The Scientific Temper (2024) Vol. 15 (spl): 32-44
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.spl.05
https://scientifictemper.com/

RESEARCH ARTICLE

Energy-efficient location-based routing protocol for wireless sensor networks using teaching-learning soccer league optimization (TLSLO)

L. Amudavalli*, K. Muthuramalingam

Abstract

Energy efficiency in wireless sensor networks (WSNs) is a crucial and fundamental design consideration. These networks typically consist of numerous small, resource-constrained sensor nodes, frequently placed in isolated or difficult-to-reach areas. This research presents a comprehensive methodology for improving the performance and energy efficiency of WSNs deployed in a designated target area. The research begins with the deployment of sensor nodes equipped with location information and the initialization of critical network parameters. Novel techniques are introduced for efficient node clustering using a Haversine-based K-means Clustering algorithm (HKMC) and an advanced hybrid optimization model, teaching-learning soccer league optimization (TLSLO), for optimal cluster head selection within clusters. Data aggregation at cluster heads is crucial for conserving energy, and data compression techniques, including the novel weighted discrete wavelet transform (WDWT)), are employed to reduce data transmission size. Furthermore, deep learning models in the form of recurrent artificial neural networks (RANN) predict energy consumption patterns, enabling the optimization of node sleep-wake schedules for a prolonged network lifetime. Simulated using Python, the proposed protocol's performance is evaluated, demonstrating its superiority in terms of energy efficiency, latency, network lifetime, and data delivery ratio compared to existing routing protocols. This research offers a holistic approach to improving WSNs enhancing their efficiency and sustainability in resource-constrained environments.

Keywords: Wireless sensor networks, Energy efficiency, Modified K-means clustering, Teaching-learning soccer league optimization, Recurrent artificial neural network.

Introduction

Various applications can be supported by WSNs depending on where the sensor nodes are placed. Despite differences in the objectives of application scenarios, the nodes' principal duty is to sense the data and communicate it to the BS. The usage of energy-efficient routing methods is required to execute this operation successfully. When building routing algorithms, it is important to consider the resources required for the anticipated application scenarios as well as the sensor nodes' energy use. Furthermore, the employment of conventional routing protocols with WSNs is prohibited by their high routing costs, which rise with network size and dynamic circumstances. Therefore, a routing method that considers network flexibility and sensor node power constraints is required. The routing protocols significantly influence the potential energy efficiency of a WSN. The clustering hierarchy algorithm-based routing protocols are crucial for enhancing energy efficiency in particular. The WSNs are broken up into homogeneous and heterogeneous networks as well. Each node in homogeneous networks receives the same amount of energy in the beginning, but each node in heterogeneous networks receives a variable quantity of leftover energy, Banerjee, A., Sufian, A., Sadiq, A. S., & Mirjalili, S. (2021), Wang, Z., Ding, H., Li, B., Bao, L., & Yang, Z. (2020), Roopali, & Kumar, R. (2020), Maheshwari, P., Sharma, A. K., & Verma, K. (2021), Mehta, D., & Saxena, S. (2022).

The location-based routing systems mainly rely on location information. In order to offer information regarding route discovery, network upkeep, data transfer, and node

Department of Computer Science, Bharathidasan University, Tiruchirappalli, Tamil Nadu, India.

***Corresponding Author:** L. Amudavalli, Department of Computer Science, Bharathidasan University, Tiruchirappalli, Tamil Nadu, India., E-Mail: amudalog@gmail.com

How to cite this article: Amudavalli, L., Muthuramalingam, K. (2024). Energy-efficient location-based routing protocol for wireless sensor networks using teaching-learning soccer league optimization (TLSLO). The Scientific Temper, **15**(spl):32-44.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.spl.05

Source of support: Nil

Conflict of interest: None.

Received: 03/09/2024 **Accepted:** 18/09/2024 **Published**: 16/10/2024

security of private information. The WSN's ability to reduce data flooding over the whole network in large part depends on direct data transfer between nodes. By measuring the distance between sensor nodes where energy is released, location routing serves as a major measurement tool. The most important location-based routing protocols are GAF, MECN, GEAR, AFR, MAR, GRC, and SMECN. Only one node is active at the moment in each cell in the geographic grid that GAF generates, despite each cell having numerous neighboring cells. GAF seeks to extend the network's lifespan and reduce energy use. Mobile ad hoc networks and WSNs are its main applications. MECN is a low-power, GPS-based routing technology that aims to cut down on the network's total power use. Since direct communication requires more energy than transferring data across several relay nodes, this protocol's concept is to send data packets through intermediary nodes rather than directly to the base station. By selecting neighbors using energy-conscious measures, the geographic protocol GEAR balances network longevity and energy use. The protocol incorporates a cost function that calculates the cost of contacting a neighbor depending on the neighbor's location and residual level. AFR is an ad hoc routing system that is based on Euclidean planar networks, which separate a plane's nodes and edges into areas known as faces. AFR employed face routing to go through the faces in a limited fashion. The protocol repeats the same processes using an eclipse that is twice the size of face routing cannot transport the data to the target place. Using the hierarchical position-based routing protocol MAR, the network is split into a geographic grid and cluster heads depending on the mobility measure. The node with the lowest mobility measure, therefore, is the cluster head. Because node energy is not taken into account when selecting a cluster head, this protocol has a major problem. The GRC protocol uses cluster-based routing, choosing the cluster leaders based on node locations and energy levels. Also utilized to restore packet loss is the inter-cluster communication phase. By characterizing a minimal graph according to the minimum energy property, SMECN, a routing protocol, improves MECN, Del-Valle-Soto, C., Mex-Perera, C., Nolazco-Flores, J. A., Velázquez, R., & Rossa-Sierra, A. (2020), Kamarei, M., Patooghy, A., Alsharif, A., & Hakami, V. (2020), Gupta, N. K., Yadav, R. S., & Nagaria, R. K. (2020), Almesaeed, R., & Jedidi, A. (2021), Jayarajan, P., Kanagachidambaresan, G. R., Sundararajan, T. V. P., Sakthipandi, K., Maheswar, R., & Karthikeyan, A. (2020), Mittal, M., Iwendi, C., Khan, S., & Rehman Javed, A. (2021), Alsafi, S., & Talab, S. A. (2020).

The LEACH protocol uses a hierarchical-based cluster method and is generally implemented on outdoor application-specific sensor networks. This protocol assigns a maximum number of sensor nodes to each set of sensors, which are divided into several categories. For instances when detected properties, like temperature, rapidly vary, a hierarchical protocol called TEEN was developed. APTEEN is a protocol for improving LEACH. It improves LEACH by including HT, ST, and counting time, all of which may respond to crises and limit the amount of data that the node transmits. Location-based routing methods, which take advantage of nodes' locations to speed up communication, are utilized in WSN. Other names for it include position-based routing systems and geographical routing protocols. These protocols extend network life and use less energy. This paper discusses the Location-Based Energy-Efficient Routing Protocol for WSN. The main contribution of the paper is as follows: Subramani, N., Mohan, P., Alotaibi, Y., Alghamdi, S., & Khalaf, O. I. (2022), Anand, R., Singh, J., Pandey, D., Pandey, B. K., Nassa, V. K., & Pramanik, S. (2022), Abdul-Wahab, Y., Alhassan, A. B., & Salifu, A. M. (2020).

- The Haversine-based K-means Clustering algorithm (HKMC) is a variation of the traditional K-means clustering algorithm that incorporates the Haversine formula to cluster data points with latitude and longitude information.
- The data aggregation module receives the compressed data, multiplies the DWT query output by weight, and introduces a weighted DWT (WDWT). The weighting can be used to highlight important features or reduce the impact of less relevant information in the data.
- Within each cluster, a cutting-edge hybrid optimization model- teaching-learning soccer league optimization (TLSLO), combining teaching learning-based optimization (TLBO) and soccer league competition (SLC) (proposed), is utilized to select the optimal CH. The TLBO-based imitation operator is introduced to efficiently choose the CH from the available nodes.
- The integration of deep learning models, such as the proposed «recurrent artificial neural network» (RANN), which combines recurrent neural network (RNN) and artificial neural network (ANN) techniques, is a promising approach for predicting energy consumption patterns. The output of the RNN is given to the ANN model as an input.

The remaining portions of the paper is structured as follows: Section 2 of this article talks about related research on WSN. Section 3, describes the proposed location-based routing for WSN. Section 4, discussed the results obtained for the proposed model. The conclusion is presented in Section 5.

Literature Review

The most recent publications on the various routing methods used in WSNs are included in this section.

The authors have suggested a routing technique for heterogeneous clustered networks. First, the enhanced WPA was used to optimize the deployment of heterogeneous nodes. The CLWPA and the heterogeneous network routing method were integrated. Finally, the performance of the method was compared to three other popular routing algorithms using simulation tests. The simulation findings show that the CLWPA improves node energy consumption uniformity, ensures that all nodes survive longer, successfully suppresses the phenomena of early cluster head death, and concentrates node death time, Xiu-Wu, Y. U., Hao, Y. U., Yong, L., & Ren-rong, X. (2020).

The authors have introduced the E-ALWO algorithm, which was utilized to establish a reliable and routing mechanism that uses less energy by forwarding the data packets to the recipient. The E-ALWO algorithm is created via combining the EWMA concept with ALO and WOA. The proposed model performs the CH routing process in such a manner that the energy and latency restrictions were used to determine the CH using the ALWO method. The proposed E-ALWO method, which was based on the fitness measure, was used to find the quickest and safest way for data transfer, Suresh Kumar, K., & Vimala, P. (2021).

The authors have employed the PSO approach to build the cluster in the WSN, and a fuzzy-based E-FEERP was presented to best use battery energy, the average SN to BS node density, distance, and communication quality to send data from CH to the BS. Simulated network performance metrics, including energy consumption, throughput, packet delivery ratio, RE, load balancing ratio, and network lifespan, show improvement when compared to those of existing techniques, Narayan, V., Daniel, A. K., & Chaturvedi, P. (2023).

The authors have recommended a cutting-edge technique called MCH-EOR. It deals with the limited lifetime of the sensor nodes of the neighboring base station. The high volume of people using the washbasin from different cluster heads contributes to the low-life issue. The MCH-EOR employs a method for choosing energy-efficient cluster heads that accounts for a number of factors, including coverage, residual energy, cost, and closeness. Optimising SailFish identified the best path from the CH to the sink node, Mehta, D., & Saxena, S. (2020).

The authors have suggested a unique Q-learning-based energy-efficient routing technique that is mindful of data aggregation. The recommended approach makes use of reinforcement learning to maximize rewards at each sensor node and determine the best path. The efficiency of the data aggregation based on sensor type, communication energy, and node residual energy were used to define rewards. Next, sensor-type dependent reward aggregation was used. Employed simulations to assess the suggested routing technique's effectiveness and compare it to the performance of the more well-established energy-aware routing algorithms, Yun, W. K., & Yoo, S. J. (2021).

The authors have introduced a fuzzy control-based EARP in which the optimal forwarder node was selected using the fuzzification, fuzzy inference, and defuzzification methods. The suggested protocol creates a fuzzy control model using connection quality and remaining node energy. The results of the simulation indicate that the proposed EARP performs better, including an improvement in network lifespan and increased data transmission reliability, Wang, X., Zheng, G., Ma, H., Bai, W., Wu, H., & Ji, B. (2021).

The authors have introduced an improved energy-efficient selection of numerous mobile sink paths in WSNs. The WSN was first segmented into several zones using the QAZP method. The suggested QAZP splits the networks into zones utilizing the mobile anchor nodes based on the residual energy of the nodes inside the specified limits. In order to choose the RP for data transmission based on each node's hop distance and the quantity of data packets delivered, a weight was assigned to each node after that. WRP was implemented to give each node a weight, Senthil Kumar, V., & Prasanth, K. (2020).

The authors have suggested a WSN sink mobility method based on the Genetic Algorithm (GA). An ideal number of clusters are formed in the network region, and a sink movement trajectory is constructed there. The GA process determines the best sink sites for each cluster. The mobile sink collects data from the associated clusters' nodes when it reaches the best sink sites. The least amount of node energy is used in data transmission at the ideal sink position. The GA initialises a population of chromosomes for selecting the best sink position for a cluster, Singh, M. K., Amin, S. I., & Choudhary, A. (2021).

The authors have combined Manhattan and Euclidean to create unique frequency hopping and average hop-length using the DV-Hop wireless sensor network location mode, was proposed by the NSGA-II approach for iterative optimization. The adaptability of the algorithm is tested through simulated experiments in isotropic and anisotropic networks. As demonstrated by the results, MDV-Hop can greatly increase the positioning flexibility of sensor arrays in isotropic and anisotropic systems and rapidly reach highly accurate positioning without introducing hardware or traffic, Huang, X., Han, D., Weng, T. H., Wu, Z., Han, B., Wang, J., Cui, M., & Li, K. C. (2022).

The authors have introduced a unique energy-conscious and dependable routing technique is suggested. Under certain reliability constraints, it is intended to increase the lifespan of WSNs by using multi-hop routing techniques, in which the source node forwards the packet to the Base Station (BS) via additional nodes functioning as relays. The optimal path is the one where the packet has the best likelihood of succeeding upon arriving at the base station and the nodes' residual energy distribution is as uniform as is practical, Almazaideh, M., & Levendovszky, J. (2020).

Problem statement

WSNs are widely used for various applications, and researchers have proposed several routing algorithms and techniques to improve the performance of these networks.

The existing literature (as outlined in the referenced studies) presents a range of routing methods, such as WPA optimization, ALO and WOA integration, PSO clustering, fuzzy-based E-FEERP, MCH-EOR, Q-learning-based routing, fuzzy control-based EARP, mobile sink path selection, GA-based sink mobility, and novel frequency hopping techniques. While these studies demonstrate advancements in various aspects of WSN routing, they lack a comprehensive and up-to-date comparative analysis of these techniques, making it challenging for researchers, engineers, and network designers to choose the most suitable routing approach for specific application scenarios.

However, there are several challenges that still need to be addressed in the field of WSNs. These challenges include optimizing energy efficiency, network reliability, and data transmission in heterogeneous and clustered networks, as well as addressing issues related to node energy consumption, cluster head selection, and data aggregation. Additionally, there is a need to improve the network's overall performance metrics such as throughput, load balancing, packet delivery ratio, and network lifespan.

Proposed Methodology

A small or large number of nodes called sensor nodes compose sensor networks. These nodes come in different sizes, and depending on their size, the sensor nodes operate effectively in various domains. WSNs include sensor nodes that are uniquely constructed in a manner that is conventional, such that they have a radio transceiver that generates radio waves, a variety of wireless communication devices, a microprocessor that controls the monitoring, and an energy supply, like a battery. Using sensors of various dimensions, the complete network operates concurrently, and by utilising a routing algorithm, they are primarily focused on delivering data from the source to the destination nodes.

Before transmitting the packets, location-based services are used to determine the location of the targeted node. By providing all layers with positioning services, it is usually possible to determine the location of neighbors. Each node transmits one of these systems on a regular basis. It is the responsibility of the nodes to maintain a specific node's network position. Greed, restricted-directional, and hierarchical forwarding are the three main types of forwarding strategies that have been applied. For delivering the packets over the location services in the first method, the transmitting nodes use the destination node's estimated positioning data. This kind of forwarding excludes the setup and maintenance phases of the routes. The packets are again transmitted to the targeted node when the neighboring nodes receive them from a broadcasting node. This keeps going till it reaches the target nodes.

The choice is made in accordance with the algorithm's specifications. Using a hierarchical network topology, hierarchical forwarding is another form. Here, the data may be processed and forwarded by the nodes with the highest energy, while the nodes with lesser energy may send it. As a result, the routers' energy efficiency is improved, and the network's lifespan is prolonged. The location-based clustering and routing is shown in Figure 1.

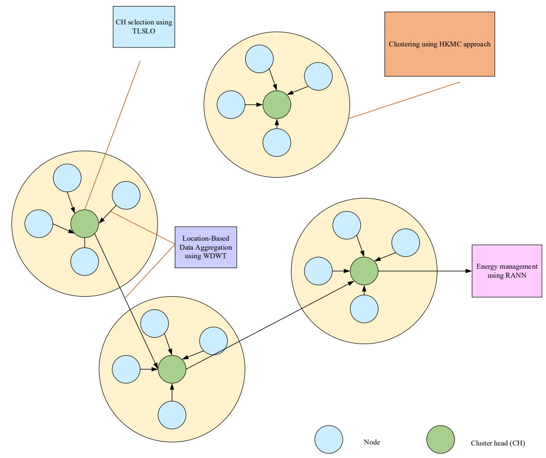

Figure 1: Block diagram of the proposed location-based routing

Data collection and Network Initialization

Designing energy-efficient routing protocols is crucial since WSNs are made up of tens of thousands of unattended, low-power, and resource-constrained sensor nodes. Several sensor nodes within communication range of one another constitute a cluster, which makes clustering-based routing protocols more helpful in the context of energy efficiency. A CH, which manages all of the nodes in a cluster, is present in every cluster. A WSN may have several base stations (BS), also known as sinks, that interact with other networks. In order to transfer data to the BS, a CH gathers information that has been received from all cluster members. Along with CH, a cluster may also have gateway nodes that are employed for inter-cluster communication. As a result, clustering methods extract a small quantity of accurate, valuable information from a vast amount of raw sensed data while consuming less energy in the process. In the literature, static sensor node clustering techniques for WSN are the norm. Because mobile sensor nodes are necessary for WSN applications, including habitat monitoring, wildlife monitoring, target tracking, and combat surveillance, these protocols are ineffective in these contexts. Furthermore, these protocols are ineffective since they merely presume that each sensor node is aware of its position rather than supporting the localization of sensor nodes. For instance, the WSN's typical static clustering protocol is the low energy adaptive clustering hierarchy (LEACH) Protocol.

Sensor Nodes

These are the specific gadgets or units that have sensors to gather data. These nodes can be any number of sensors, including ones that measure temperature, humidity, motion, or other factors important to the study topic.

Location Information

In order for the network to be spatially aware, each sensor node will have its own position information. Knowing each node's specific geographic location is made easier with the aid of this information. It's essential for keeping track of and analyzing data depending on its source or position in the study area.

Network Initialization

A sensor network must have certain parameters configured during setup to ensure appropriate operation. Some essential criteria include:

- *Communication Range*

This setting establishes the greatest possible distance that a sensor node may go while transmitting data to other nodes or a centralized data gathering station. How far the signals may go throughout the network must be determined.

- *Data Transmission Rates*

Data transmission speed between nodes is controlled by this parameter. Setting suitable data transfer speeds depending on the unique requirements of the research study is crucial. Lower rates can save energy, but higher rates can be required for real-time monitoring.

- *Energy Levels*

Monitoring and controlling the energy levels of sensor nodes is essential since they frequently run on batteries. To reduce the amount of energy consumed to transmit data, the energy parameter may be used to specify thresholds or criteria for when and how often nodes should do so.

- *Initial Data Readings*

The sensor nodes begin gathering data readings after the network is configured and its settings are initialized. The study project's initial data readings serve as its beginning point. The information gathered at this point may serve as background or baseline information that will serve as a benchmark for further measures. Researchers utilise this preliminary information to comprehend the environmental circumstances and actions of the system or region under observation.

Node Clustering Using Haversine based K-means Clustering algorithm (HKMC)

To enhance the K-Mean clustering method, several research projects have been conducted in the past. They just sought to address the shortcomings of the earlier presented approach and enhance the clustering outcome. We worked on the issue to identify the original cluster (R). We also made an effort to locate the original centroid. In the final stage of our algorithm, we look for workable points to reduce calculation. We discovered a powerful modified k-means algorithm when we merged all of these ideas.

K-Means algorithm is one of the most used clustering algorithms. It is widely used in many different sectors, such as ad hoc networks, sensor networks, and data mining. This is an unsupervised learning method that is simple to use and organizes a collection of data into the K initial clusters. Reducing the distance between the cluster leader and its members is its main objective. K clusters are originally chosen by the algorithm. There are K clusters to be created from a set of points with coordinates 1 ≤ j ≤ N. This is accomplished by having K-Means randomly select K points xi, each of which belongs to cluster C, as centroids from the data set with 1 ≤ i ≤ K of points. The application then assigns each collected data point to the nearest centroid. Totaling the squared distances between each cluster serves as the objective function upon which this process is built. The objective function is used to calculate the result is given in Eq. (1).

$$avgmin_c \sum_{i=1}^{K} \sum_{x_j \in C_i} d(x_j, u_i) = avgmin_c \sum_{i=1}^{K} \sum_{x_j \in C_i} \left| x_j - u_i \right|^2 \tag{1}$$

In K-means clustering, the Haversine distance formula is used when dealing with geographic data, such as latitude and longitude coordinates. The Haversine distance is a modification of the standard Euclidean distance, designed to account for the curvature of the Earth's surface when measuring distances between points on a sphere, like the Earth. The Haversine distance formula provides more accurate distance measurements for geographic data. It considers the spherical shape of the Earth, which is important when working with latitude and longitude coordinates. Using the standard Euclidean distance in such cases would result in distortions and inaccurate clustering. When measuring distances between spherical objects, the haversine distance is correct. Eq. (2) mentions the equation to determine the distance $d(x_j, u_i)$ between a data point and a centroid.

$$d(x_j, u_i) = 2\gamma \sin^{-1} \sqrt{sin^2\left(\frac{x_1 - u_1}{2}\right) + \cos(x_1)\cos(u_1)\, sin^2\left(\frac{x_2 - u_2}{2}\right)} \tag{2}$$

Centroid is represented by x_1 and x_2 where u_1 and u_2 are data points. In such case the cluster centroid's distance from the point is determined by $d(x_j, u_i) = \left|x_j - u_i\right|^2$. The coordinates of the point are x_j, the centroid is u_i, where $i = 1$ and the number of clusters is K. As shown in Figure 2, after placing the points in each cluster, the K-Means method adjusts the location of each centroid by utilizing Eq. (3):

$$u_i = \frac{1}{|C_i|} \sum_{j \in C_i} x_j, \forall i \tag{3}$$

As depicted in Figure 2, the clusters finally take shape. The accompanying pseudo-code for the modified K-means clustering is given in Algorithm 1.

Consequently, K-Means has successfully tackled several problems that arose throughout the WSN clustering

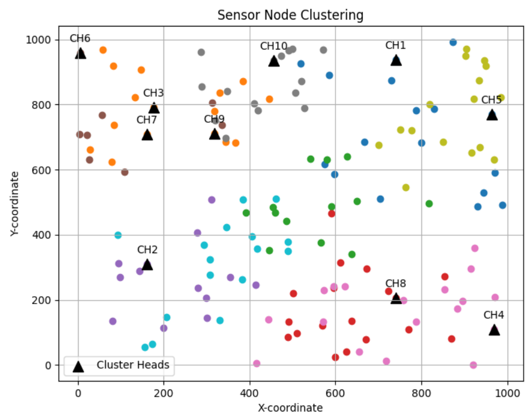

Figure 2: Clustered formed in the WSN

procedure. There are certain disadvantages, nevertheless, especially with the random selection of the initial number of clusters and the objective function for assigning each point to a cluster because of the frequent changes in the number of connected cars and the network structure.

CH selection using Teaching-Learning Soccer League Optimization (TLSLO)

For the purpose of resolving optimization issues, it is possible to simulate competitions between football clubs for league success and between players for being an SP or SSP. Similar to a football league where every player strives to be the best, each solution vector in an optimization issue seeks the global optimal location. Accordingly, it is conceivable to think of the Super Star Player (SSP), the local optimum, and the solution vector for each player in a league, as well as the Star Player (SP) for each club. This league consists of M teams, each of which has two types of players: fixed players and substitutes. Each player in the SLC algorithm is a solution vector. Each player has an objective function computed for them that represents their power and may be determined using the formula below:

$$PP(i,j) = C(i,j) \; i \in team, j \in player, C = Objective\ function \quad (4)$$

The average power value of a team's fixed players is said to represent its overall strength. In a knapsack issue, objective function values with bigger values represent strong players (PP). The total strength of a team is considered to be represented by the average power value of its set members. Finding a Team's Power (TP) is made easy with the following formula.

$$TP(i) = \left(\frac{1}{nF}\right)\sum_{j=1}^{nF} PP(i,j) \quad (5)$$

Where the total number of fixed players on the i^{th} team is represented by nF. The team with superior strength often prevails in each game. Each team's chance of winning a game is calculated using:

$$Pv(k) = \frac{TP(k)}{(TP(i)+TP(k))} \quad (6)$$

Algorithm 1: Pseudo-code for the HKMC

Input:
k (Number of centroids)
N (Set of data points)
Ck (List of centroids randomly assigned)
Output:
Clusters (Set of clusters along with their respective centroids)
Begin
1: Iterate
2. For each data point in the set N
3: Initialize an array distances[k]
4: For i from 1 to k do
5: distances $[i]$ = Calculate the distance using Eq. (1).
6: End for
7: The data point should be assigned to the cluster that is connected to the closest centroid: cluster_index = argmin(distances)
8: End for
9: For each cluster in Ck:
10: Calculate the new centroid position using Eq. (3).
11. End for
12. Repeat until all data points belong to a cluster or the maximum number of iterations is reached.
13. End

$$Pv(i) = \frac{TP(i)}{(TP(i)+TP(k))} \quad (7)$$

Where Pv represents the probability of success. It should be observed that $Pv(k)$ and $Pv(i)$ add up to 1. Each game has a winner and a loser, and certain players (solution vectors), such as fixed and substitution, go through modifications. These adjustments, intended to boost individual and team performance, are modeled using the following operators: Provocation operator and imitation operator. These are the steps involved in the SLC process:

Step 1. Set the problem and the algorithm's initial parameters

The knapsack issue is explained in Step 1 as follows:

$$Max\ C = \sum_{k=1}^{N} v_k X_k \quad (8)$$

Subject to $\sum_{i=1}^{N} w_k X_k \leq W$ (9)

Where $X_k = 0 \ or \ 1 \ and \ k = 1,2,\dots,N$. The N is the total number of items, v_k is the item's profit, w_k is its weight, and W is the knapsack's capacity. The value X_k Indicates if item k is in the knapsack or not. The objective function of the knapsack problem is the profit function. The decision variables in the problem are whether an item is present or absent. Following that, the number of seasons (n Season), teams participating in the league (nT), fixed players (nF), and substitutes (nS) are established.

Step 2. Samples generation

To find out how many players are in a league overall, apply the following Eq. (10):

$$n\ players = nT \times (nF + nS) \quad (10)$$

In most problems, it is suggested that $4 \leq nT \leq 10$ and $nF = nS = 10$. The number of teams working on high-dimensional challenges should be increased in order to foster variety. This stage involves creating as many randomly generated solution vectors for the league's participants, one for each player. It's possible to construct the SLC's initial population via:

$$FP_i \text{ or } RP_i = X_{i\,min} + \tau_i(X_{i\,max} - X_{i\,min}) \quad (11)$$

Where $X_{i\,min}$ and $X_{i\,max}$ represent the maximum and minimum boundaries and τ_i is a random vector with a value of 0 or 1. The lowest and maximum bounds in this issue are 0 and 1, respectively. Consequently, the matrix TEAM that is created at random is provided as follows.

$$TEAM = \begin{bmatrix} FP_1 \\ FP_2 \\ \vdots \\ FP_{nF} \\ S_1 \\ S_2 \\ \vdots \\ S_{nS} \end{bmatrix} = \begin{bmatrix} XF_1^1 & XF_2^1 & \dots & XF_N^1 \\ XF_1^2 & XF_2^2 & \dots & XF_N^2 \\ \vdots & \vdots & \dots & \vdots \\ XF_1^{nF} & \dots & \dots & XF_N^{nF} \\ XR_1^1 & XR_2^1 & \dots & XR_N^1 \\ XR_1^2 & XR_2^2 & \dots & XR_N^2 \\ \vdots & \dots & \dots & \vdots \\ XR_1^{nS} & \dots & \dots & XR_N^{nS} \end{bmatrix} \quad (12)$$

Then, a penalty cost associated with each solution vector (player's power) and the goal function (profit) are ascertained. In this study, restricted 0-1 knapsack issues are addressed using a penalty function technique. In order to maximise efficiency, it applies the penalty on unfeasible solution vectors and searches the viable region.

$$Max\ f(x) = \sum_{k=1}^{N} v_k X_k - \lambda \times \max\left(\frac{\sum_{k=1}^{N} w_k X_k}{W} - 1, 0\right) \quad (13)$$

Where the punishment coefficient is denoted by λ. The fitness value of the best impracticable choice is certain to be lower than the fitness score of the worst feasible choice by this enormous constant.

Step 3. Team evaluation

All players are assigned to teams in this stage and are placed according to their calculated power. Each team's strength is determined by the average power of its set players. Players are placed into the team's fixed and substitute positions depending on their abilities in this stage. The value of the fixed player cost amount is equal to the team's power.

Step 4. League start

In this phase, the acting players of the winning teams are targeted by the provocation and imitation operators accordingly. All conceivable combinations of the league's teams are entered into competitions. Detailed definitions of operators are provided in the next section.

- *TLBO based Imitation operator*

The winning team's FP copies the league's SSP as well as the SP in their own squad to improve their performance going forward. In a manner similar to this, the winning team's fixed players' solution vectors typically favour their own team's solution as well as the best option for the league. The following formulae carry out imitation in the SLC algorithm:

$$FP(i,j) = FP(i,j) + \tau_1(SSP - RP1(i)) \quad (14)$$

$$FP(i,j) = FP(i,j) + \tau_1(SP(i) - RP1(i)) \quad (15)$$

$$FP(i,j) = FP(i,j) + \tau_1\big(RP2(i) - RP1(i)\big) + Difference\_Mean_{RP} \quad (16)$$

Where τ_1 is a uniformly distributed random integer between 0.2 and 0.8 points. The i^{th} team's star player is designated as $SP(i)$, whereas $RP1$ and $RP2$ are arbitrary random players and $FP(i,j)$ denotes the i^{th} team's j^{th} fixed player. The result vector of SSP is the direction in which the winning team's solution vector of fixed players (FP) initially moves (Eq. 14). The mean value of the RP_i is calculated using the TLBO algorithm to improve the prediction accuracy. At any iteration i, let RP_i play the nasty character and T_i the instructor. T_i will attempt to elevate mean RP_i to its own level, making T_i the new mean (also known as Mnew). Using the difference between the old and new means provided by, the solution is revised.

$$Difference\_Mean_{RP} = r_i(RP_{new} - T_F * RP_i) \quad (17)$$

If at this particular location, the newly constructed solution vector was better than the prior one, the former solution vector is replaced. In the absence of such a change, the solution vector shifts in the direction of the SP resultant vector in Eq. (15). It gets changed out with a fresh solution vector if this solution is superior to the previous one. If not, the solution vector is forced to travel in the direction of the resulting vector of a team member chosen at random, using Eq. (16). The player remains in place with no modification if none of the suggested motions resulted in a better solution vector.

- *Provocation operator*

A replacement of a winning side (S) must perform at a level equivalent to the average overall performance score of the fixed individuals on their team in order to be eligible to be considered a fixed player. The provocative operator in the SLC algorithm executes this procedure, which is explained by

$$S(i,j) = G(i) + X_1(G(i) - S(i,j)) \quad (18)$$

$$S(i,j) = G(i) + X_2(S(i,j) - G(i)) \quad (19)$$

Where $G(i)$ is the mean value of the resulting vectors for fixed participants in the i^{th} team, and $X_1 \sim U(0.9,1), X_2 \sim U(0.4,0.6)$, are uniformly distributed random values. $S(i,j)$ is the j^{th} replacement for the i^{th} team. The weakest substitute player on the winning side first moves their solution vector in the direction of the fixed players' gravity center (Eq. 19). It is switched to the new solution vector if the recently generated one proves to be better than the old one for this new point; otherwise, it is left alone. Otherwise, the aforementioned individual will slide backward towards the gravitational center.

Step 5. Update the League

After every season, players are categorized according to their present power. The top teams in the league standings receive the best players, average teams receive middle-tier players, and bottom-tier teams receive the worst players before the next season begins.

Step 6. Check the stopping criterion

Until the n-Season termination criterion is satisfied, Steps 3, 4, and 5 continue in this phase of the technique.

Location-Based Data Aggregation using Weighted DWT

A mathematical method for analyzing signals and images is called the Weighted DWT. A signal or picture is broken down into a collection of wavelet coefficients at various sizes, which can reveal both low-frequency and high-frequency information. The DWT is typically implemented using a filter bank approach. The DWT of a signal $x[m]$ at a particular scale j and position k is computed as follows:

$$W_{j,k} = \sum_m x[m] \,.\, \psi_{j,k}[m] \tag{20}$$

Where $W_{j,k}$ represents the wavelet coefficient at scale j and position k, $x[m]$ is the input signal, $\psi_{j,k}[m]$ is the wavelet basis function (wavelet) at scale j and position k. The summation is typically performed overall values of m in the range of the wavelet function. The data aggregation module receives the compressed data and multiplies the DWT query output by weight. The minimum value of the data from the DWT is obtained when the query is supplied, as described in the preceding section, such that the data is multiplied by the weight. The weight is determined by taking the logarithmic function of the difference between the data recordings at a specific period. The combined information is therefore shown as,

$$P_t = \omega_t * W_{j,k} \tag{21}$$

$$\omega_t = [1 - \log(t_2 - t_1)] \tag{22}$$

Where ω_t is the weight function and t represents the time. To finalise data aggregation and prepare the data for transmission to the cluster head, the aggregated data is next submitted to the weighted DWT technique. Last but not least, the data is sent to the sink node in a method that uses less energy for data connection.

Deep Learning for Predictive Energy Management

The Recurrent Artificial Neural Network (RANN) is a hybrid deep learning model that integrates the capabilities of RNN and ANN. RNNs are known for their ability to model sequences and time-series data, while ANNs are versatile for pattern recognition and prediction. The combination of these two architectures can enhance the understanding and prediction of energy consumption patterns within a WSN.

ANN

A network of artificial neurons makes up an artificial neural network (ANN), which is capable of processing inputs, adjusting internal states in response to those inputs, and computing outputs based on those inputs and internal states. Learning can change the weights inside these artificial neurons. In the neural network model, the value of the output layer is gradually calculated from the input layer by using the output of the prior layer as the input for the subsequent layer. The ANN uses the following syntax:

$$y = f\big(f\big(x_i w_{ij} + b_j\big) w_{jk} + b_k\big) \tag{23}$$

Eq. (23), weight w_{ij} is the activation function, which demonstrates how the input variable x_i is multiplied by weight w_{ij} and added with bias b_j, this layer's output is the inputs for the following layer, and the end result y is the forecast value.

RNN

Traditionally used RNNs combine supervised and unsupervised learning. This model's input sequence data might have a length as long as its depth. The design of the RNN model consists of a feedback loop that connects each layer with the capacity to retain information from the previous input. As a result, it may make the model more trustworthy. In this model, $x(t)$ is the input layer with index i at time t, and $h(t-1)$ is the hidden layer with index s at time $t-1$. At time t, layer $h(t)$ with index j is concealed. $Y(t)$ is the output layer with index e at time t. The hidden layer with indexes i and j is connected to the input via the weight matrix U. Layers with the indices s and j are joined by the weight matrix W, that were previously concealed. The number of input units is m, and the weight matrix V connects the hidden layer and the output layer with the index j, e. There are n hidden units. There are k output units. The calculations of an RNN are governed by the following formulae. The input at the present step and the previously concealed state are used to determine $h(t)$ in the first step:

$$h(t) = f(Ux(t) + Wh(t-1)) \tag{24}$$

Where the nonlinear function f is one like tanh or ReLU. The first hidden state, which is commonly initialised to all zeroes, must be calculated using $h(t-1)$. The output at step t is determined as $y(t)$ in the second step using the formula:

$$y(t) = f(Vh(t)) \tag{25}$$

For recurrent networks, the following formulas are used to derive $h_j(t)$ and $y_e(t)$:

$$h_j(t) = f\big(\textstyle\sum_i^m x_i(t)\, u_{ij} + \sum_s^n h_s(t-1) w_{sj}\big) \tag{26}$$

$$y_e(t) = f\big(\textstyle\sum_{j=1}^k h_j(t) v_{je}\big) \tag{27}$$

Concatenation is the process of connecting these results along a predetermined axis to produce a composite representation of the data. The model's output is the last prediction it makes, which is frequently a binary classification.

Result and discussion

In this section, the results obtained for the proposed model are compared with the existing techniques. The performance of the existing techniques like CNN, Long Short-Term Memory (LSTM), RNN, and ANN are evaluated in terms of performance metrics like energy consumption, energy efficiency, latency, network life time, and data delivery ratio. The energy consumption is compared in Table 1.

Table 1: Comparison of the performance metrics for simulation time 100

Metrics	*CNN*	*RNN*	*LSTM*	*ANN*	*PROPOSED*
Energy consumption	61.75428	52.49008	69.21156	49.80706	40.32007
Energy efficiency	161000.7	1002570	142362.8	1239053	1500000
Latency	13.06257	12.84857	13.92563	11.91647	4.217153
Network lifetime	1.7E+09	1.8E+09	1.6E+09	2.05E+09	2.27E+09
Data delivery ratio	0.912634	0.952634	0.903153	0.962234	0.983693

Table 2: Comparison of the performance metrics for simulation time 200

Metrics	*CNN*	*RNN*	*LSTM*	*ANN*	*PROPOSED*
Energy consumption	64.62988	53.26194	72.56296	51.23165	42.32007
Energy efficiency	1136585	992214.3	1092563	1036220	1490850
Latency	21.2655	19.0316	22.2965	12.26366	5.982551
Network lifetime	1.52E+09	1.8E+09	1.53E+09	1.91E+09	2.2E+09
Data delivery ratio	0.906594	0.946924	0.892316	0.951365	0.980006

Table 3: Comparison of the performance metrics for simulation time 300

Metrics	*CNN*	*RNN*	*LSTM*	*ANN*	*Proposed*
Energy consumption	65.95465	53.86555	72.69897	51.94646	43.25165
Energy efficiency	1092349	985009.2	1051230	1003652	1402561
Latency	21.91635	19.56987	23.64645	12.94157	7.56165
Network lifetime	1.49E+09	1.7E+09	1.51E+09	1.8E+09	2.16E+09
Data delivery ratio	0.900053	0.932067	0.890237	0.949653	0.961635

Table 1 compares several neural network models and a suggested model, all evaluated during a 100-simulation-time period. The metrics evaluated include energy consumption, energy efficiency, latency, network lifetime, and data delivery ratio. Among the models, the proposed model stands out with the lowest energy consumption (40.32007 units), highest energy efficiency (1,500,000 units), significantly lower latency (4.217153 units), longest estimated network lifetime (2.27E+09 units), and the highest data delivery ratio (0.983693). Conversely, the traditional CNN, RNN, LSTM, and ANN models exhibit varying performance across these metrics, demonstrating the potential advantages of the Proposed model in terms of energy efficiency, low latency, network longevity, and reliable data delivery, making it a promising choice for various applications with stringent performance and energy constraints.

Table 2 provides a comparative assessment of performance metrics for five neural network models, namely CNN, RNN, LSTM, ANN, and a Proposed model, conducted over a simulation time of 200. The metrics considered encompass energy consumption, energy efficiency, latency, network lifetime, and data delivery ratio. Notably, the proposed model consistently exhibits superior performance, with the lowest energy consumption (42.32007 units), highest energy efficiency (1,490,850 units), shortest latency (5.982551 units), longest projected network lifetime (2.2E+09 units), and the highest data delivery ratio (0.980006). Conversely, the traditional CNN, RNN, LSTM, and ANN models present varying results across these metrics, underscoring the benefits of the proposed model in terms of energy efficiency, low latency, network durability, and dependable data delivery, making it a compelling choice for applications with extended simulation times and stringent performance requirements.

Table 3 presents a comparative analysis of performance metrics for five neural network models, namely CNN, RNN, LSTM, ANN, and a Proposed model, evaluated in a simulation time of 300. The metrics considered encompass energy consumption, energy efficiency, latency, network lifetime, and data delivery ratio. Notably, the proposed model consistently outperforms the other models with the lowest energy consumption (43.25165 units), highest energy efficiency (1,402,561 units), lowest latency (7.56165 units), longest projected network lifetime (2.16E+09 units), and the highest data delivery ratio (0.961635). Conversely, the traditional CNN, RNN, LSTM, and ANN models exhibit varying performance across these metrics, highlighting the advantages of the Proposed model in terms of energy efficiency, low latency, network durability, and reliable data delivery. These results make the Proposed model a

Table 4: Comparison of the performance metrics for simulation time 400

Metrics	*CNN*	*RNN*	*LSTM*	*ANN*	*Proposed*
Energy consumption	65.95465	53.86555	72.69897	51.94646	46.12645
Energy efficiency	1031257	989562	992651.4	982666	1392570
Latency	25.31163	20.88498	24.16342	15.12565	10.51654
Network lifetime	1.03E+09	1.51E+09	1.49E+09	1.79E+09	2.07E+09
Data delivery ratio	0.892647	0.928656	0.889565	0.942465	0.951165

Table 5: Comparison of the performance metrics for simulation time 500

Metrics	*CNN*	*RNN*	*LSTM*	*ANN*	*Proposed*
Energy consumption	67.98846	54.02615	74.56617	56.25654	50.96645
Energy efficiency	995562.5	980235.1	990592.7	972655.6	1376259
Latency	25.94652	22.03217	25.03116	15.94644	11.95164
Network lifetime	1.02E+09	1.5E+09	1.49E+09	1.76E+09	2E+09
Data delivery ratio	0.882269	0.920565	0.872362	0.93621	0.9496

compelling choice for applications with extended simulation times and stringent performance requirements, further reinforcing its suitability in resource-constrained scenarios.

A thorough comparison of the performance metrics for each of the five neural network models is shown in Table 4, including CNN, RNN, LSTM, ANN, and a proposed model, all evaluated under a prolonged simulation time of 400. The metrics considered encompass energy consumption, energy efficiency, latency, network lifetime, and data delivery ratio. Impressively, the proposed model consistently stands out with the lowest energy consumption (46.12645 units), highest energy efficiency (1,392,570 units), shortest latency (10.51654 units), longest estimated network lifetime (2.07E+09 units), and the highest data delivery ratio (0.951165). In contrast, the conventional CNN, RNN, LSTM, and ANN models demonstrate variable performance across these metrics, emphasizing the merits of the Proposed model in terms of energy efficiency, low latency, network longevity, and reliable data delivery.

Table 5 offers a comprehensive comparative analysis of performance metrics for five neural network models—CNN, RNN, LSTM, ANN, and a Proposed model—assessed under an extended simulation time of 500. These metrics encompass energy consumption, energy efficiency, latency, network lifetime, and data delivery ratio. Notably, the Proposed model consistently emerges as the standout performer with the lowest energy consumption (50.96645 units), highest energy efficiency (1,376,259 units), shortest latency (11.95164 units), longest estimated network lifetime (2E+09 units), and a high data delivery ratio (0.9496). In contrast, the traditional CNN, RNN, LSTM, and ANN models exhibit variable performance across these metrics, further emphasizing the superiority of the Proposed model in terms of energy efficiency, low latency, network longevity, and reliable data delivery.

Energy Consumption

When comparing the «Energy Consumption» metric across Tables 1 to 5 with increasing simulation times (100–500), a notable trend emerges. In all cases, as the simulation time extends, there is a general increase in energy consumption for all models, which is expected given the longer operational duration.

However, the key observation is that the proposed model consistently outperforms the other models by consuming the least energy at each simulation time point. This consistent trend underscores the remarkable energy efficiency of the Proposed model, as it manages to maintain low energy consumption even in longer simulations. It establishes the proposed model as a highly energy-efficient option, particularly suitable for applications demanding prolonged operational periods while minimizing energy consumption when compared to traditional models like CNN, RNN, LSTM, and ANN.

Energy Efficiency

When comparing the «Energy Efficiency» metric across Tables 1 to 5, which correspond to increasing simulation times from

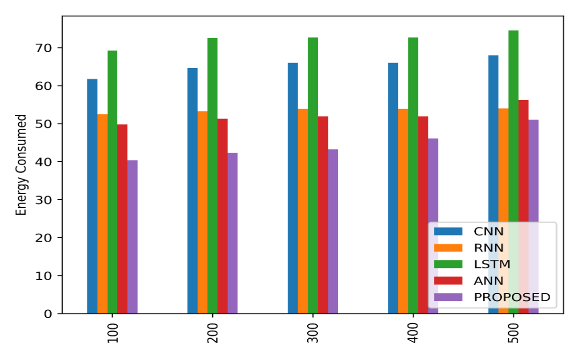

Figure 3:

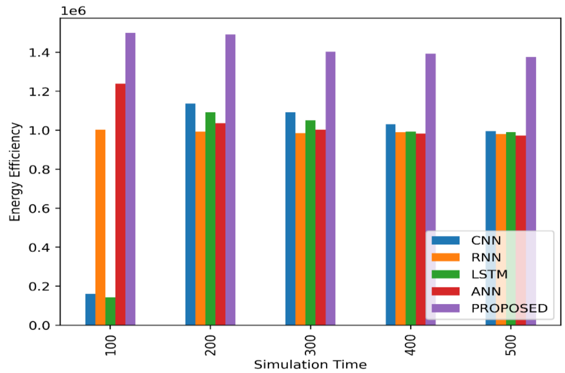

Figure 4:

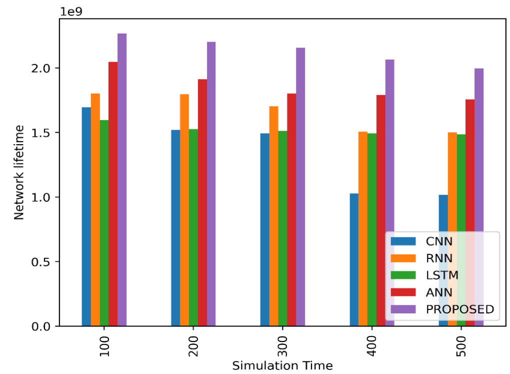

Figure 6:

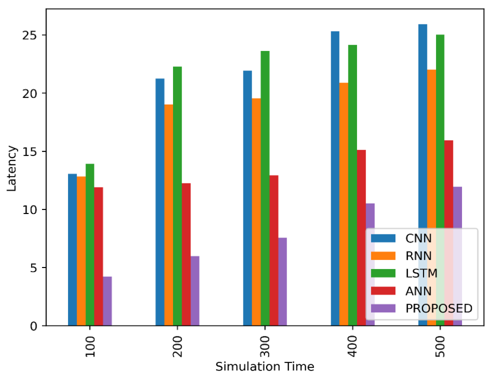

Figure 5:

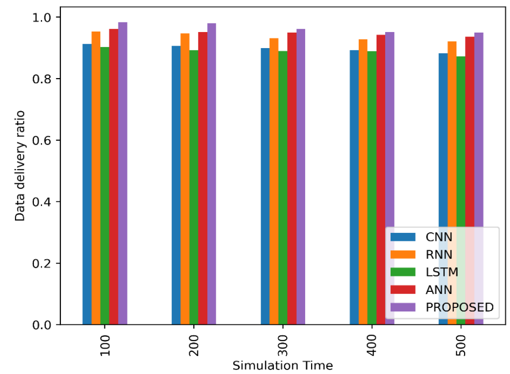

Figure 7:

100 to 500, a notable pattern becomes evident. While there are fluctuations in energy efficiency for all models as the simulation time varies, the consistent observation is that the Proposed model consistently maintains the highest energy efficiency among the models at each time point.

This highlights the remarkable energy optimization capabilities of the Proposed model, as it excels in conserving energy even as the simulation duration extends. In essence, the Proposed model emerges as a standout choice for applications requiring not only extended operational periods but also the highest energy efficiency, setting it apart from conventional neural network models like CNN, RNN, LSTM, and ANN in terms of energy-efficient performance.

Latency

When comparing the «Latency» metric across Tables 1 to 5, which correspond to increasing simulation times from 100 to 500, a clear pattern emerges. While there are fluctuations in latency for all models as the simulation time varies, the consistent observation is that the Proposed model consistently maintains the lowest latency among the models at each time point.

This showcases the exceptional processing speed and efficiency of the Proposed model, as it consistently delivers results with minimal delay, even as the simulation duration extends. In essence, the proposed model stands out as a superior choice for applications demanding not only prolonged operational periods but also minimal latency, setting it apart from conventional neural network models like CNN, RNN, LSTM, and ANN in terms of low-latency performance.

Network lifetime

When comparing the «Network Lifetime» metric across Tables 1 to 5, covering simulation times from 100 to 500, a clear and consistent trend emerges. The Proposed model consistently maintains the longest estimated network lifetime among the models at each time point.

This demonstrates the extraordinary network longevity of the Proposed model, as it manages to ensure prolonged network operation even as the simulation duration extends. The Proposed model stands out as a robust choice for applications that require not only extended operational periods but also the assurance of a longer network lifespan, setting it apart from conventional neural network models like CNN, RNN, LSTM, and ANN in terms of network durability and sustainability.

Data delivery ratio

When comparing the «Data Delivery Ratio» metric across Tables 1 to 5, spanning simulation times from 100 to 500, a distinct pattern emerges.

Although there are variations in data delivery performance for all models as the simulation time extends, the Proposed model consistently maintains the highest data delivery ratio among the models at each time point. This highlights the remarkable reliability of the Proposed model in successfully delivering data, even in prolonged simulations. The Proposed model stands out as an optimal choice for applications where the dependable and high data delivery ratio is critical, setting it apart from conventional neural network models like CNN, RNN, LSTM, and ANN in terms of data delivery performance and ensuring reliable transmission of critical information.

Conclusion

In conclusion, this research underscores the paramount importance of energy efficiency in WSNs and introduces a comprehensive approach to enhance their performance in designated areas. The deployment of location-aware sensor nodes and the optimization of network parameters lay the foundation for an energy-efficient framework. Innovations in node clustering through Modified K-Means and cluster head selection via TLSLO reduce unnecessary communication, ultimately extending the network's lifespan. The integration of data compression techniques, notably the I-DWT, further conserves energy by minimizing data transmission sizes. Moreover, RANN predicts energy consumption patterns, enabling smart node scheduling and prolonged network lifetime. Extensive Python-based simulations confirm the superiority of this holistic approach over existing protocols, making it a promising solution for achieving efficiency and sustainability in resource-constrained WSNs across various applications.

References

Abdul-Wahab, Y., Alhassan, A. B., & Salifu, A. M. (2020). Extending the lifespan of wireless sensor networks: A survey of LEACH and non-LEACH routing protocols. *International Journal of Computer Applications*, *975*, 8887.

Almazaideh, M., & Levendovszky, J. (2020). Novel reliable and energy-efficient routing protocols for wireless sensor networks. *Journal of Sensor and Actuator Networks*, *9*(1), 5. https://doi.org/10.3390/jsan9010005

Almesaeed, R., & Jedidi, A. (2021). Dynamic directional routing for mobile wireless sensor networks. *Ad Hoc Networks*, *110*, 102301. https://doi.org/10.1016/j.adhoc.2020.102301

Alsafi, S., & Talab, S. A. (2020). Threshold sensitive energy efficient sensor network protocol. *International Journal of Academic Engineering Research (IJAER)*, *4*(11), 48-52.

Anand, R., Singh, J., Pandey, D., Pandey, B. K., Nassa, V. K., & Pramanik, S. (2022). Modern technique for interactive communication in LEACH-based ad hoc wireless sensor network. In *Software Defined Networking for Ad Hoc Networks* (pp. 55-73). Springer International Publishing. https://doi.org/10.1007/978-3-030-72306-1_5

Banerjee, A., Sufian, A., Sadiq, A. S., & Mirjalili, S. (2021). Minimum energy transmission forest-based geocast in software-defined wireless sensor networks. *Transactions on Emerging Telecommunications Technologies*, *32*(9), e4253. https://doi.org/10.1002/ett.4253

Del-Valle-Soto, C., Mex-Perera, C., Nolazco-Flores, J. A., Velázquez, R., & Rossa-Sierra, A. (2020). Wireless sensor network energy model and its use in the optimization of routing protocols. *Energies*, *13*(3), 728. https://doi.org/10.3390/en13030728

Gupta, N. K., Yadav, R. S., & Nagaria, R. K. (2020). 3D geographical routing protocols in wireless ad hoc and sensor networks: An overview. *Wireless Networks*, *26*, 2549-2566. https://doi.org/10.1007/s11276-019-02094-1

Huang, X., Han, D., Weng, T. H., Wu, Z., Han, B., Wang, J., Cui, M., & Li, K. C. (2022). A localization algorithm for DV-Hop wireless sensor networks based on Manhattan distance. *Telecommunication Systems*, *81*(2), 207-224. https://doi.org/10.1007/s11235-022-00965-w

Jayarajan, P., Kanagachidambaresan, G. R., Sundararajan, T. V. P., Sakthipandi, K., Maheswar, R., & Karthikeyan, A. (2020). An energy-aware buffer management (EABM) routing protocol for WSN. *The Journal of Supercomputing*, *76*, 4543-4555. https://doi.org/10.1007/s11227-019-03019-2

Kamarei, M., Patooghy, A., Alsharif, A., & Hakami, V. (2020). SiMple: A unified single and multi-path routing algorithm for wireless sensor networks with source location privacy. *IEEE Access*, *8*, 33818-33829. https://doi.org/10.1109/ACCESS.2020.2974086

Maheshwari, P., Sharma, A. K., & Verma, K. (2021). Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Networks*, *110*, 102317. https://doi.org/10.1016/j.adhoc.2020.102317

Mehta, D., & Saxena, S. (2020). MCH-EOR: Multi-objective cluster head based energy-aware optimized routing algorithm in wireless sensor networks. *Sustainable Computing: Informatics and Systems*, *28*, 100406. https://doi.org/10.1016/j.suscom.2020.100406

Mehta, D., & Saxena, S. (2022). Hierarchical WSN protocol with fuzzy multi-criteria clustering and bio-inspired energy-efficient routing (FMCB-ER). *Multimedia Tools and Applications*, *81*(24), 35083-35116. https://doi.org/10.1007/s11042-022-12339-4

Mittal, M., Iwendi, C., Khan, S., & Rehman Javed, A. (2021). Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using Levenberg-Marquardt neural network and gated recurrent unit for intrusion detection system. *Transactions on Emerging Telecommunications Technologies*, *32*(6), e3997. https://doi.org/10.1002/ett.3997

Narayan, V., Daniel, A. K., & Chaturvedi, P. (2023). E-FEERP: Enhanced fuzzy based energy efficient routing protocol for wireless sensor network. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-023-10699-y

Roopali, & Kumar, R. (2020). Energy efficient dynamic cluster head and routing path selection strategy for WBANs. *Wireless Personal Communications*, *113*, 33-58. https://doi.org/10.1007/s11277-020-07399-9

Senthil Kumar, V., & Prasanth, K. (2020). Weighted rendezvous planning on Q-learning based adaptive zone partition with PSO based optimal path selection. *Wireless Personal Communications*, *110*, 153-167. https://doi.org/10.1007/s11277-019-06817-0

Singh, M. K., Amin, S. I., & Choudhary, A. (2021). Genetic algorithm based sink mobility for energy efficient data routing in wireless sensor networks. *AEU-International Journal of Electronics and Communications*, *131*, 153605. https://doi.org/10.1016/j.aeue.2020.153605

Subramani, N., Mohan, P., Alotaibi, Y., Alghamdi, S., & Khalaf, O. I. (2022). An efficient metaheuristic-based clustering with routing protocol for underwater wireless sensor networks. *Sensors*, *22*(2), 415. https://doi.org/10.3390/s22020415

SureshKumar, K., & Vimala, P. (2021). Energy efficient routing protocol using exponentially-ant lion whale optimization algorithm in wireless sensor networks. *Computer Networks*, *197*, 108250. https://doi.org/10.1016/j.comnet.2021.108250

Wang, X., Zheng, G., Ma, H., Bai, W., Wu, H., & Ji, B. (2021). Fuzzy control-based energy-aware routing protocol for wireless body area networks. *Journal of Sensors*, *2021*, 1-13. https://doi.org/10.1155/2021/8891165

Wang, Z., Ding, H., Li, B., Bao, L., & Yang, Z. (2020). An energy-efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks. *IEEE Access*, *8*, 133577-133596. https://doi.org/10.1109/ACCESS.2020.3010817

Yun, W. K., & Yoo, S. J. (2021). Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks. *IEEE Access*, *9*, 10737-10750. https://doi.org/10.1109/ACCESS.2020.3049417

Xiu-Wu, Y. U., Hao, Y. U., Yong, L., & Ren-rong, X. (2020). A clustering routing algorithm based on wolf pack algorithm for heterogeneous wireless sensor networks. *Computer Networks*, *167*, 106994. https://doi.org/10.1016/j.comnet.2019.106994