergence of 5G technology and WSNs offers immense potential for enabling real-time, high-speed communication with ultra-low latency and massive device connectivity. This integration enhances the capabilities of WSNs in applications like IoT, smart cities, and industrial automation by supporting intelligent data routing, adaptive resource allocation, and energy-efficient operations [43]. In the next section, we discuss the design and implementation of the proposed DARR protocol.

3. Proposed DARR Protocol - Design and Implementation

The proposed Data Aggregation with Redundancy Removal (DARR) protocol is explained in detail in this section. We suggest a two-phase design and implementation process for our protocol. The working mechanism of the proposed protocol is shown in Figure 1. In the first phase, we determine the adjacent and redundant nodes based on the spatial distance between them. Then we set the redundant nodes to Sleep state. The sleep nodes do not take part in data sensing and transmission so the transmissions from redundant nodes are avoided. This reduces the communication overhead and network energy consumption. In the second phase, we remove the redundant data from source nodes using temporal correlation. This aggregated data is sent to the next active node or intermediate node, and then to the sink node using the shortest path [34]. A detailed explanation of phase I and phase II is given in section 3.3.

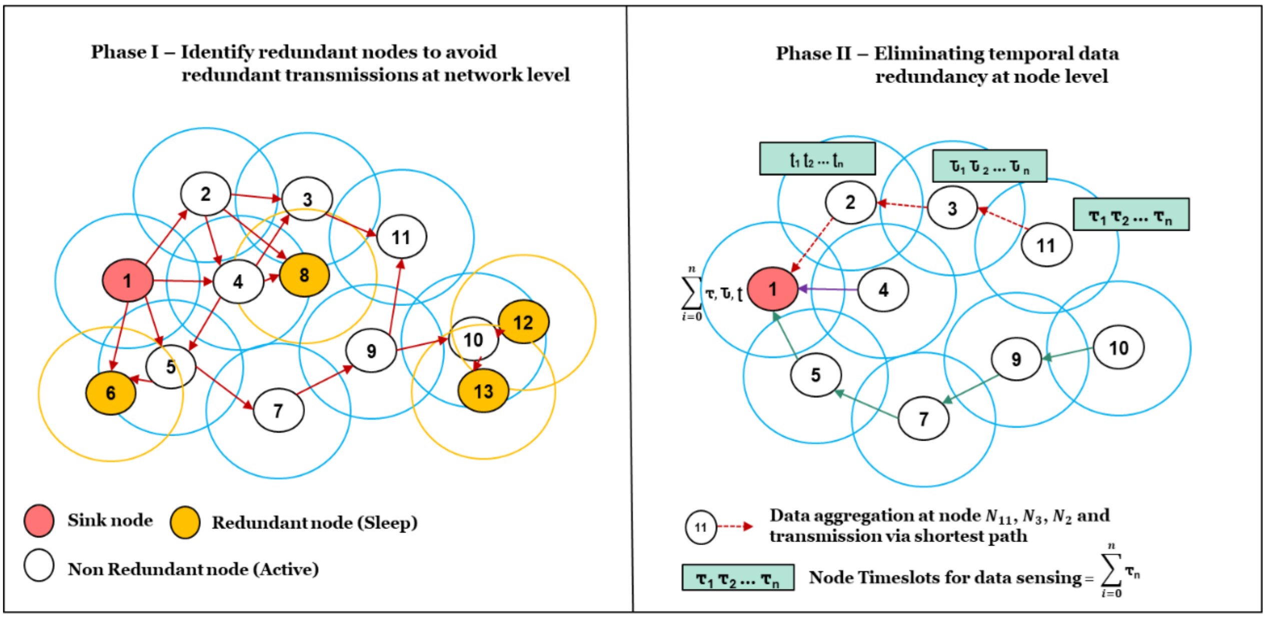

Figure 1. Working mechanism of proposed DARR protocol

3.1. Assumptions for DARR protocol

The design of the DARR protocol is based on the following assumptions:

- The nodes are dispersed uniformly and randomly over the network field.
- The location of the nodes is not known.
- After deployment, nodes are stationary.
- Every node has the same amount of memory, battery capacity, and computational power.
- The node's sensing range (Rs) is equivalent to half of its communication range (Rc).

3.2. Network Operation

Once the network is initialized, the network is operated in the following sequence.

(i) The redundant nodes are set to Sleep state and non-redundant nodes are set to Active state.

(ii) The sensed data is aggregated using temporal correlation at the active source nodes and is send to the sink node using the shortest path.

(iii) Periodically, as per the Node-Alive-Check Time (τ) value, the Node-Alive Index (η) is compared with the Node-Alive Threshold (Γ).

(iv) If Node-Alive Index goes less than Node-Alive threshold value, the sequence of operation is started again from the first step.

Table 1 shows the symbols and notations used in the proposed protocol.

Table 1. Symbols and Notations

Symbols	Description
S_i	i^{th} source node, where i is an integer
N	Number of sensor nodes
R_n	Redundant node
Ad_n	Adjacent node
R_c	Communication range
R_s	Sensing range
d	Spatial distance between adjacent nodes
x	Redundancy threshold
δ	Number of nodes alive at an instant
γ	Number of nodes alive in last iteration
τ	Node-Alive Check Time
η	Node-Alive Index
Γ	Node-Alive threshold
Ʈ	Time slot for source node to sense data
t	Time interval assigned to node to sense data before aggregation
M	Data measurement in $Ʈ_j$ slot, where j is an integer
ε	Similarity measure threshold for data values
μ	Mean value
w	Weight value

Next, we will examine the network operation in detail as outlined below.

3.2.1. Monitoring of Alive Nodes

As the data transmission starts from the source nodes to the adjacent nodes, with time, few nodes may fail due to environmental conditions or due to energy depletion. We do not consider dead nodes as intermediate or aggregator nodes to forward the data. So to consider only the alive nodes, after every Node-Alive Check Time (τ) interval, we check the node status. We can change the value of (τ) depending on the application. Based on the Node-Alive Index (η) value we decide the reconfiguration of the network. We denote the 'number of nodes alive at previous network reconfiguration instant' as γ. The Node-Alive Index (η) is given by Equation 1.

$$\eta = \delta/(\gamma) * 100 \tag{1}$$

For illustration, we consider, there are 100 sensor nodes alive in the network. After Node-Alive Check Time (10 minutes), 93 nodes are alive, and 7 nodes are dead, so according to Equation 1 the value of the Node-Alive Index (η) is 93. The Node-Alive threshold (Γ) value is set to 95. Now again after the next Node-Alive Check Time interval (10 minutes), the Node-alive Index is computed. Whenever the Node-Alive Index is less than the Node-Alive threshold, the network operation will begin again and will recalculate the redundant nodes in the network, repeating the whole process as mentioned in section 3.2. Once we monitor the network for alive nodes then we identify the adjacent nodes and redundant nodes as discussed un the following section.

3.3. Phase I - Identifying Redundant Nodes

In Phase I of the proposed protocol we identify the redundant nodes in the network. The procedure to find the adjacent and redundant nodes is discussed in the following subsections.

3.3.1. Identifying Adjacent nodes

For each node, the sink node identifies its adjacent nodes depending on the spatial distance (d) between the nodes. The spatial distance between the two nodes is measured by the sink node and is dependent on the Received Signal Strength Indicator (RSSI) readings [33]. When the spatial distance between two nodes is equal to or smaller than the communication range (R_c), then we consider them to be adjacent as mentioned in Equation 2.

$$\text{Adn} = \begin{cases} 1 & \text{if } d_{im} \leq R_c \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where i and m are integers; i represents the current node and m represents an adjacent node.

According to Equation 2, the adjacent-node flag is set to the value one if the distance between two nodes is less than or equal to the communication range otherwise, it is set to the value 0.

3.3.2. Identifying Redundant nodes

Once we determine the adjacent nodes, the proposed protocol determines the redundant nodes (Rn) as given by Equation 3.

$$R_n = \begin{cases} 1 & \text{if } (x * R_c) / d_{im} \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here, x denotes the redundancy threshold whose value varies from 0.1 to 0.9.

For illustration purpose Figure 2 shows a network of twelve sensor nodes. The sink node computes the spatial distance (d) for all the nodes using the received signal strength measurements [33]. We have set the redundancy threshold (x) value as 0.45 and the communication range R_c as 10m. The redundancy threshold (x) is selected depending on the requirement of redundancy percentage, which depends on the application. The sink node identifies its adjacent nodes in the network using Equation 2. Further, the sink node computes redundant nodes using Equation 3 from its adjacent nodes.

The detail computation of redundant nodes is illustrated in Figure 2 (a), (b), (c), and (d) and Table 2.

Step I – The nodes adjacent to the sink node (1) are nodes (2,4,5 and 6) and as shown in Figure 2 (a) and Table 2. Now, the sink considers two adjacent nodes, (1 and 2). The distance between them is 8 meters and by using Equation 3, we find the value of R_n is 0.56, which is not equal to or not greater than 1, and hence the redundant flag is set to 0 so node-1 and node-2 are not redundant as shown in Table 2. Similarly, sink node (1) computes the redundancy with node-4, node-5 and node-6 and finds that they are not redundant with node (1). After step I, we determine the redundant nodes for node-2.

Step II – The adjacent nodes of node-2 are node-3, node-4 and node-8 as shown in Figure 2 (b). Now the sink considers two adjacent nodes, node-2 and node-4. The distance between them is 6 meters. Using Equation 3, we find that R_n is 0.75, which is not equal to or not greater than 1. Thus, node-2 and node-4 are not redundant. Similarly, the sink checks the redundancy with the adjacent node-3 and node-8.

Step III – As shown in Figure 2.c), the adjacent nodes of node 4 are node-3, node-8 and node-5. Now the sink considers two adjacent nodes, node-4 and node-8. The distance between them is 4.5 meters. Using Equation 3, we find that R_n is 1, and hence node-4 and node-8 are redundant.

Step IV – The adjacent nodes of node-5 are node-4, node-6 and node-7. Now the sink considers two adjacent nodes, node-5 and node-6. The distance between them is 4.5 meters. Using Equation 3, we find that R_n is equal to 1 so the redundant flag is set to 1, and

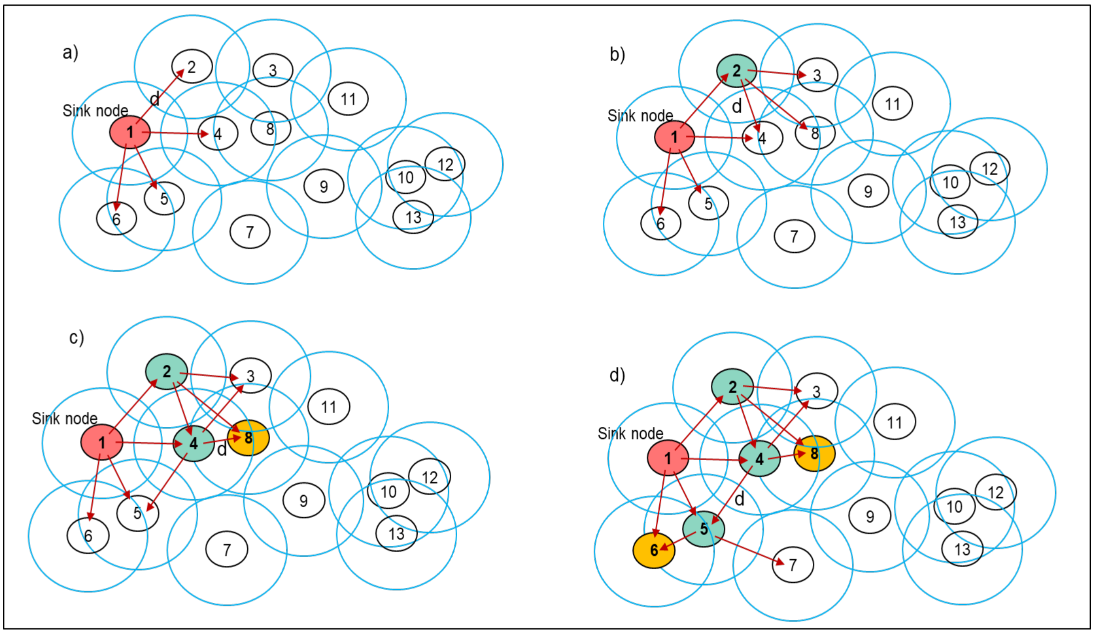

Figure 2. Identification of redundant nodes

Table 2. Computation of redundant nodes

Visited Nodes	Sleep Flag	Adjacent (non-visited) Nodes	Redundant Node-count	Redundant Node-Id
1	0	2, 4, 5, 6	0	-
2	0	3, 4, 8	0	--
4	0	3, 5, 8	1	**8**
5	0	4, 6, 7	1	**6**
6	**1**	--	--	--
3	0	8, 11	0	--
8	**1**	--	--	--
7	0	9	--	--
11	0	9, 10	--	--
9	0	10, 13	--	--
10	0	12, 13	2	**12, 13**
12	**1**	--	--	--
13	**1**	--	--	--

hence node-5 and node-6 are redundant. It checks the redundancy for the remaining adjacent nodes in a similar fashion. The process mentioned in the above steps, I to IV, continues until the sink finds all the adjacent and redundant nodes in the network. The sink also maintains the adjacent-nodes and redundant-nodes table with their node-id as shown in Table 2. The redundant nodes are highlighted (yellow colour) in Figure 3 and refer Table 2 for the same. Node-6, node-8, node-12 and node-13 represent the redundant nodes with respect to node-5, node-4, node-10, and node-10 respectively.

Identifying redundant nodes in the data aggregation process is also presented in Algorithm 1.

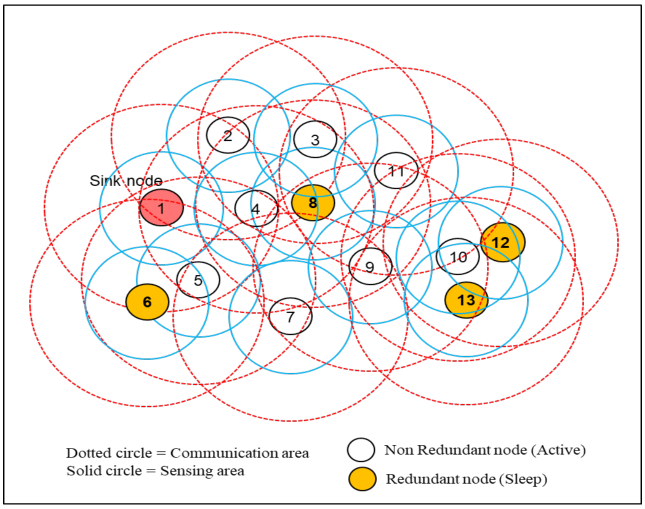

Figure 3. Redundant nodes in a network

Once we identify the redundant nodes, we set them to Sleep state till the Node-Alive Check Time, and the non-redundant nodes are set to Active state for the same period. They will remain in Active or Sleep state depending on the number of alive nodes in the network. The Active node aggregates the data and further transmit it to the adjacent node and finally to the sink node using the shortest path [34].

Algorithm1: Identifying redundant nodes

Input : Node lists/arrays used in algorithm
SrcNodes – Input list of nodes to be processed

Nodes – list to add and process nodes during program execution

Node – variable to process a single node

VisNodes – List of visited and processed nodes during program execution

AdjNodes – List of adjacent nodes found for a current node being processed

RedNodes – List of redundant nodes

Rc – communication range, x - redundancy threshold, d - distance

```
Start
Initialize node lists SrcNodes, Nodes, AdjNodes, VisNodes
Initialize variables d, x = 4.5, Rc = 10
Nodes.AddNode(SrcNodes)          – First SinkNode is added in list Nodes to start processing
Repeat till list Nodes is empty     – process nodes in loop
    Node = Nodes.GetNode()       – Get node from list and assign it for processing
    VisNodes.AddNode(Node)       –  add in visited node list
    If Node.SleepFlag = 1        –  if node is already marked for sleep, proceed to next node
            Next iteration of loop
    End If
    AdjNodes = Node.FindAdjNodes() – Find adjacent nodes for a current node
    Repeat till AdjNodes is empty               – Process all adjacent nodes in loop
      If AdjNodes.Node is not in VisNodes list
        d = FindDistance (Nodes.Node, AdjNodes.Node)      – Find distance between nodes
        AdjNodes.SleepFlag = IdentifyRedundancy (d, Rc, x)    – Identify redundancy
        If AdjNodes.SleepFlag  = 1                  – If another redundant node identified
```

```
            ++VisNodes.node.RedNodesCount               – increase redundant node count for
                                                          the node
            VisNodes.node.RedNodes.add(AdjNodes.Node)   – add redundant node in list
         End if
         Nodes.AddNode(AdjNodes.Node)         – Add adjacent node to nodes list
         AdjNodes = AdjNodes.next          – take next adjacent node for processing
      End If
   End loop
End loop
Stop
```

Function: IdentifyRedundancy (d, Rc, x)

```
Start
   Int SleepFlag
   SleepFlag = Int (Rc*x/d )              – Calculate formula to identify node to mark Sleep flag
   If SleepFlag >= 1
      SleepFlag = 1
   End if
Return SleepFlag
```

In the next section, we discuss the working of Phase II of our protocol.

3.4. Phase II – Data aggregation using temporal correlation

Each source node senses the data during the time interval t and then aggregates it. Each time interval (t) consists of j time slots ($Ʈ_j$) during which a node senses and records the data, as indicated by Equation 4.

$$t = \sum_{j=1}^{n} Ʈ_j \qquad (4)$$

where j is the timeslot number in time interval (t); n is the maximum number of timeslots in time interval (t).

For illustration, as shown in Figure 4, we consider sensor node-1 senses ten events in a single time interval t_1, at various time slots from $Ʈ_1, Ʈ_2, upto\ Ʈ_{10}$. We set the similarity measures threshold (ε) value to 0.05. The sensed data measurements in time slots $Ʈ_1$, $Ʈ_2$ and $Ʈ_3$ is redundant and denoted as Redundant Data Value (RDV). The data measurements in $Ʈ_4$, $Ʈ_5$, and $Ʈ_9$, $Ʈ_{10}$ is also redundant whereas the data in the remaining time slots represent Non-Redundant Data Value (NRDV). After aggregating the data using temporal correlation, from ten data readings, only six are sent to the next intermediate node. Using modified DARR protocol with HCT mechanism only two data measurements are transmitted instead of 10 data measurements as shown in Figure 4 and step-by-step procedure is explained in Table 3.

We use the Cognate function to identify similar data as given by Equation 5 which indicates the difference between two data measurements.

$$\text{Cognate}\ (S_{i,j},\ S_{i,j+1}) = \begin{cases} 1 & \text{if } \|\, S_{i,j},\ S_{i,j+1} \,\| \le \varepsilon \\ 0, & \text{otherwise} \end{cases} \qquad \ldots\ldots\ldots \quad (5)$$

where, S represents the sensor node, and (i,j) are integer variables representing the data of i^{th} node at j^{th} timeslot.

The similarity measures threshold value (ε), depends on the type of WSN application. The Cognate function returns the value "1", which means the data values are redundant, i.e. two data measurements are the similar.

As shown in Table 3, we have set the similarity measures threshold (ε) value to 0.05, and the 'Weight' value of the current data-value or data-measure as 1. When the data is identified as redundant, its weight is incremented by 1. In the first iteration, we consider the first data value sensed at T1 (5.11) with its weight value 1. In the second iteration, we apply the 'Cognate' function for data-value sensed at T1 (5.11) and data-value sensed at T2 (5.09). As per Equation 5, Cognate function returns value 1, that means, data-values 5.11 and 5.09 are redundant. We calculate the 'Mean' of these two data values (5.10) and its weight value now becomes 2. Similarly, we apply the Cognate function for the next successive measurements. i.e. data-value at at T2 (5.09) and at T3 (5.12), which returns the value 1, so data values 5.09 and 5.12 are redundant. Now, we find the 'Mean' of previous redundant values (5.11, 5.09) and the current successive value (5.12) which is 5.107 with its new weight value updated to 3.

If the data values are not redundant (Cognate function returns zero value) then 'Mean' is not computed and the earlier Mean

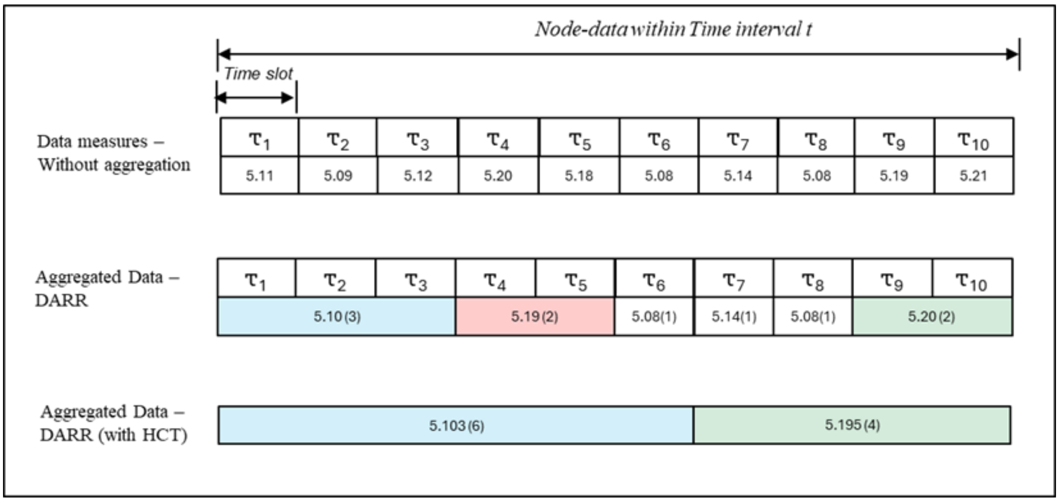

Figure 4. Data aggregation using temporal correlation

value and the current data-value are retained. The final mean values at the 10th iteration are the aggregated data values with their weight which are reduced from 10 data-values to 6 non-redundant data-values. Weight value represents the occurrence of similar or redundant data value. The removal of redundant data using temporal correlation in the data aggregation process is explained using Algorithm 2.

Table 3. Removal of redundant data using temporal correlation

Steps	Data-values $S_{i,j}$	DARR (Compared with consecutive data values)		
		Similarity measures threshold (ε) = 0.05		
		Cognate ($S_{i,j}$, $S_{i,j+1}$)	μ	(μ, w)
1	5.11			(5.11, 1)
2	5.09	1	5.10	(5.10, 2)
3	5.12	1	5.107	(5.107, 3)
4	5.20	0		(5.107, 3) (5.20, 1)
5	5.18	1	5.19	(5.107, 3) (5.19, 2)
6	5.08	0		(5.107, 3) (5.19, 2) (5.08, 1)
7	5.14	0		(5.107, 3) (5.19, 2) (5.08, 1) (5.14, 1)
8	5.08	0		(5.107, 3) (5.19, 2) (5.08, 1) (5.14, 1) (5.08, 1)
9	5.19	0		(5.107, 3) (5.19, 2) (5.08, 1) (5.14, 1) (5.08, 1) (5.19, 1)
10	5.21	1	5.20	(5.107, 3) (5.19, 2) (5.08, 1) (5.14, 1) (5.08, 1) (5.20, 2)

Algorithm 2 : Data Aggregation using Temporal Correlation

Input : *DataA* (data array contains j number of time slots with sensed data values), *MeanA* (array to store mean values), *WeightA* (to store weight for each mean value), *TempA* (array to keep data values to calculate mean)
1: **Start**
2: Initialize array *MeanA*, *WeightA*, *TempA* and integers n=0, k=0, l=0
3: ε = 0.05 – Constant for similarity measures threshold
4: $TempA_l = DataA_n$, $MeanA_k = DataA_n$, ++n – Assign first data value
5: **Repeat** from 2nd value till jth value – Last value in *Data A*

6: **If** difference ($DataA_n$, $DataA_{n-1}$) > ε – If difference between current and previous value is more than ε
7: l=0, ++k – Initialize temp array l, increase array iterator
8: **Else**
9 ++l – Increase iterator of temp array
10: $TempA_l = DataA_n$
11: $MeanA_k$ = Average ($TempA_0$ …. $TempA_l$) – Take mean of values and store in *MeanA*
12: $WeightA_k$ = l +1 – Store weight for above mean value
13: **Endif**
14: ++n
15: **End loop**
16: **Return** (*MeanA* array, *WeightA* array) – Returns aggregated data
17: **Stop**

The sink node computes node redundancy; the whole network will die if the sink node fails. Further, for very high node density (more than 1000 nodes), the DARR protocol causes higher computational overheads. DARR protocol does not capture the redundant data values if those are not consecutive in the given interval. We have proposed a DARR-HCT protocol to remove this limitation. The computation of eliminating redundant data using DARR-HCT is discussed in the next section.

3.4.1. Removing redundant data using DARR-HCT protocol

We have used the DARR-HCT (High Compression with Temporal correlation) protocol to eliminate large number of redundant data measurements. In this technique, the current data value is compared with the mean of prior redundant data values as shown in Table 4. If the difference is above the similarity measures threshold (ε), it is considered non-redundant and retained, otherwise, its mean is computed with previously identified redundant values. We consider the same data set (10 Data values) as considered in the earlier DARR protocol. DARR protocol determines the number of non-redundant data-values as six (6) shown in Table 3. It is further reduced to non-redundant data-values as two (2) by using DARR-HCT protocol. We illustrate the steps to find the aggregated data measurements in Table 4.

Table 4. Removal of redundant data using HCT

Steps	Data values $S_{i,j}$	DARR-HCT (Compared with mean value of each set)		
		Similarity measures threshold (ε) = 0.05		
		Cognate ($S_{i,j}$, $S_{i,j+1}$)	μ	(μ, w)
1	5.11			(5.11, 1)
2	5.09	1	5.1	(5.10, 2)
3	5.12	1	5.107	(5.107, 3)
4	5.20	0		(5.107, 3) (5.20, 1)
5	5.18	1	5.19	(5.107, 3) (5.19, 2)
6	5.08	1	5.1	(5.10, 4) (5.19, 2)
7	5.14	1	5.108	(5.108, 5) (5.19, 2)
8	5.08	1	5.103	(5.103, 6) (5.19, 2)
9	5.19	1	5.19	(5.103, 6) (5.19, 3)
10	5.21	1	5.195	(5.103, 6) (5.195, 4)

The identification and removal of redundant data from all the given data measurements using the DARR-HCT protocol is explained using Algorithm 3.

Algorithm 3 : Data Aggregation using HCT

```
Input: DataA   is input data values array contains j number of time slots with data values),
MeanA is array contains mean values stored after calculation,
WeightA is array contains weightage for each mean value,
TempA is array to keep data values to calculate mean
Start
Initialize array MeanA, WeightA, TempA and variables n=0, k=0, l=0, m=0,
                                        within_threshold = No
ε = 0.05     – Constant for similarity measures threshold
TempAl = DataAn , MeanAk = DataAn , ++n    – Assign first data value
Repeat from 2nd value till jth value    – Last value in DataA
    Repeat till m <= k – Repeat for all mean values calculated
        If difference (DataAn, MeanAm) <= ε    – To check if difference is within
                                        one of the mean values in the set
            within_threshold = Yes
            Exit loop
        Else
            ++m
        Endif
    End loop
    If within_threshold = Yes
        l = WeightAm
        TempAml = DataAn
        MeanAm = Average (TempAm0 .. TempAml)    – Take a mean of values and store in Mean A
        ++WeightAm
    Else
        ++k
        MeanAk = DataAn
        WeightAk = 1
    Endif
    m=0, ++n, within_threshold = No
End loop
Return (MeanA array, WeightA array)         – Returns aggregated data
Stop
```

4. Results and discussion

We have used the NS2 simulator [36] to assess the performance of the suggested DARR protocol. In the simulation scenario, up to five hundred wireless sensor nodes and one sink node are established. Every node establishes a constant bit rate (CBR) session with the sink node. There are orphan nodes that are unable to transfer packets to the sink node when there are relatively few nodes, for example 10 or 20 nodes. This is due to the fact that since nodes are spread randomly throughout the network, certain nodes are not within the communication range of these orphan nodes. After a certain number of iterations, a few nodes may fail or run out of energy. Each parameter has been simulated ten times, and the average outcomes are shown. The calculation of various losses over the communication link is out of the scope of our paper. Hence, we have assumed the communication link to be ideal. Table 5 displays the simulation parameters that were taken into account when evaluating various protocols.

We compared the performance of the proposed DARR protocol with two cutting-edge protocols used STCDRR [12], and DA-AFM [3]. The evaluation parameters for WSN, such as Percentage of Redundant nodes, Left-out nodes, Network Energy consumption, Reduction in data measurements, Data measurements transmitted, Network Lifespan, and Latency, are analyzed. During the simulation, the number of data measurements (M) is considered from 10 to 250. Table 6, Table 7 and Table 8 display the readings for varying node counts and evaluation parameters used in the data aggregation process. The standard deviation values for the simulation's evaluated parameters have been found to vary from -2.5% to +2.5%.

Table 5. Experimental parameter settings

Parameter	Numerical Value	Parameter	Numerical Value
Network area	200 * 200 sq. m	Mac protocol type	IEEE 802.11b
Number of sensor nodes	25 - 500	Bandwidth	1 Mbps
Number of sink node	1	Frequency	2.4 GHz
Node id size	1 byte	Packet arrival rate	0.1 sec
Mobility model	None	Simulation time	600 sec
Control packet size	32 bytes	Transmission Range	20 m
Data packet size	256, 512 bytes	Redundancy threshold (x)	0.1 – 0.9
Application Type	CBR	Similarity measures threshold (ε)	0.01 – 0.1
Initial energy	1 J	Node-Alive threshold (Γ)	95

4.1. Percentage of Redundant Nodes

The percentage of redundant nodes increases with the increase in redundancy threshold as shown in Figure 5, with the node density varying from 25, 50, 100, 150, and 200 nodes in the given network area.

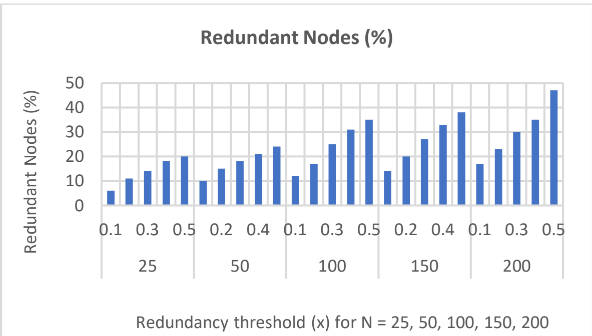

Figure 5. Redundancy threshold Vs Redundant nodes

Figure 5. shows that for 25 nodes, with redundancy threshold (x) as 0.1, the redundant nodes are 6% and with the value of x as 0.5, the redundant nodes are 20%. For 200 nodes, with redundancy threshold (x) of 0.1, the redundant nodes are 17%, and with x as 0.5, the redundant nodes are 47%.

4.2. Left-out nodes

In the randomly distributed network, some of the nodes are not in the communication range, so they cannot communicate with each other. Such nodes are called left-out nodes. The left-out nodes increase with the redundancy threshold (x). When x is increased, more nodes enter the Sleep state, therefore many nodes can be left out. The value of x can be optimized, based on the needs of the application to minimize the number of nodes that are left out, maximize area coverage, and ensure that the network uses less energy. Figure 6 shows the redundancy threshold versus left-out nodes. The left-out nodes are 4% when the spatial constant(x) is 0.1 and 22% when x is 0.5 with 200 nodes in the network.

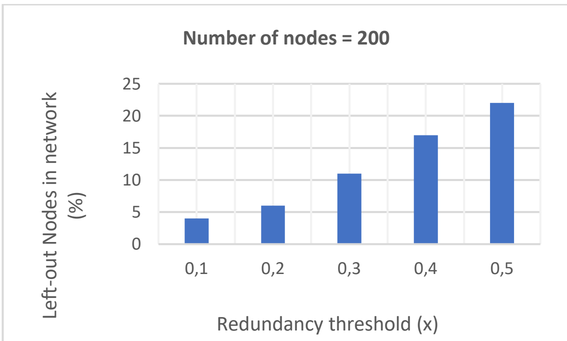

Figure 6. Redundancy threshold Vs Left-out nodes

4.3. Network Lifespan

Figure 7 shows the relationship between the redundancy threshold and network lifespan. If we increase the redundancy threshold, the more number of nodes becomes redundant.

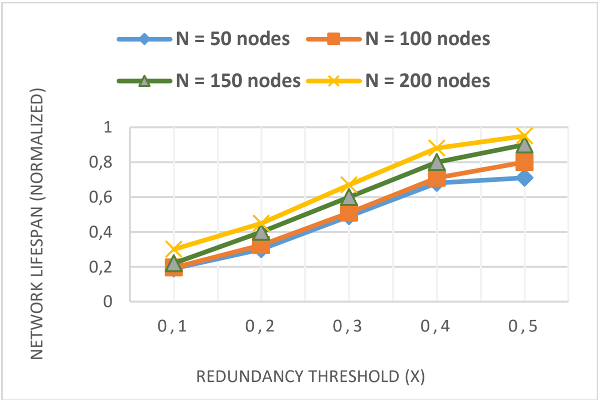

Figure 7. Redundancy threshold Vs Network Lifespan

The DARR protocol sets all redundant nodes to Sleep state. The network lifetime is extended because more nodes in the Sleep state prevent redundant data transmission from the sleeping nodes, which saves their energy.

From Figure 7, we observe that with a network of 50 nodes, the normalized value of the average network lifespan is 0.19 and 0.71 when the redundancy threshold is set to 0.1 and 0.5, respectively. Similarly, with a network of 200 nodes, the normalized value of the average network lifespan is 0.3 and 0.95 when the redundancy threshold is set to 0.1 and 0.5, respectively.

Table 6 indicates the percentage of redundant nodes, percentage of left-out nodes, and normalized value of the network lifespan for varying numbers of sensor nodes from 25 to 500 with a redundancy threshold in the range from 0.1 to 0.5. For the simulation results mentioned below, we have considered the packet size of 256 bytes.

Table 6. Evaluation parameters for Number of nodes, N = 25, 50, ...500 for DARR protocol

Number of nodes (N) →	Redundancy threshold (x) ↓	N=25	N=50	N=100	N=150	N=200	N=300	N=400	N=500
Redundant Nodes (%)	0.1	6	10	12	14	17	21	24	26
	0.2	11	15	17	20	23	30	35	39
	0.3	14	18	25	27	30	38	42	46
	0.4	18	21	31	33	35	44	47	50
	0.5	20	24	35	38	47	51	56	60
Left-out Nodes (%)	0.5	16	12	11	11.33	11	10	9.6	9
Network Lifespan (normalized)	0.1	0.11	0.19	0.20	0.22	0.3	0.36	0.4	0.42
	0.2	0.23	0.3	0.324	0.4	0.45	0.47	0.5	0.51
	0.3	0.38	0.49	0.51	0.6	0.67	0.69	0.7	0.71
	0.4	0.44	0.68	0.71	0.8	0.88	0.91	0.91	0.93
	0.5	0.59	0.71	0.8	0.9	0.95	0.95	0.96	0.97

4.4. Energy consumption at source nodes

When the source nodes sense the data during the time interval (t), much of the sensed data is found to be redundant. The redundant data is not transmitted to the adjacent nodes, so the energy consumption of the source nodes is less. The value of the data sensing time interval (t) is defined by the application.

The graph of the similarity measures threshold (ε) against the average normalized energy consumption at a randomly chosen source node-97 is displayed in Figure 8. We have considered twenty and fifty number of sensed data measurements (M) for simulation.

For 50 data measurements, we have observed that, without aggregation or no aggregation (NA), the normalized value of energy consumption is 100%.

In comparison to energy consumption without aggregation, energy consumption with temporal aggregation is 90% at the threshold (ε) value set to 0.03, and 40% at ε value set to 0.1.

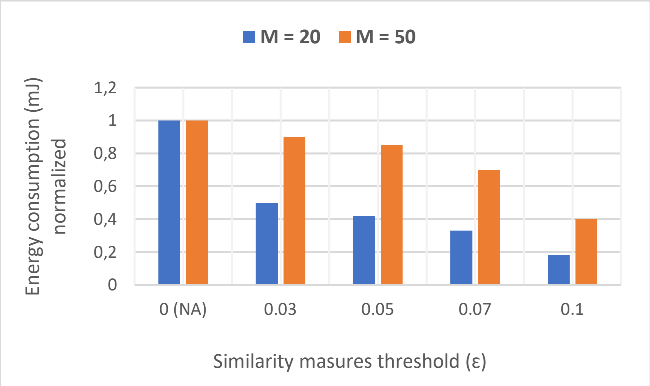

Figure 8. Energy consumption at source nodes

4.5. Reduction in datasets after aggregation

The amount of redundant data removed at source nodes depends on the similarity measures threshold (ε). Redundancy in the data measurements increases as we increase the value of similarity measures threshold. Figure 9 shows the percentage reduction in data measurements sent for varying values of several data measurements (M) from 10 to 50. The reduction in datasets, for ε value set to 0.03, is 6% and 19.5% for a number of data measurements 10 and 50, respectively, for DARR protocol. For ε value set to 0.03, the reduction in datasets is 3% and 16.5% using the STCDRR protocol and, 2% and 15% in the DA-AFM protocol with values of M set to 10 and 50, respectively. The decrease in datasets for DARR, STCDRR, and DA-AFM protocol is 48%, 43%, and 42%, respectively, for ε value set to 0.1 with 50 data measurements. As the similarity measures threshold increases, we observe a larger reduction in datasets.

In our proposed mechanism, the protocol takes the average of two successive data observations. Then, the average value is compared with the subsequent consecutive data measurement, so the difference or distance between the mean and subsequent data measurement is minimized and thus we extract more redundant data, than the other two protocols.

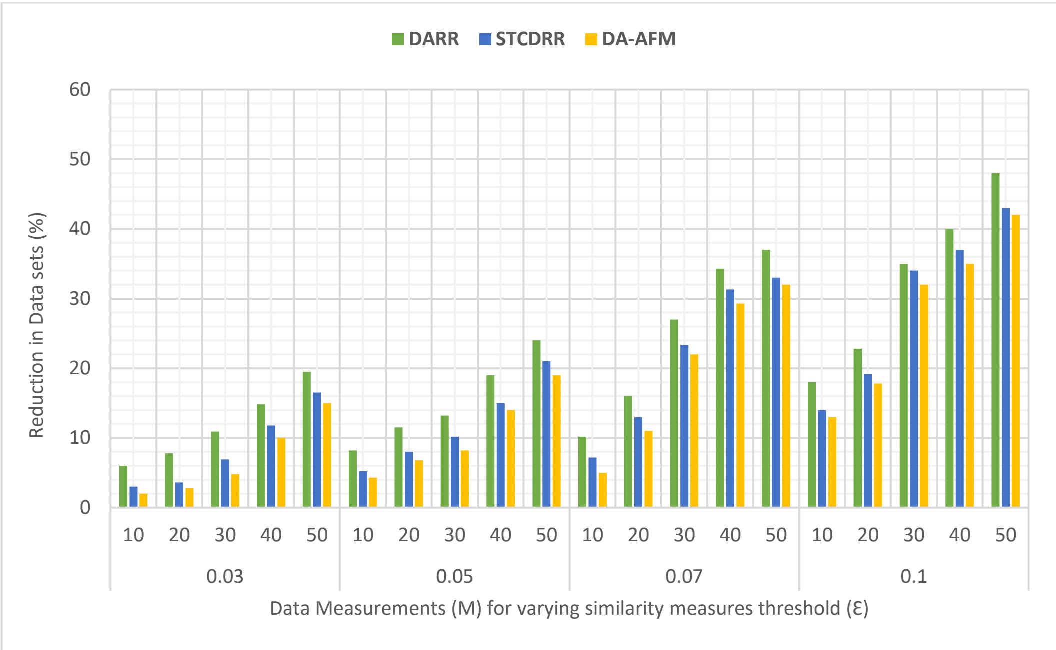

Figure 9. Data measurements Vs Reduction in Data sets

The comparative results in percentage reduction in data measurements for M ranging from 10 to 50 data measurements with varying similarity measures threshold values for DARR, STCDRR, and DA-AFM protocol are shown in Table 7. We have considered the packet size of 256 bytes to obtain the simulation results shown in Table 7.

Table 7. Reduction in data sets after aggregation (%)

Similarity measures threshold (ε)	Number of Data Measurements (M)	Reduction in Data sets (%)				
		M=10	M=20	M=30	M=40	M=50
0.03	DARR	6	7.8	10.9	14.8	19.5
	STCDRR	3	3.6	6.9	11.8	16.5
	DA-AFM	2	2.8	4.8	10	15
0.05	DARR	8.2	11.5	13.2	19	24
	STCDRR	5.2	8	10.2	15	21
	DA-AFM	4.3	6.8	8.2	14	19
0.07	DARR	10.2	16	27	34.3	37
	STCDRR	7.2	13	23.3	31.3	33
	DA-AFM	5	11	22	29.3	32
0.1	DARR	18	22.8	35	40	48
	STCDRR	14	19.2	34	37	43
	DA-AFM	13	17.8	32	35	42

4.6. Reduction in datasets using DARR-HCT

Figure 10 shows the percentage reduction in data measurements for DARR and DARR-HCT protocol. When employing the DARR-HCT protocol, data measurements are reduced by 60 % for 250 data measurements, while the DARR protocol reduces data measurements by 38 % for a similarity measures threshold value of 0.05. However, there is a trade-off between the reduction in datasets and the accuracy of the retrieved data.

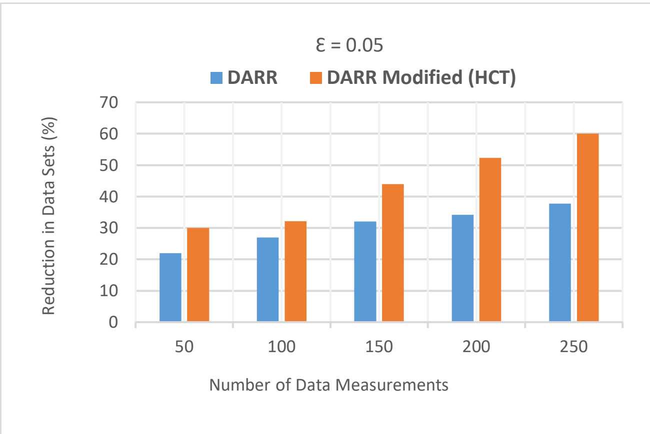

Figure 10. Reduction in datasets using HCT

4.7. Data measurements sent

Figure 11 shows the percentage of data measurements sent for varying values of redundancy threshold for 200 nodes. Without data aggregation, the Active nodes send all the data measurements, including redundant data to the sink node so the network consumes more energy. The DARR protocol transmits comparatively fewer data measures than the other two protocols i.e. 39% for DARR protocol, 52% and 64% for STCDRR and DA-AFM method respectively. In the proposed protocol, the redundant nodes are identified in the network and put into a sleep state, and only the remaining active nodes send the data measurements. Since the redundant nodes are in a sleep state, they do not participate in sending the data, so only Active nodes send the data. This results in the transmission of very less data measurements to the sink node. In contrast, in the compared protocols, all the nodes send the data measurements to the cluster head and redundant data is eliminated at CH and the aggregated data is sent to the sink node.

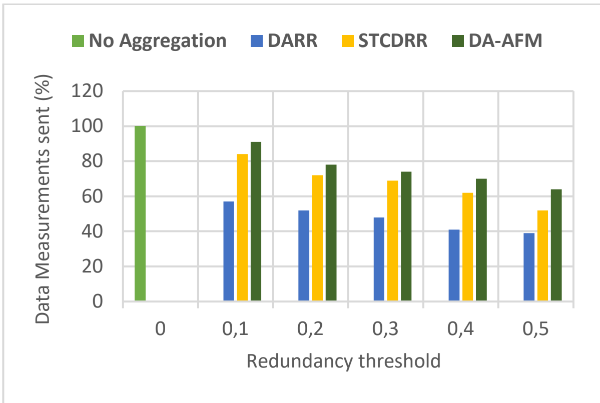

Figure 11. Redundancy threshold vs. Data measurements sent (%)

4.8. Network Energy consumption

When sensor nodes are densely populated, each node has numerous redundant neighbours due to the random distribution of sensor nodes. According to the DARR protocol, redundant nodes hold off on transmitting the data until the residual energy of the Active node in its sensing area drops below a certain threshold. Unlike the comparable protocols, this protocol saves network energy by setting all redundant nodes to the Sleep state. In the STCDRR and DA-AFM protocols, the data aggregation works at two levels, i.e., source node and cluster heads. In STCDRR, in the aggregation process, every node periodically reads the measurements of the data it has acquired and uses the Euclidean distance function to send its data set and weights to the CH. The periodic data transmission results in saving the network energy. In the DA-AFM method, data redundancy is eliminated during aggregation by applying filters using the Relative Deviation and Adaptive Frame methods at both levels (source node and CH), which introduces additional overhead. The amount of network energy consumed increases with the number of data collection cycles or epochs.

Figure 12 shows the normalized network energy consumption for various data-aggregation rounds for the packet size of 512 bytes. In the 250th round, the network energy consumption using the DARR protocol is 21% whereas, for the STCDRR protocol and DA-AFM technique, it is 54% and 62%, respectively. As the number of data aggregation rounds further increases to 1000, network energy consumption using DARR protocol is 62% whereas, for STCDRR protocol and DA-AFM technique, it is 71% and 90% respectively. Table 8 shows the comparative data for the below mentioned three protocols for data aggregation rounds varying from 250 to 1000 for the packet sizes of 256 and 512 bytes. The DARR protocol outperforms the compared, STCDRR and DA-AFM protocols in terms of network energy.

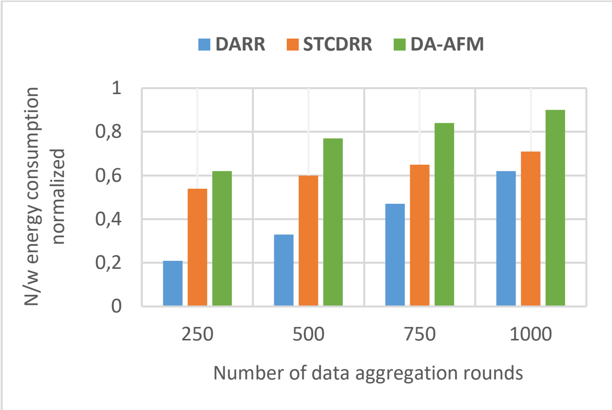

Figure 12. Network Energy consumption

Table 8. Network energy consumption

Number of data aggregation rounds	Packet size (bytes)	Network energy consumption (normalized)		
		DARR	STCDRR	DA-AFM
250	256	0.18	0.32	0.43
	512	0.21	0.54	0.62
500	256	0.25	0.44	0.54
	512	0.33	0.60	0.77
750	256	0.39	0.51	0.6
	512	0.47	0.65	0.84
1000	256	0.44	0.58	0.79
	512	0.62	0.71	0.90

4.9. Network Lifetime

Figure 13 shows the normalized network lifetime for the three protocols discussed. The DARR protocol's data aggregation technique reduces the number of data transmissions at two levels. First, redundant nodes are put into sleep mode, which prevents unnecessary data transmissions. Second, redundant data is removed through temporal correlation when source nodes aggregate data over a set period. This decrease in network energy usage extends the network lifetime. The DARR protocol achieves a network lifetime that is 31% longer than the STCDRR method and 58% longer than the DA-AFM method over 1,000 data transmission rounds. The DA-AFM approach, which uses three-layer single-hop clusters in distributed sensor networks, has higher energy consumption due to increased data transmissions. In next subsection, we have presented real time application of DARR protocol used in animal tracking system.

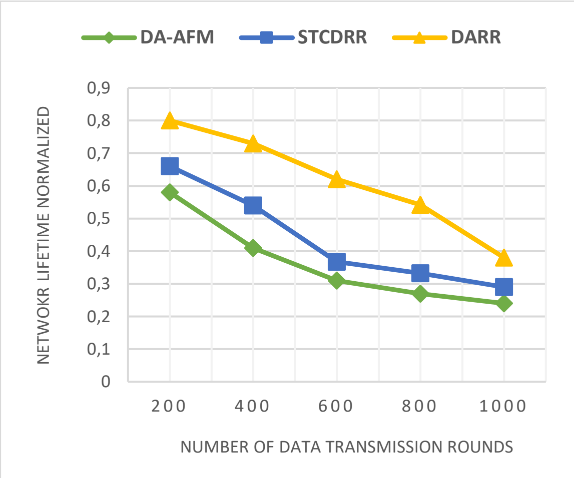

Figure 13. Average Network Lifetime

4.10. Application Scenario

The DARR protocol can be used in various applications, such as wildlife tracking, early warning systems, etc., in dense forest environments. For example, the DARR protocol effectively addresses the challenges of data and node redundancy in animal tracking systems within dense forest environments. It operates through a two-phase strategy: The protocol identifies and optimizes sensor node usage by evaluating spatial distances between nodes. This ensures that only strategically active nodes monitor animal movements while redundant nodes are switched to a sleep state. By conserving energy at inactive nodes, the network achieves reduced redundancy and prolonged operational life. Applying a temporal correlation-based similarity measure, the DARR protocol filters redundant data frames generated during animal tracking. Only unique data is transmitted to the sink node, further minimizing energy consumption and maximizing efficiency. By integrating these mechanisms, the DARR protocol not only ensures efficient resource utilization but also significantly extends the network's lifespan while maintaining effective animal tracking capabilities.

5. Conclusion and Future Scope

We have proposed the DARR protocol for energy-efficient and reliable data aggregation in wireless sensor networks. With the use of a spatial distance technique, the proposed protocol identifies the redundant nodes and configures the network environment to meet application requirements. Unlike other protocols, periodic updates are not necessary to maintain accurate spatial correlation between the aggregated data sets, as we set the redundant nodes to Sleep state. This reduces the number of data transmissions; hence it largely reduces the communication overhead. Periodically monitoring the Node-Alive Index and setting the redundant nodes to Sleep state, reduces the energy consumption of the network. Furthermore, it aggregates data at each source node for a set period using temporal correlation. Cognate function removes redundant data from consecutive time slots at the source node level, resulting in fewer data transmissions. Additionally, HCT mechanism reduces redundant data leading to very few aggregated data transmissions thereby increasing the network lifetime. The simulation result shows that the proposed protocol has a network lifespan that is 31% longer than the STCDRR protocol and 58% longer than the DA-AFM protocol.

Our future work will focus on improving scalability, real-time adaptation, and redundancy removal by using advanced Artificial Intelligence (AI) techniques. AI-based techniques can be used to identify patterns in the data, differentiate meaningful data from redundant data, perform adaptive routing, and form flexible clusters in dynamic environments. These techniques will evolve to provide more innovative, efficient, highly scalable and resource-conscious aggregation strategies while maintaining data integrity and performance.

References

[1] Bomnale, A., Malgaonkar, S., "Power optimization in wireless sensor networks", In: 2018 International Conference on Communication Information and Computing Technology (ICCICT), pp. 1-6 (2018), IEEE. doi:10.1109/ICCICT.2018.8325870

[2] More, A., Raisinghani, V., "A node failure and battery-aware coverage protocol for wireless sensor networks", Computers & Electrical Engineering, 64, 200-219 (2017). https://doi.org/10.1016/j.compeleceng.2016.12.024

[3] Jain, K., Kumar, A., "Energy-efficient data-aggregation technique for correlated spatial and temporal data in cluster-based sensor networks", International Journal of Business Data Communications and Networking (IJBDCN) 16(2), 53-68(2020). doi:10.4018/IJBDCN.2020070103

[4] Li, J., Cheng, S., Cai, Z., Yu, J., Wang, C., Li, Y., "Approximate holistic aggregation in wireless sensor networks", ACM Transactions on Sensor Networks (TOSN) 13(2), 1-24 (2017). https://doi.org/10.1145/302748

[5] Cheng, S., Cai, Z., Li, J., Gao, H., "Extracting kernel dataset from big sensory data in wireless sensor networks", IEEE Transactions on Knowledge and Data Engineering 29(4), 813-827 (2016) https://doi.org/10.1109/TKDE.2016.2645212.

[6] Yan, M., Ji, S., Han, M., Li, Y., Cai, Z., "Data aggregation scheduling in wireless networks with cognitive radio capability", In: 2014 Eleventh Annual IEEE International

Conference on Sensing, Communication, and Networking (SECON), pp. 513-521 (2014), IEEE. https://doi.org/10.1109/GLOCOM.2016.7841716

[7] Villas, L.A., Boukerche, A., Guidoni, D.L., de Oliveira, H.A.B.F., de Araujo, R.B., Loureiro, A.A.F., “An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks”, Computer Communications 36(9), 1054-1066 (2013). https://doi.org/10.1016/j.comcom.2012.04.007

[8] Singh, C., Basha, S., Bhushan, A., Venkatesan, M., Chaturvedi, A., Shrivastava, A., “A secure iot based wireless sensor network data aggregation and dissemination system”, Cybernetics and Systems, 1-13 (2023). https://doi.org/10.1080/01969722.2023.2176653

[9] Gulati, K., Kumar Boddu, R.S., Kapila, D., Bangare, S.L., Chandnani, N., Saravanan, G., “A review paper on wireless sensor network techniques in internet of things (iot) ”, Materials Today, Proceedings 51, 161-165 (2022). https://doi.org/10.1016/j.matpr.2021.05.067

[10] Sekar, K., Suganya Devi, K., Srinivasan, P., “Energy efficient data aggregation using spatio-temporal compressive sensing for wsns”, Wireless Personal Communications 117(2), 1279-1295 (2021). doi:10.1007/s11277-020-07922-x

[11] Maivizhi, R., Yogesh, P., “Spatial correlation based data redundancy elimination for data aggregation in wireless sensor networks”, In: 2020 International Conference on Innovative Trends in Information Technology (ICITIIT), pp. 1-5 (2020), IEEE. doi:10.1109/ICITIIT49094.2020.9071535

[12] Dash, L., Pattanayak, B.K., Mishra, S.K., Sahoo, K.S., Jhanjhi, N.Z., Baz, M., Masud, M., “A data aggregation approach exploiting spatial and temporal correlation among sensor data in wireless sensor networks”, Electronics 11(7), 989 (2022). doi:10.3390/electronics11070989

[13] Kaur, M., Munjal, A., “Data aggregation algorithms for wireless sensor network, A review”, Ad hoc networks, vol. 100, 102083 (2020). https://doi.org/10.1016/j.adhoc.2020.102083

[14] Sharmin, S., Ahmedy, I., Md Noor, R., “An energy-efficient data aggregation clustering algorithm for wireless sensor networks using hybrid pso”, Energies 16(5), 2487 (2023). doi:10.3390/en16052487

[15] Daanoune, I., Baghdad, A., Ballouk, A., “Improved leach protocol for increasing the lifetime of wsns”, Int J. Electr. Comput. Eng. IJECE 11(4), 3106-3113 (2021). doi:10.11591/ijece.v11i4.pp3106-3113

[16] Begum, B.A., Nandury, S.V., “Data aggregation protocols for wsn and iot applications - a comprehensive survey”, Journal of King Saud University - Computer and Information Sciences 35(2), 651-681 (2023). https://doi.org/10.1016/j.jksuci.2023.01.008

[17] Fitzgerald, E., Pioro, M., Tomaszwski, A., “Energy-optimal data aggregation and dissemination for the internet of things”, IEEE Internet of Things Journal 5(2), 955-969 (2018). doi:10.1109/JIOT.2018.2803792

[18] Behera, T.M., Samal, U.C., Mohapatra, S.K., “Energy-efficient modified leach protocol for iot application”, IET Wireless Sensor Systems 8(5), 223-228 (2018). https://doi.org/10.1049/iet-wss.2017.0099

[19] Rahman, H., Ahmed, N., Hussain, M.I., “A qos-aware hybrid data aggregation scheme for internet of things”, Annals of Telecommunications 73, 475-486 (2018). doi:10.1007/s12243-018-0646-3

[20] Zhang, L., Zhou, Y., Wang, J., “An energy saving strategy of wsns based on data sensing and similarity”, Wireless Personal Communications 131(3), 2241- 2268 (2023). https://doi.org/10.1007/s11277-023-10540-y

[21] Ismael, W.M., Gao, M., Yemeni, Z., “Esrra-iot, “Edge-based spatial redundancy reduction approach for internet of things”, Internet of Things 14, 100388 (2021). https://doi.org/10.1016/j.iot.2021.100388

[22] Yemeni, Z., Wang, H., Ismael, W.M., Wang, Y., Chen, Z., “Reliable spatial and temporal data redundancy reduction approach for wsn”, Computer Networks 185, 107701 (2021). https://doi.org/10.1016/j.comnet.2020.107701

[23] Yadav, S., Yadav, R.S., “Redundancy elimination during data aggregation in wireless sensor networks for iot systems”, In: Recent Trends in Communication, Computing, and Electronics, Select Proceedings of IC3E 2018, pp. 195-205 (2019), Springer. doi:10.1007/978-981-13-2685-1_20

[24] Liazid, H., Lehsaini, M., Liazid, A., “Data transmission reduction using prediction and aggregation techniques in iot-based wireless sensor networks”, Journal of Network and Computer Applications 211, 103556 (2023). https://doi.org/10.1016/j.jnca.2022.103556

[25] Jain, K., Bhola, A., “Data aggregation design goals for monitoring data in wireless sensor networks”, Journal of Network Security Computer Networks, vol. 4, pp. 1-9 (2018). doi:10.5281/zenodo.1464554

[26] Bagaa, M., Younis, M., Djenouri, D., Derhab, A., Badache, N., “Distributed low-latency data aggregation scheduling in wireless sensor networks”, ACM Transactions on Sensor Networks (TOSN) Vol 11(3), 1-36 (2015). doi:10.1145/2744198

[27] Chen, K., Gao, H., Cai, Z., Chen, Q., Li, J., “Distributed energy-adaptive aggregation scheduling with coverage guarantee for battery-free wireless sensor networks”, In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, pp. 1018-1026 (2019), IEEE. https://doi.org/10.1109/INFOCOM.2019.8737492

[28] Nguyen, T.-D., Le, D.-T., Vo, V.-V., Kim, M., Choo, H., "Fast sensory data aggregation in iot networks: Collision-resistant dynamic approach", IEEE Internet of Things Journal 8(2), 766-777 (2020). doi:10.1109/JIOT.2020.3007329.

[29] Chen, Q., Cai, Z., Cheng, L., Gao, H., "Structure-free general data aggregation scheduling for multihop battery-free wireless networks", IEEE Transactions on Mobile Computing 21(09), 3342-3359 (2022). doi:10.1109/TMC.2021.3053557

[30] Bagaa, M., Challal, Y., Ksentini, A., Derhab, A., Badache, N., "Data aggregation scheduling algorithms in wireless sensor networks: Solutions and challenges", IEEE Communications Surveys & Tutorials 16(3), 1339-1368 (2014). doi:10.1109/SURV.2014.031914.00029

[31] Yun, W.-K., Yoo, S.-J., "Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks", IEEE Access, vol. 9, 10737-10750 (2021). doi:10.1109/ACCESS.2021.3051360

[32] Agarkhed, J., Kadrolli, V., Patil, S.R., "Efficient bandwidth-aware routing protocol in wireless sensor networks (EBARP)", International Journal of Information Technology, Vol.14, 1967-1979 (2022). https://doi.org/10.1007/s41870-021-00828-2

[33] Haseeb, K., Islam, N., Saba, T., Rehman, A., Mehmood, Z., "Lsdar: A light-weight structure based data aggregation routing protocol with secure internet of things integrated next-generation sensor networks", Sustainable Cities and Society 54, 101995 (2020). doi:10.1109/ACCESS.2021.3051360

[34] Bomnale, A., More, A., "Node utilization index-based data routing and aggregation protocol for energy-efficient wireless sensor networks", The Journal of Supercomputing 79, 1-33 (2023). doi:10.1007/s11227-023-05800-4

[35] Ghori, M.R., Wan, T.-C., Sodhy, G.C., Rizwan, A., "Optimization of the aodv-based packet forwarding mechanism for ble mesh networks", Electronics 10(18), 2274 (2021). https://doi.org/10.3390/electronics10182274

[36] The network simulator ns-2. http://www.isi.edu/nsnam/ns/ (last used: 2019-2024)

[37] Priyadarshi, R., "Exploring machine learning solutions for overcoming challenges in IoT-based wireless sensor network routing: a comprehensive review", Wireless Network 30, 2647–2673 (2024). https://doi.org/10.1007/s11276-024-03697-2

[38] Nayak, Padmalaya, et al., "Routing in wireless sensor networks using machine learning techniques: Challenges and opportunities", Measurement 178 (2021). https://doi.org/10.1016/j.measurement.2021.108974

[39] Ravinder Singh, Sudha Subramani, et.al., "Antisocial Behaviour Identification from Twitter Feeds Using Traditional Machine Learning Algorithms and Deep Learning", EAI Endorsed Transactions on Scalable Information Systems, SIS 10(4) (2023). doi:10.4108/eetsis.v10i3.3184

[40] Lakshmi, M. S., K. J. Kashyap, S. M. Fazal Khan, N. J. S. Vratha Reddy, and V. B. Kumar Achari. "Whale Optimization Based Deep Residual Learning Network for Early Rice Disease Prediction in IoT", EAI Endorsed Transactions on Scalable Information Systems, vol. 10, no. 6, (2023). doi:10.4108/eetsis.4056.

[41] [41Chen, Xiaoyu & Zhang, Junkai & Ren, Pengshan & Hua, Xian & Ni, Yanfeng. (2023), "Database System Based on 3Dmax Big Data Mining Technology", ICST Transactions on Scalable Information Systems, (2023) doi:10.4108/eetsis.3727.

[42] C. -Q. Huang et al., "XKT: Toward Explainable Knowledge Tracing Model With Cognitive Learning Theories for Questions of Multiple Knowledge Concepts," in IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 11, pp. 7308-7325, (2024). doi:10.1109/TKDE.2024.3418098.

[43] Vijayakumar, K., P. Thirumaraiselvan, B. Sivakumar, and P. Seenuvasan. "Revolutionizing Data Transmission-A Comprehensive Review of Deep Learning-Based Routing Mechanism for 5G Wireless Sensor Networks." In 2024 10th International Conference on Communication and Signal Processing (ICCSP), pp. 565-570. IEEE, (2024). doi: 10.1109/ICCSP60870.2024.1054433