International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

An Energy-Efficient Clustering and Scheduling Framework in MATLAB with Visualization for Wireless Sensor Networks

Suhas B. Pharne[1], Jaymala K. Patil[2]

[1] Research Scholar, [2]Associate Professor
[1,2]Department of Electronics and Telecommunication Engineering
[1,2]Bharati Vidyapeeth's College of Engineering Kolhapur, Affiliated to Shivaji University, Kolhapur, Maharashtra, India

Abstract:

Wireless Sensor Networks (WSNs) are increasingly deployed for various monitoring applications. Energy efficiency is a critical concern in WSNs due to the limited power resources of sensor nodes. This paper presents a MATLAB-based simulation framework for exploring a fundamental Energy-Efficient Data Collection (EEDC) strategy. The architecture uses similarity-based clustering, using Euclidean distance to cluster sensor nodes with similar temporal data patterns through single-linkage hierarchical clustering. A simple, randomized scheduling scheme is then used within each cluster to control the active times of sensor nodes. The framework supplies thorough visualizations, such as a dendrogram of the clustering, time series data with cluster assignments, distribution of clusters on the network layout, sensor node working schedules (Gantt chart), and a dissimilarity matrix heatmap. This simulation is a basic platform for exploring the effect of data similarity and minimal coordination on possible energy savings in WSNs.

Keywords: Wireless Sensor Networks, Energy Efficiency, Data Collection, Clustering, Scheduling, Simulation, MATLAB.

1. Introduction:

Wireless Sensor Networks (WSNs) are collections of spatially localized sensors that sense environmental or physical parameters. Energy efficiency continues to be an essential challenge due to limited power resources of sensor nodes. Clustering and dynamic scheduling are established approaches to improving energy use. The work presented in this paper presents an implementation on EEDC framework using MALAB simulation software. It utilizes dissimilarity-driven clustering and periodic scheduling to reduce redundant sensing operations in sensor networks. WSNs are typically deployed in inaccessible or hostile environments where battery replacement through human intervention is practically critical. As such, intelligent data collection and communication strategies that reduce energy consumption are critical. Clustering techniques are traditionally used to organize sensor nodes to achieve load balance and optimize communications. Nevertheless, dynamic methods taking into account instantaneous sensor behaviour have the potential for improved efficiency. EEDC goes further by creating clusters through behavioural similarity, an adaptive approach compared to static ones.

The EEDC framework improves sensor network lifespan by incorporating temporal consciousness into scheduling. Based on exploring the similarity in time-series data, it recognizes nodes that are able to share sensing tasks. This results in fewer active sensors at a specific instant without affecting data quality. Additionally, EEDC has a modular architecture with strong visual aid and is suitable for researchers and engineers in need of interpretable, simulation-based optimization methods for energy-restricted WSNs. MATLAB has been widely used in various studies to simulate and evaluate energy-efficient data collection, routing and clustering techniques in Wireless Sensor Networks (WSNs). For instance, Pandey et al. (2023) employed MATLAB to simulate Energy Efficient Data Collection Schemes (EEDCS) for IoT-enabled WSNs, optimizing cluster head selection and transmission strategies to reduce energy consumption and extend network lifetime. Similarly, Ahmed et al. (2022) used MATLAB for their Secure and Energy Efficient Data Gathering Technique (SEEDGT), which combines trust evaluation, public key cryptography, and Compressive Sensing, demonstrating its effectiveness in ensuring secure and efficient data transmission. In another study, Sharmin et al. (2023) tested the HPSO-ILEACH algorithm in MATLAB, which integrates Particle Swarm Optimization (PSO) with Improved LEACH (ILEACH) for optimal cluster head selection, showing a 28% reduction in energy consumption and a 55% improvement in node survival. Singh et al. (2021) leveraged MATLAB to evaluate their secure and energy-efficient data transmission model, comparing various private key cryptography algorithms like AES and Blowfish for their energy efficiency in encryption and decryption processes.

Additionally, Manchanda and Sharma (2021) used MATLAB to simulate their Energy-Efficient Compressive Data Gathering (NFECG) framework, which improved network stability by 52.59% and network lifetime by 46.09% compared to traditional data gathering protocols. Azooz et al. (2022) applied MATLAB simulations to test their Optimal Real-Time Clustering (ORTC) technique, which significantly improved energy consumption and network lifetime by delaying the first dead node occurrence. Kodoth and Edachana (2021) also employed MATLAB to simulate an energy-efficient data gathering scheme using the Hybrid Crow Search Algorithm (HCSA), which optimized cluster head selection and data gathering node selection, outperforming existing methods in terms of total energy consumption and network longevity. In the context of mobile WSNs, Sathish Kumar et al. (2022) used MATLAB to simulate a mathematical network model, which demonstrated a reduction in energy consumption compared to methods like LEACH, ZTR, and DSR. Further, Nazib and Moh (2021) demonstrated the effectiveness of their energy-efficient and fast data collection (EFDC) scheme for UAV-aided WSNs in MATLAB, showing improvements in energy consumption, scalability, and control overhead. Bhasker and Murali (2024) implemented the Energy-Efficient Cluster-Based Data Aggregation (E2CDA) method for agricultural irrigation systems in MATLAB, proving superior energy savings and network lifetime compared to LEACH and LEACH-C. Their subsequent work also employed MATLAB to simulate a dynamic cluster head-based energy-efficient routing system, which outperformed traditional methods in terms of energy usage and data packet delivery. Finally, Adumbabu and Selvakumar (2022) applied MATLAB for their simulation of an energy-efficient routing protocol that incorporated Deep Q-Networks (DQN) for cluster head selection and Predictive Coding Theory for data compression, achieving significant improvements in energy efficiency and quality of service.

Across all these studies, MATLAB played a pivotal role in simulating various algorithms, optimizing network performance, and comparing the effectiveness of different energy-efficient techniques. It provided an ideal platform for assessing key performance metrics such as energy consumption, network lifetime, scalability, and data packet delivery, helping researchers refine their methods and validate their practical applicability. The subsequent sections give a step-by-step description associated with implementation of the EEDC framework in MATLAB. This paper is organized as follows: Section 2 gives an overview of the methodology adopted to simulate sensor readings, calculate dissimilarities, perform clustering, and produce schedules. Section 3 discusses the simulation outcomes, accompanied by several visualizations. Section 4 elaborates on the implications and insights of the results. Lastly, Section 5 concludes the implemented work.

2. Methodology:

The proposed framework is implemented in MATLAB 2023b, leveraging its powerful computational and visualization tools. MATLAB's flexible environment allows for efficient simulation of sensor data, computation of dissimilarity matrices, and implementation of clustering algorithms. The framework also utilizes MATLAB's visualization capabilities to display clustering results, sensor schedules, and dissimilarity relationships. This ensures a comprehensive analysis of sensor network performance. The work flow for the implemented work is as shown in figure 1.

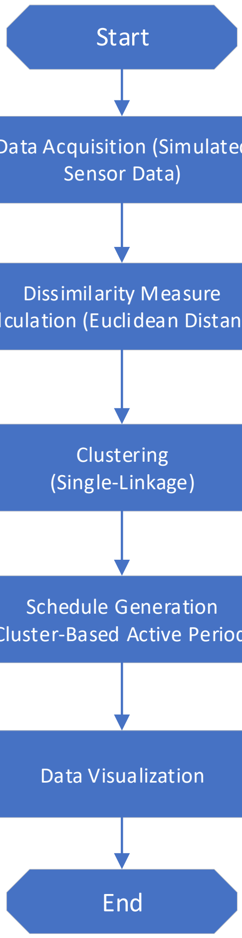

Figure 1 : Work flow of implemented work

As shown in figure 1, the process starts with obtaining simulated sensor data. Then, dissimilarity between sensor data is computed based on Euclidean distance. This dissimilarity data is utilized for clustering sensors through single-linkage. Lastly, the resulting data and clusters are displayed through visualization.

2.1 Data Simulation:

For simulating the sensor network scenario, time series data is randomly created for 15 sensor nodes over 100 time units. A random location is assigned to each sensor within a specified area, simulating an actual deployment of sensors. The randomly created data points are used to represent sensor readings over time, e.g., temperature, humidity, or light intensity, which are common measurements in sensor networks. The randomness guarantees that the data covers an extensive variety of sensor behaviours and enables an effective testing of the clustering and scheduling techniques.

2.2 Dissimilarity Calculation:

The core of clustering in this work relies on calculating a dissimilarity matrix. To quantify how different the time series data from each sensor is, Euclidean distance is used. This distance measures the straight-line difference between sensor data points, effectively capturing how much the behaviour of one sensor differs from another over the given time period. The resulting dissimilarity matrix serves as the foundation for the clustering step, helping group sensors that have similar data patterns over time. The dissimilarity between two sensor time series, ts_1 and ts_2, is calculated using the Euclidean distance. The formula is:

$$d(ts_1 , ts_2) = \sqrt{\sum_{t=1}^{T} \left(ts_1(t) - ts_2(t)\right)^2} \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad [1]$$

Where,

- $ts_1(t), ts_2(t)$ Represent the sensor readings at time t for sensor 1 and sensor 2, respectively.
- T is the total number of time units.

2.3 Clustering Algorithm:

After the dissimilarity matrix is created, hierarchical single-linkage clustering is then used. The algorithm connects sensors by their dissimilarity values, with the single-linkage technique joining two clusters by the minimum distance between them. The final cluster labels are determined by a pre-set distance threshold so that only sensors with data that are close enough are clustered together. The resulting clusters are sensors that have similar behaviour patterns throughout the simulated time.

2.4 Schedule Generation:

Once the sensors are clustered, a schedule is assigned to each sensor within a cluster to manage its activation. The objective of the schedule is to reduce redundant sensing operations, hence minimizing the network's overall energy consumption. Periodic wake-up schedules are developed for each sensor, which means that a fixed time for which each sensor is woken up at intervals. This is to avoid a cluster of multiple sensors sensing at the same instance of time and resulting in energy-efficient operation of the sensor network.

$$\text{schedule}_i(t) = \begin{cases} 1, & \text{if sensor i is active at t} \\ 0, & \text{if sensor i is inactive at t} \end{cases} \qquad \text{..................} [2]$$

2.5 Visualization Techniques:

A few visualization methods are employed to aid interpretation and analysis of the outcome. Dendrograms are utilized to present the hierarchical clustering process, illustrating how sensors are increasingly combined according to their dissimilarities. This presents a clear and visual representation of the hierarchy of clustering.

- Time series plots are utilized to present the sensor data over time. Various colors are utilized to represent which cluster each sensor falls into, and this assists in identifying patterns and differences between clusters.
- Network layout visualizations display the positions of the sensors in the specified area. The visualizations also illustrate the assignments of clusters to identify how sensors are spatially distributed and grouped.
- Gantt-type charts are employed to graphically display the sensor operational schedules, showing at which times each sensor is active. The visual aid facilitates the easy depiction of periodic sensor activation times for all the sensors in the network.
- Heatmaps are used to visually represent the dissimilarity matrix. They represent an easy and intuitive method for seeing how dissimilar or similar sensor data is throughout the entire network, while numerical labels also provide more detail on the exact dissimilarity between each pair of sensor.

2.6 Adaptive Clustering and Cost Evaluation:

An adaptive clustering scheme was adopted to dynamically reorganize clusters of sensor nodes according to intra-cluster variance. Whenever the variance of a cluster was above an underlying threshold value (0.5), it was partitioned to ensure homogeneity so that more effective data aggregation and prediction could be facilitated. This resulted in more precise cluster memberships that were more in conformity with patterns of sensor behaviour. Then, communication and computation costs were calculated by observing each sensor's activity in a predictive transmission model. Communication cost was defined by the number of actual transmissions, while computation cost by local prediction usage. The resulting k-ratio or k parameter (communication-to-computation cost) measures the balance between energy-consuming transmissions and cheap local processing. The k-ratio is determined using the following formula:

$$\mathbf{k} = \frac{\textbf{Communication Cost}}{\textbf{Computation Cost}}$$

Where:

- Communication Cost is the number of transmissions carried out by the sensor network.

- Computation Cost is the number of computations done by sensors to create predictions or process local data.

3. Result and Discussion :

We did simulations using 15 sensor nodes spread over a two-dimensional plane to assess the efficacy of the suggested energy-efficient data transmission technique. Every sensor produced a smooth time-series signal over 100 time steps and was positioned at random. Additionally, each sensor's active operation times were assigned at random; each sensor operated for ten time units. A realistic and dynamic network environment is reflected in the small differences in outcomes between iterations caused by this randomized deployment and scheduling. The system used a linear regression-based adaptive prediction mechanism to decide, based on a predetermined error threshold, whether to send sensor readings or rely on locally predicted values. The transmission data and energy savings from each sensor demonstrated encouraging results. In one typical iteration, a number of nodes reduced communication and relied on accurate forecasts to increase overall energy efficiency. The simulation's 119 transmissions and 75 prediction calculations demonstrate a reasonable trade-off between communication cost and computational effort with a k-ratio up to 10.

Results from several iterations, however, revealed a more general pattern: the k-ratio usually varied up to 10, and the energy savings were noted contingent on prediction accuracy and network conditions. These results highlight the trade-off between communication and computation in sensor networks and show that significant energy savings can be obtained without sacrificing data integrity by carefully adjusting prediction parameters and cluster scheduling.

This section presents the simulation results of the proposed Energy-Efficient Duty Cycling (EEDC) framework, applied to a network of 15 sensor nodes. Each sensor generates synthetic time-series data, and the framework is evaluated in terms of clustering effectiveness, scheduling efficiency, and visual interpretability. The overall goal is to optimize energy consumption by leveraging behaviour-based clustering and coordinated duty cycling. The dendrogram in Figure 2 illustrates the hierarchical structure of the clusters.

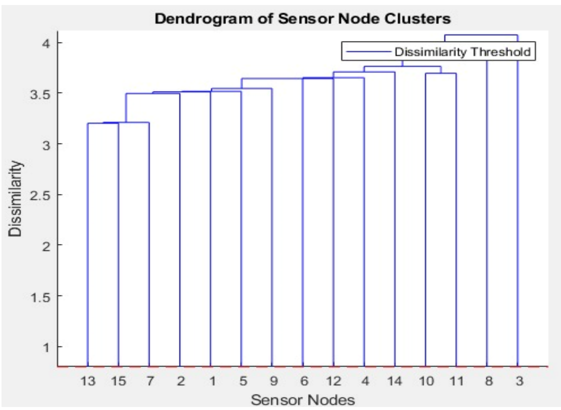

Figure 2 : Dendrogram of sensor node clusters

The red dashed line indicates the dissimilarity threshold used for partitioning. The clusters formed at this level indicate strong intra-cluster cohesion and inter-cluster separation, demonstrating the algorithm's capacity to group sensors with similar behaviour effectively. Figure 3 illustrates the clustering results of multivariate time series sensor data, where each line represents the temporal readings of a sensor over 100 time steps. The data has been clustered based on the similarity of temporal patterns, and the resulting cluster assignments are visualized using a colormap ranging from blue to yellow, indicating different cluster memberships. This visualization highlights the diversity of temporal behaviours among the sensors and demonstrates the effectiveness of the clustering approach in grouping similar time series. Such analysis is crucial for applications like anomaly detection, condition monitoring, and pattern recognition in sensor networks.

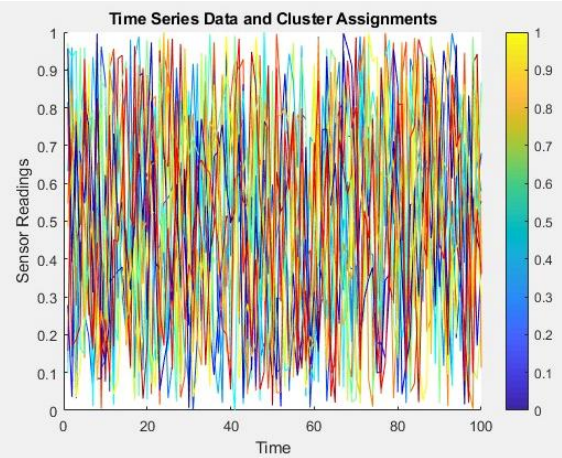

Figure 3: Time series data and cluster assignments

Figure 4 presents the spatial distribution of sensor nodes along with their cluster assignments, visualized on a 2D network layout. Each numbered circle represents a sensor node, positioned according to its physical X and Y coordinates within the deployment area. The color of each node corresponds to its assigned cluster, as indicated by the colormap on the right, which ranges from blue to yellow. This visualization provides insight into how clustering algorithms group sensors based not only on time series patterns but also in relation to their spatial locality. Such clustering can help in optimizing network tasks likedata aggregation, fault detection, and energy-efficient communication in wireless sensor networks.

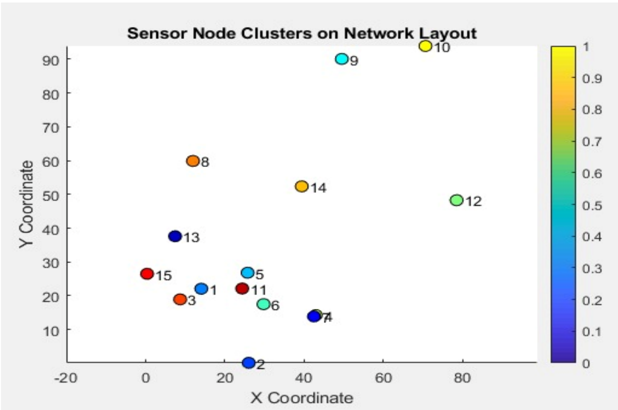

Figure 4: Sensor node clusters on network layout

Figure 5 displays the working schedules of individual sensor nodes over time, where each horizontal blue bar represents an active time slot for a specific node. The y-axis lists the sensor node IDs (1 to 15), and the x-axis denotes the timeline. This schedule visualization highlights the duty cycling behaviour of the network, ensuring that sensors operate in a coordinated and energy-efficient manner by activating only during designated periods. Such scheduling is crucial for extending network lifetime while maintaining adequate coverage and data collection in wireless sensor networks.

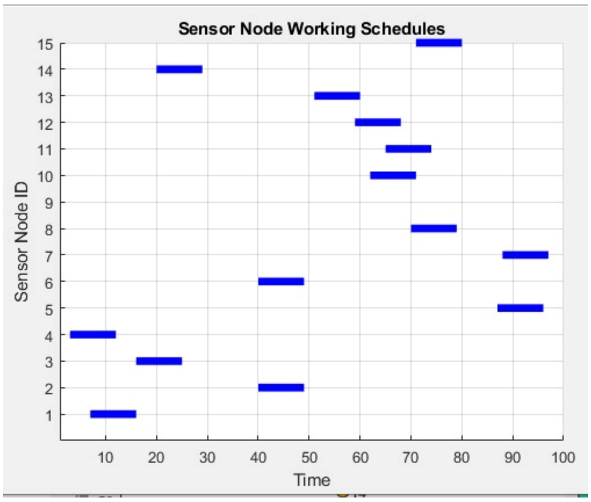

Figure 5 : Sensor node working schedules

Figure 6 shows the dissimilarity matrix heatmap of sensor nodes, with numerical values overlaid to indicate pairwise dissimilarity scores. Each cell (i, j) represents the dissimilarity between sensor node i and sensor node j, based on their time series data or operational patterns. The diagonal values are zero, indicating perfect self-similarity. The color gradient, ranging from black (low dissimilarity) to red and yellow (high dissimilarity), visually emphasizes the extent of dissimilarity across the network. This matrix is instrumental in understanding the behavioural differences between nodes, which can aid in tasks like clustering, anomaly detection, and efficient network reconfiguration.

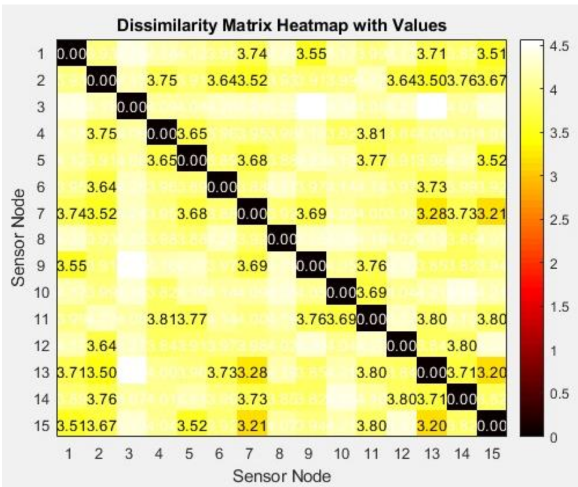

Figure 6 : Dissimilarity matrix heatmap with values

The figure 7 illustrates how sensor nodes transmit data selectively over time based on an adaptive clustering strategy. In the background, an adaptive clustering algorithm is applied, where nodes are grouped into clusters based on the similarity of their sensor data. The provided code shows how clusters are dynamically updated: if a cluster has more than two members and shows high internal variance (greater than 0.5), it is split into smaller, more coherent clusters by sorting nodes according to their variance. This splitting ensures that sensor nodes with significantly different data patterns are separated, allowing for more accurate prediction models and more efficient data transmission. Each colored group of dots in the figure corresponds to transmissions from sensor nodes within a certain cluster, showing periods when only selected nodes transmit. This method significantly reduces unnecessary communication, saving energy and bandwidth in the network while maintaining the accuracy of monitoring.

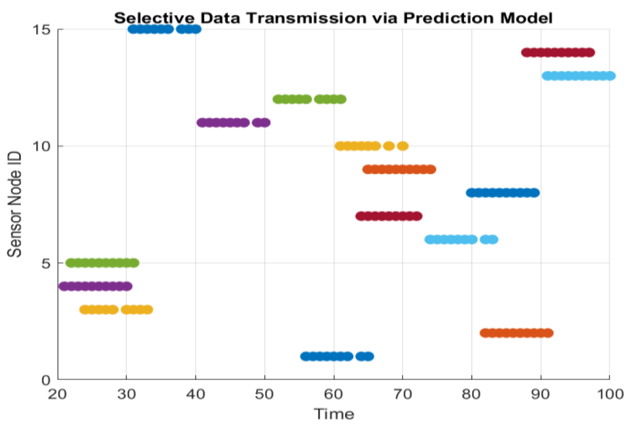

Figure 7 Selective Data Transmission via Prediction Model

4. Conclusion:

The EEDC framework efficiently boosts energy efficiency of wireless sensor networks using intelligent dissimilarity-based clustering and periodically optimized scheduling. By doing this, the framework minimizes duplicate sensing operations tremendously, prolonging the network operational lifetime. Constructed and coded in MATLAB 2023b, the framework leverages the strong computational and visualization capabilities of MATLAB to emulate sensor behaviour, calculate dissimilarities, execute hierarchical clustering, and produce intuitive, informative graphs. These rich visualizations–such as dendrograms, heatmaps, Gantt charts, and network layouts–not only assist with analysis but also render the framework extremely accessible to both scholarly research and actual sensor network administration. Overall, EEDC is a flexible, scalable, and informative solution to energy-aware sensor deployment and monitoring.

References:

1. S. Pandey, K. Dubey, R. Dubey, et al., "EEDCS: Energy Efficient Data Collection Schemes for IoT Enabled Wireless Sensor Network," Wireless PersCommun., vol. 129, pp. 1297–1313, 2023, doi: 10.1007/s11277-023-10190-0.

2. A.Salim, W. Osamy, A. Aziz, and A. M. Khedr, "SEEDGT: Secure and energy efficient data gathering technique for IoT applications based WSNs," J. Netw. Comput. Appl., vol. 202, 2022, Art. no. 103353, doi: 10.1016/j.jnca.2022.103353.
3. S. Sharmin, I. Ahmedy, and R. Md Noor, "An Energy-Efficient Data Aggregation Clustering Algorithm for Wireless Sensor Networks Using Hybrid PSO," Energies, vol. 16, no. 5, pp. 2487, 2023, doi: 10.3390/en16052487.
4. A.K. Singh, M. Alshehri, S. Bhushan, M. Kumar, O. Alfarraj, and K. R. Pardarshani, "Secure and energy-efficient data transmission model for WSN," Intelligent Automation & Soft Computing, vol. 27, no. 3, pp. 761, 2021, doi: 10.32604/iasc.2021.012806.
5. R. Manchanda and K. Sharma, "A novel framework for energy-efficient compressive data gathering in heterogeneous wireless sensor network," Int. J. Commun. Syst., vol. 34, no. 3, 2021, Art. no. e4677.
6. S. M. Azooz, et al., "Implementation of energy-efficient routing protocol within real time clustering wireless sensor networks," Bull. Electr. Eng. Informatics, vol. 11, no. 4, pp. 2062-2070, 2022.
7. P. K. Kodoth and G. Edachana, "An energy efficient data gathering scheme for wireless sensor networks using hybrid crow search algorithm," IET Commun., vol. 15, no. 7, pp. 906-916, 2021.
8. L. Sathish Kumar, et al., "Modern energy optimization approach for efficient data communication in IoT-based wireless sensor networks," Wireless Commun. Mobile Comput., vol. 2022, Art. no. 7901587, 2022.
9. R. A. Nazib and S. Moh, "Energy-Efficient and Fast Data Collection in UAV-Aided Wireless Sensor Networks for Hilly Terrains," IEEE Access, vol. 9, pp. 23168-23190, 2021, doi: 10.1109/ACCESS.2021.3056701.
10. B. Bhasker and S. Murali, "An Energy-Efficient Cluster-based data aggregation for agriculture irrigation management system using wireless sensor networks," Sustain. Energy Technol. Assess., vol. 65, Art. no. 103771, 2024.
11. I.Adumbabu and K. Selvakumar, "Energy efficient routing and dynamic cluster head selection using enhanced optimization algorithms for wireless sensor networks," Energies, vol. 15, no. 21, Art. no. 8016, 2022.