



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 3, March 2025

Proposed Energy Awareness and Trust Based Hierarchical Routing Protocol (EATHRP) for Wireless Sensor Networks using Bio-Inspired and Enhanced Deep Learning Approaches

Velpula Ravindra Reddy, Dr. V. Raghunatha Reddy

Research Scholar, Department of Computer Science & Technology, S.K. University, Ananthapuramu,
Andhra Pradesh, India

Professor, Department of Computer Science & Technology, S.K. University, Ananthapuramu, Andhra Pradesh, India

ABSTRACT: Wireless Sensor Networks (WSNs) are sensor nodes that expand their connectivity to establish networks without the need for pre-existing networks. Energy consumption is a critical performance characteristic of WSNs and affects the lifespan of a network. This study proposes an Energy Awareness and Trust Based Hierarchical Routing Protocol (EATHRP) for Wireless Sensor Networks, which uses Bio-Inspired and Deep Neural Network to address the challenges of scalability, eco-friendly features, and reliable data transfer. Initially, the new scaling challenge required the standard deviation together with Manhattan-based farthest first K-means (SDM-FFKM) to create better and more consistent clusters. In addition, the HPWO (Hybrid Pufferfish Walrus optimization algorithm) is suggested for cluster head selection (CH) to address convergence difficulties. In addition, for hierarchical route planning, the optimal pathways were chosen using Deep Neural Network with Bidirectional Long Short-Term Memory (DNN-BiLSTM) and the Enhanced Coati Optimization Algorithm (ECO). Finally, the sensed data is securely transmitted to the Base Station (BS) using the cumulative distributed four-point curve cryptography (CDFPCC) method. The suggested method EATHRP protocol was further developed by experimental assessment, and the results were compared with current protocols such as LEACH, Fuzzy C-Means (FCM), Deep Kronecker Neural Network (DKNN), and Deep Neural Network (DNN). Metrics used to assess performance include Network longevity, Loss ratio, End-to-End delay, Delivery ratio, Buffer occupancy, Delay, Energy efficiency, and Network performance. The evaluation based on the important metrics of 95.5% energy efficiency, 90.2% throughput, 4.8% latency, and 95.3% network lifetime illustrate the success of the EATHRP protocol in optimizing energy consumption and overall network performance. All outcomes achieved by the proposed technique outperformed those of its existing counterparts.

KEYWORDS: Energy Awareness and Trust based Hierarchical Routing Protocol(EATHRP), Wireless Sensor Networks(WSNs), hybrid Puffer Fish Walrus Optimisation, Deep Neural Networks, Bidirectional Long Short-Term Memory, Enhanced Coati Optimisation Algorithm, Base Station(BS), Sink Node(SN), Cluster Head(CH).

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are composed of numerous tiny, inexpensive, and low-power sensor nodes[1]. These nodes collect important data from a specific area and send it to a central point (called the sink node, base station, or control center) using either one or more hops [2]. WSNs are used in many industries, including aerospace, home, and military applications [3]. The sensor nodes are often placed in tough environments or remote locations, which makes various forms of attacks on their routing protocols [4]. Both internal and external attacks are possible [5]. To help WSNs operate securely, measures like encryption and identity checks have been suggested to protect against outside threats. However, these methods do not work well against attacks that happen inside the network itself [6]. This is because these systems need network nodes that are trustworthy and reliable, which is hard to guarantee [7]. Additionally, they require complex calculations and a lot of memory, which can lead to higher energy usage. As a result, researchers have proposed security solutions based on trust to defend against internal attacks in WSNs[8].

One technique for lowering energy usage in Wireless Sensor networks is clustering. It functions by assembling nodes into smaller clusters, with a leader known as the “Cluster Head” in each cluster. The cluster head manages the collection and processing of data. Each sensor node in the cluster sends its data to the cluster head. Clustering helps enhance the performance of the network, especially when there are many sensor nodes. It also makes the network more scalable by reducing the need for a central control system and allowing for local decision-making within each cluster. As long as each cluster remains active, the network can continue to function effectively. Figure 1 shows a typical structure of a Wireless Sensor Network.

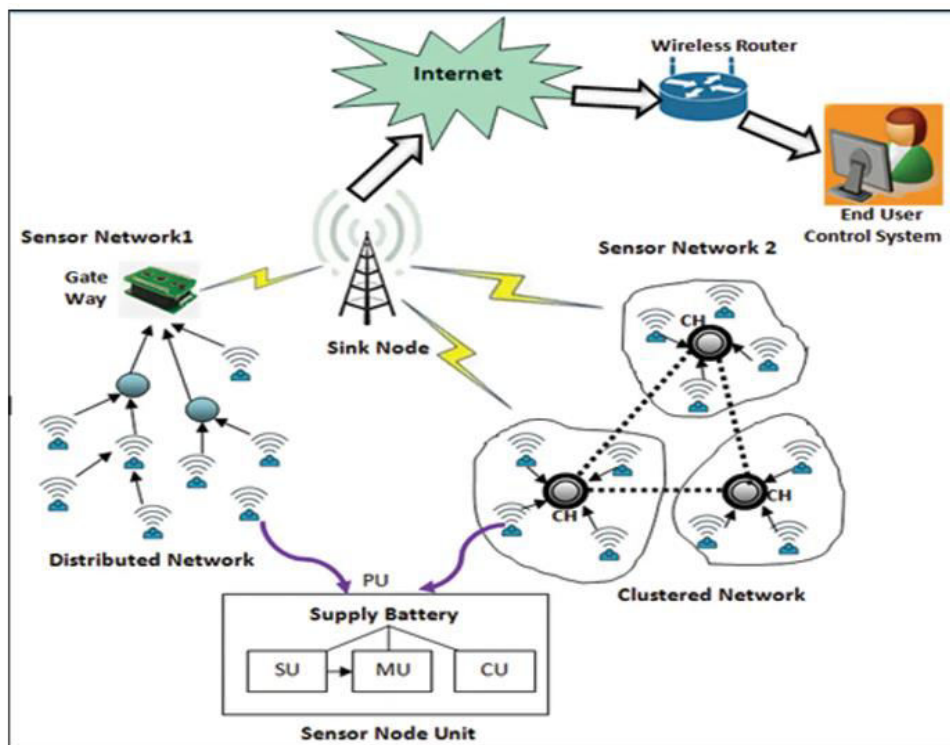


Figure 1: Architectural Framework of a Wireless Sensor Network (WSN)

The Cluster Head (CH) is a crucial issue in cluster-based routing protocols that are NP-hard to optimize. When choosing a CH, these challenges are addressed by employing metaheuristic approaches to find the best or nearly optimal answer. Cluster-based routing for WSN is now the focus of substantial study. To ensure consistent node energy consumption, the CH in a cluster was randomly selected. However, if the battery capacity is limited, the energy of the sensor battery will be exhausted. Nonetheless, the battery of the sensor will eventually run out of charge owing to its limited capacity. Energy Harvesting (EH) technology has recently been included in WSN to alleviate energy constraints [10].

These characteristics make it impossible to develop cluster-based protocols for EH-WSN directly [11]. However, because of CH selection, the achieved energy usage efficiency is neglected when building data transmission methods for cluster-based protocols. Swarm intelligence and nature-inspired architecture can maintain the energy consistency of sensor nodes, thereby increasing network longevity [12]. However, when the sensor nodes are left unattended or in hostile locations, their routing algorithms are vulnerable to a vast range of attacks. These attacks fall into two categories: internal and external attacks. Identity verification and cryptography-based security solutions are recommended to combat external WSN attacks. This enables them to function in a safe and secure WSN environment. However, these approaches were unsuccessful in internal network attacks [13]. Consequently, trust-based security solutions have been offered as a possible response to internal WSN attacks. In a trust-based security system, historical data are used to forecast the future behaviour of nodes. Trust management is an effective way to create trust links

between the sensor nodes. Traditional security systems have drawbacks such as greater energy consumption, the inability to combat multiple types of threats at once, and poor detection of malicious activity.

However, cluster head selection is random in each round and each sensor node get an equal chance of becoming a cluster head. As a result, even after several complete cluster head rounds, the probability of numerous low-energy nodes becoming cluster remains consistent. If these low-energy nodes are chosen at this time, the entire cluster will be rendered inoperable, endangering the security and life cycle of the network. Probability-based selection procedures produce some variations in the location and number of CHs. Consequently, the network's CHs spread unevenly. Some cluster heads were spread over the cluster, while others were at the boundary. Regardless of the conditions, it affects network performance. Fortunately, an increasing number of specialists have demonstrated that trust management is a viable technique for strengthening security [14]. WSN energy optimization aims to extend the operational period of the network by maximizing the use of available energy sources. To improve network efficiency and extend sensor node life, energy optimization methodologies have been used to solve the energy loss caused by transmission faults such as emission noise, collision and overheard communication, and idle listening. To select the appropriate CH and then provide an enhanced cluster solution [15], many researchers have started to explore various swarming-based algorithms, including Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Genetic Algorithms (GA), Differential Evolution (DE), Artificial Bee Colony Algorithm (ABC), Grey Wolf optimization (GWO), and Fire hawk optimizers (FHO) and other within the same framework.

Energy efficiency is considered the most serious challenge with WSNs because sensor nodes rapidly expend energy, rendering them inoperable. However, when searching for such an optimal cluster scheme in a WSN, postponing the heuristic algorithms used in existing cluster routing protocols has limitations because they typically produce results in a short time period, resulting in a region of the parameter space that is lower than the general ideal situation. This could indicate more energy usage and a shorter network lifetime. Furthermore, a cluster head arrangement that wastes a large amount of energy by providing data directly to the sink degrades the network. In this instance, the structure or protocol must consider efficient data transmission from the sink to the associated cluster head via the shortest method available. Experiments have shown that improper routing practices result in rapid battery losses. Techniques for developing and implementing energy-efficient routing algorithms are critical to increasing the lifespan of WSNs.

The structure of the remaining paper is as follows : Section 2 explain related study and objectives, section 3 describes the proposed method, section 4 shows the simulation results and performance evaluations and section 5 provides the conclusion and future work.

II. RELATED STUDY

This section reviews relevant studies on Energy-Aware Routing in Wireless Sensor Networks (WSN), highlighting the problems, solutions, results and limitations found in the literature.

Keerthiks et al. [16] introduced an energy-efficient dynamic routing system called Reinforcement-Learning Optimized Routing (RLER), aimed at reducing node power consumption and improving network performance. The RLFIS technique selects the best routing node based on parameters like algebraic connectivity (AC), neighbourhood overlap (NOVER), and bioactivity index (BI). Their method uses grid-tree-based clustering, where the root node forms a tree structure for inter-cluster communication, considering energy and transmission distance between nodes. The system also employs a hybrid BAT-row search algorithm (BCSA) to optimize node positions based on individual fitness values.

Abadi et al. [17] proposed an energy efficient routing and control system for WSNs based on reinforcement learning (RL). Their algorithm helps control energy consumption in the network by using machine learning to improve routing policies. Three strategies are used for energy management: Reducing distance to minimize energy use, A sleep-scheduling algorithm to conserve energy and A method that adjusts data transmission rates based on the node's power supply.

Liu et al. [18] introduced a DNN-HR approach to optimize WSN reinforcement learning in three ways. First, a two-level learning system is used, where reinforcement learning is performed by discrete nodes interacting with neighbouring nodes. Neural networks guide the learning process of these nodes. Second, sensor networks are divided

into two subspaces to create weight incremental propagation trees. Third, weight increments found by sensor nodes are reported to CHs, which then send the results to a base station for updates.

Wang et al. proposed a cooperative model called CRLM, which combines meta heuristic algorithms with reinforcement learning. This model solves network communication problems, such as Clustering and Routing [19], in a single phase and uses reinforcement learning to enhance the solution. The model also incorporates load balancing to prevent energy wastage by using new multi-hop protocols in densely populated areas.

Surrender et al. [20] explored a Deep Learning-based Grouping Model Approach (GMA) to optimize energy usage in WSNs. GMA improves energy efficiency by selecting and managing Cluster Heads using deep learning techniques like RNNs and LSTM, QoS. In this traffic level, and energy efficiency showed that GMA effectively links energy consumption with overall network performance.

Xue et al. introduced a Cross-Layer Harris Hawk Optimization (CL-HHO) algorithm [21] combined with K-medoids and an improved Artificial Bee Colony (K-IABC) for energy-efficient clustering. The goal of this approach is to reduce power consumption and transmission delay and enhance its performance compared to other existing methods.

Bekal et al. [22] developed a PSO based Enhanced LEACH protocol for hierarchical WSNs. This approach uses Particle Swarm Optimization to explore multiple routes, overcoming limitations of the traditional LEACH protocol, such as total energy usage and node survival. The PSO method helps optimize energy dissipation, improving the protocol's efficiency in real-time applications.

Liu et al. focused on reducing energy usage associated with frequent cluster formation by implementing an interval-based Cluster Head (CH) Re-election Mechanism and Dual-CH Clustering Routing approach. They also introduced a Hybrid Optimization algorithm [23] called GWOA_CH, which combines Grey Wolf Optimization (GWO) and Whale Optimization Algorithm (WOA) to improve CH election schemes.

Ramalingam et al. [24] proposed an adaptive fuzzy approach combined with the Adaptive Sailfish Optimizer (ASFO) for CH selection, based on the Enhanced Elephant Herd Optimization (EEHO) technique. The hybrid method, implemented in MATLAB, was compared with other techniques like PSO, HCCHE, GA, and IABC-C. Results showed that the Fuzzy-ASFO approach improved quality of service, energy consumption, packet loss ratio, and other network factors.

He et al developed a Low-Energy Clustering and Multi-hop Routing methods based on a Bio-inspired technique, called Hierarchical Chimp Optimization (HCO). The method [25] optimizes Multiple-hop Routing and Clustering simultaneously, and the results were compared with those of LEACH, TEEN, MPSO, PSO and IPSO-GWO systems. This approach helps reduce energy consumption and extend the lifespan of WSNs.

2.1. Objectives of the Study

- Design and implement the Energy-Aware and Trust based Hierarchical Routing Protocol (EATHRP) for the goals of energy efficiency and improvement of the life of Wireless Sensor Networks (WSN).
- Cluster formation is done by Standard Deviation with Manhattan-based Farthest First K-Means (SDM-FFKM) used to form stable and optimized clusters and balance load over the sensor nodes.
- Enhancing Cluster Head Selection (CHS) The proposed method Hybrid Puffer Fish Walrus Optimization (HPWO) algorithm was created to reduce the convergence act and find the energy-efficient CH to get rid of all power use.
- Deep Neural Networks Bi-directional Long Short Term Memory DNN-BiLSTM is used to generate the learned paths and enable adaptive route planning in wireless sensor networks (WSNs).
- Optimize Path Selection is done by using the Enhanced Coati Optimization Algorithm (ECOA) to determine the most efficient routing paths, resulting in both reduced end-to-end delay and increased reliability in data delivery.
- Protect networks from wrongful access, CDFPCC helps to provide additional protection against wrongful access and cyber threats to strengthen network security.

- The study focuses on Performance Comparison and Validation, specifically benchmarking the EATHRP protocol against popular protocols such as LEACH, FCM, DKNN, and DNN, to demonstrate that EATHRP has the best performance metrics among these protocols.
- High implementation network efficiency accounts for major performance metrics such as Energy efficacy, Throughput, Latency, Network lifetime and some other metrics are Loss ratio, Buffer capacity, End-to-End delay, etc., which are the most performing WSN architecture.

2.1. Problem Statement

Most of the work carried out to optimize energy usage and prolong the lifetime of WSNs has focused on the development of complex algorithms and protocols meant for solving challenges related to providing support for both energy efficiency, eco-friendly features, and reliable data transfer. To enhance the selection of CHs, optimize routing patterns, and constrain energy usage in SNs, researchers have investigated a variety of techniques, such as deep learning models, reinforcement-learning based routing protocols for WSN networks with mobile nodes autonomously selecting paths through an evolutionary algorithm (EA), genetic algorithms, or hybrid meta heuristic methods combining both EA and Levy mechanisms. Such approaches typically employ advanced techniques to minimize node energy consumption, evenly distribute the load, and reduce communication overhead, including neural network hybrid clustering models or bio-inspired optimization algorithms. However, existing systems struggle to balance energy efficiency with network stability and quality of service (QoS), optimal CH selection, and adaptation in dynamic networks. Another substantial barrier is the continued industry demand for massive, customizable solutions that successfully manage WSNs through their varied and generally unpredictable lives.

2.2. Motivation of the Work

In this research, the energy aware and trust based hierarchical routing protocol (EATHRP) for Wireless Sensor Networks is proposed, which employs a Deep Ensemble Neural Network to handle the issues of scalability, eco-friendly features, and dependable data transport. The vast number of sensor nodes that make up these networks has limited energy resources; thus, techniques that offer maximum utilization in terms of energy for extremely long lifetimes. The increasing demand for more reliable, scalable, and energy-efficient WSNs has prompted research into next-generation algorithms and protocols that can intelligently manage resources, optimize communication overheads, and ensure constant data flow. By addressing these challenges, this study aims to significantly extend and enhance WSN lifetime performance. When extended network sustainability and energy savings are critical, this will enable WSNs to operate more effectively in real-world settings.

III. PROPOSED METHODOLOGY

The proposed Energy Awareness Trust and Hierarchy Routing Protocol (EATHRP) incorporate two key methodologies for optimizing WSN performance: Energy-aware cluster head selection using the Hybrid Puffer Fish Walrus Optimization (HPWO) Algorithm and route planning with (DNN-BiLSTM) Deep Neural Networks with Bi-directional Long Short-Term Memory. These methods are explained in detail below:

3.1 Energy Aware model

Managing power in individual sensor nodes helps reduce energy use [36], but it is equally important to ensure that communication between nodes is energy-efficient. Since Wireless data transmission consumes a significant amount of energy, optimizing inter node communication can lead to greater overall energy savings. Additionally, integrating power management into the communication process allows energy-aware decisions to extend beyond a single sensor node to a group of nodes, improving the lifespan of entire network regions. To enable energy-efficient communication, it is crucial to identify and utilize different trade-offs between performance and energy consumption within the communication system.

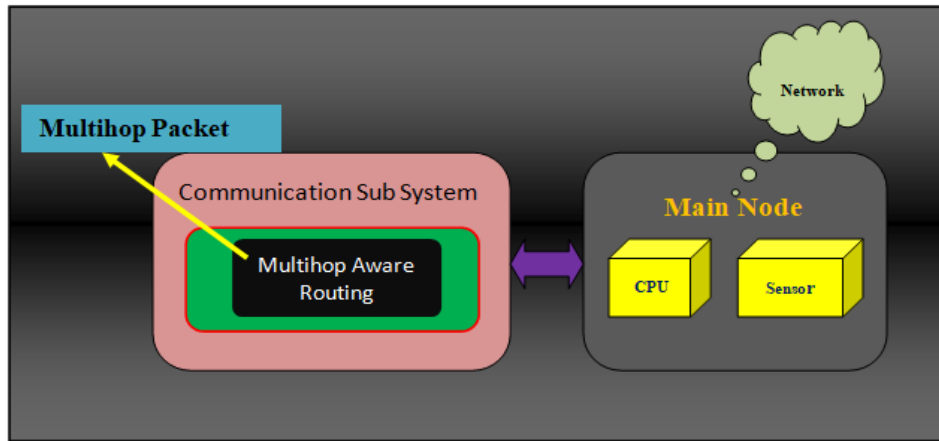


Figure 2 : Energy Aware Packet Forwarding Model

Energy-Aware Multihop Communication Equation

The following equation that reflects the energy-efficient communication in Wireless Sensor Networks (WSNs) as described energy-aware multihop communication that consider both transmission power and internode distance.

$$E_{Total} = \sum_{i=1}^N (E_{elec} \times \mathcal{K} + E_{amp} \times \mathcal{K} \times d_i^n) \quad (1)$$

Where E_{Total} = Total energy consumed for transmitting data across multiple hops.

N = No. of hops between the sender and the destination

E_{elec} = Energy required per bit for processing (E.g. Encoding, Modulation)

\mathcal{K} = No. of bits being transmitted

E_{amp} = Power amplifier energy consumed per bit

d_i = Distance between the i^{th} transmitting node and its next-hop neighbour

n = Path loss exponent (typically 2 for free-space, 4 for multipath fading)

The applications of WSN, such as smart cities and environmental monitoring, have attracted significant interest in recent years. A network of many tiny sensor nodes is called a WSN (which functions as both routers and data collectors) inside the network. Energy efficiency is critical, as these nodes operate from energy sources that are not infinite (non-renewable) and cannot be replaced or charged. Because routing decisions are directly linked to the energy consumption of each node, we need better control of conserving energy to improve the network lifetime. Despite the work currently being collectively explored by research scientists, numerous issues remain unexplored. Specifically, the predicament of how to send as much data as possible without it leading to catastrophic battery drain without endangering network performance is still being addressed.

Although the following research clearly demonstrates a strong technique in combination with energy-aware optimum clustering and secure routing in WSNs, it provides a comprehensive solution to the aforementioned issues. This technique is divided into the following sub-stages: Stage 1: formation of a network with nodes deployed. Stage 2: Formation of clusters using the SDM-FFKM algorithm; Stage 3: Selection hierarchical routing protocol of cluster heads (CH) using the HPWO algorithm. Stage 4: Extraction node attributes to assist in making routing decisions. Stage 5: Learning of path using DNN-BiLSTM by employing ECOA to identify the optimum among the options. Stage 6: Transmissions of secure data using CDFPCC Figure 3 shows the detailed block diagram of the proposed technique.

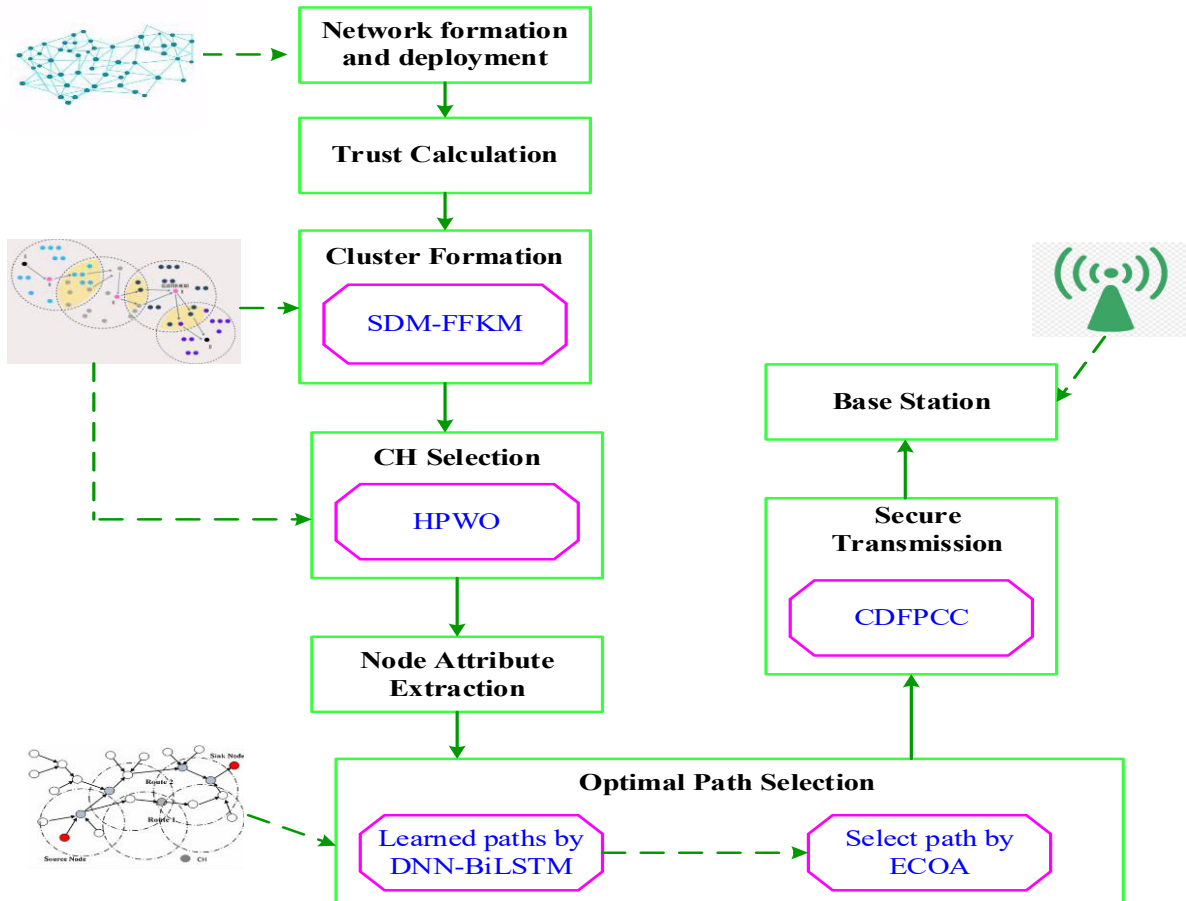


Figure 3: Structure for the Research methodology

3.2 Network Formation and Deployment

Initially, a network of sensor nodes was established. A value can then be identified by sending it to a base station via nodes. Measuring the node performance is crucial for network implementation. Equation (2) explains how this is expressed in an initialized node.

$$\bar{\mu}_y = \{\bar{\mu}_1, \bar{\mu}_2, \dots, \bar{\mu}_n\} \tag{2}$$

Where $\bar{\mu}_n$ indicates the n number of initialization nodes, and $\bar{\mu}_y$ denotes the initialized node set.

3.3 Trust calculation

During CH selection, nodes with higher trust values and energy levels are preferred. Trust aware routing protocols ensure that data is forwarded through high trust CHs to improve reliability. For this secure data transmission trust value of the node is important. Here trust value of node is calculated by using successful data transmissions. The following formula is used to calculate the trustworthiness of a node.

$$\text{Trust}(N_i) = \frac{\text{Total Successful Transmissions}}{\text{Total No.of Attempts}}$$

Where **Total Successful Transmissions** refers to the number of times node N_i has successfully transmitted data.

Total No.of Attempts refers to the total number of times node N_i has attempted to send data. This includes both successful and unsuccessful transmission attempts.

3.4 Cluster formation using Standard Deviation Manhattan-based Farthest First K-means Algorithm (SDM-FFKM)

The method of clustering is essential to the implementation of the energy-awareness and trust based hierarchical routing protocol for wireless sensor networks to optimize lifetime of the network and minimize the energy usage. In this research it is done based on the Standard Deviation and the Manhattan-based first K-means (SDM-FFKM) technique. Using a centralized approach, the sink node or Base Station (BS) performs the clustering process and assigns each SN to a certain cluster according to the location-related information.

This section presents an improved Farthest Initial K-means clustering algorithm, which is an extension of the k-means method [27]. We imported a developed rule in the Farthest Initial K-means clustering which is used to select the distinct distance measure for our method. In this method, first cluster centroids were chosen using the density-based clustering technique and the distance concept, which is more in accordance with the real distances of the dataset. Ultimately, clustering is completed, depending on the computations. The K-means method uses Manhattan distance as its distance measure. However, Manhattan distance can sometimes be problematic because differences in scale can result in different groupings. Daniel proposed a rule for choosing a suitable distance measure for the K-means algorithm. In this research a distance measure based on standard deviation was selected, using both the Manhattan technique and the Farthest First K-means algorithm. The rule for selecting the distance measure is as follows:

Rule: The Manhattan distance measure is chosen for the algorithm if $q \geq 8.4595$. The following is the formula for rule (q):

$$q = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^4 / \sigma^4 \quad (3)$$

Where $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$, $\sigma = \left(\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \right)^{\frac{1}{2}}$,

\bar{X} is the standard deviation, and σ is the sample mean. The No.of objects in the dataset were n.

The Manhattan distance is defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{im} - x_{jm}| \quad (4)$$

Where $i = (x_{i1}, x_{i2}, \dots, x_{im})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jm})$ are two m-dimensional objects.

The farthest first k-means algorithm had a default rule. If any data do not meet the previously specified requirements, K-means algorithm selects the Manhattan distance measure method. The improved approach refines the initial centroids, each of which may represent a group of related objects, and removes outliers when the distance measure is finished.

A. Getting the initial centroids

Instead of using random centroids, the K-means algorithm is better represented by choosing k predetermined centroids that are the furthest from one another. Outliers are prevalent in real-world datasets; however, and worse, they may lead to low-quality clusters. Because the small object area in most datasets separates the high-density object area, the low-density data item needs to be taken into account. To overcome this issue, the initial centroids in this study are the common farthest data objects, which are situated in high-density areas. This allows researchers to ascertain the density of the region to which the object belongs.

The Farthest-First procedure has two main steps : Selecting centroids and Assigning clusters. In the first step, the first cluster center is randomly chosen from the data. In the cluster assignment step, The second center is chosen to be the data point that is furthest distant from the first center. The next centers are chosen using the same method: they are selected to be far from the centers that were chosen earlier. Once the desired number of centroids (K) is selected, the process ends by assigning all remaining data points to the nearest centroid.

Unlike K-Means, the Farthest-First algorithm clusters data points in a single run. It does not calculate the average position of attributes to update centroids. Instead, it chooses centroids directly from the data points. This approach is a two-approximation for choosing the optimal centroids based on the distance between clusters, which is

known as the k-centers problem, even though the process begins with a random selection and only occurs once. To put it simply, this method selects centroids that are no more than twice as far apart as the ideal distance. Algorithm 1 describe the Farthest First Method.

Algorithm 1. Farthest first algorithm

1. Choose a data point at random to serve as the first centroid.
2. Find the data point farthest from the first centroid.
3. Choose the next data point that is the farthest from the two selected centroids.
4. Repeat the process for $i=3,4,-----,n$ until all centroids are selected.
5. Each data point belongs to the cluster that minimizes the maximum distance to the centroids, by using the following formula

$$\text{Min}(\max(\text{dist}(p_i, p_1)), \max(\text{dist}(p_i, p_2)), -----)$$

These clusters are referred to as SNs. A collection of nodes was formed.

3.5 Hierarchical Routing Protocol for Wireless Sensor Networks

Wireless Sensor Networks (WSNs) are consist of sensor nodes distributed across an area to monitoring and collecting data from the environment. Considering limited energy, bandwidth, and processing power, efficient routing techniques are important for augmenting the lifetime of a network and preserving communication reliability. Energy-efficient data transmission is done by the hierarchy in which nodes are organized into clusters with a number of cluster heads (CHs). Here is an introduction to a Hierarchical Routing Protocol (HRP) for WSNs. Using the HPWO algorithm for the selection of optimal cluster heads; it brings significant improvement in criteria like network lifetime, scalability, and reliability. The Hybrid Pufferfish Walrus optimization algorithm (HPWO) was used to select a CH based on the sensor node's lowest energy transmission. The suggested HPWO devices with the global search capability of POA and WOA algorithms refined local exploitation strengths, creating a synergistic approach tailored during the process of cluster head selection. First, by enlarging the search space, the POA algorithm converges with the WOA method. Furthermore, by fine-tuning the solution for the local optima, the WOA enhances performance. Nevertheless, combining local and global searches significantly outperformed the HPWO method. Owing to the problem of delayed convergence, the POA algorithm is also shown to be appropriate for local search exploitation as opposed to search space exploration.

3.6 CH selection using Hybrid Pufferfish Walrus Optimization Algorithm (HPWO)

The HPWO algorithm is derived from the pufferfish - Bio inspired based exploration method, where cluster heads are chosen in the case of distributed networks and developed by integrating pufferfish-inspired-based exploration with systematic walk-based optimization techniques. So, the HPWO is not trapped in a local optimum and explores widely and exploits narrowly simultaneously to arrive at the nodes with good energy coverage, connectivity, and optimal positions collectively, enhancing the performance. In addition to this, it also optimizes multiple objectives: residual energy, node density, and communication cost, which guarantees energy efficiency and load balancing, thus being important to enhance the lifetime of the wireless sensor network. In addition, being adaptive and convergence robust, it can be generalized to large dynamic networks, and its fast convergence allows near real-time decision-making in varying environments. The HPWO proved to be a practical method where much better nodes cover area and implemented scales applied to other applications.

The population-based method, known as the POA strategy, can produce efficient CHE through a population search. In this field, iterative processes have been used to solve these difficulties [28]. Therefore, the issue can be formally represented as a vector, where each element represents a decision variable and every member of the POA is a possible solution. All POA members combined make up the algorithm's population. The community at a large of these vectors is mathematically represented as a matrix in equation (5). At the beginning of the process, the primary location of each POA member is initialized using equation (6).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_f \\ \vdots \\ X_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_{1,1} & \dots & x_{1,d} & \dots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{t,1} & \dots & x_{t,d} & \dots & x_{t,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \dots & x_{N,d} & \dots & x_{N,m} \end{bmatrix}_{N \times m} \quad (5)$$

$$x_{t,d} = lb_d + r \cdot (ub_d - lb_d) \quad (6)$$

where $r \in [0,1]$ is a random integer, X_i is the i th POA member, $x_{i,d}$ is the d^{th} dimension in the CH selection space, and m is the outcome variable. The lower and upper bounds of these variables are represented as lb_d and ub_d , respectively. where N is the total No. of people in the population. Equation (7) shows the objective function vector.

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_t \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_t) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \quad (7)$$

The evaluated values of the objective function act as benchmarks to assess how well the solutions from each POA member are performing. The highest value indicates the optimal solution, while the least value represents the worst solution. By mimicking the natural behaviour of pufferfish and their predators, the POA method adjust the location of the population members within the problem-solving area. In nature, the pufferfish is the predator's first target. However, the pufferfish uses its defence mechanism to transform into a spiky ball, putting the predator at risk and allowing the pufferfish to escape. The POA method involves two stages for updating the population : (i) Exploration and (ii) Exploitation.

Phase 1: predictor Attack on Pufferfish (Phase of Exploration)

During the first phase of the POA, the population members were updated by simulating the predator's attack strategy on the pufferfish. Since pufferfish move slowly, they are easily caught by hungry predators. The global search algorithm significantly alters the positions of POA members, enhancing their ability to explore, by mimicking the predator's movement around the pufferfish. The following equation is used to determine the set of pufferfish associated with each population member.

$$CP_i = \{X_k: F_k < F_i \text{ and } k \neq i\}, \text{ where } i = 1, 2, \dots, N \text{ and } k \in \{1, 2, \dots, N\} \quad (8)$$

Here, X_k is the member of the overall population whose objective function value is higher than that of the i^{th} predator, P_k is the value of its objective functionality, and CP_i is the gathering of possible pufferfish habitats for the i^{th} predator. In the POA design, the selected pufferfish (SP) was assumed to represent the outcome of a predator randomly selecting one of the suggested pufferfish species from the CP collection. The updated location of each POA member in the CH search space of the predator's strategy to the pufferfish was obtained using equation (9). Equation (10) is then used to increase the new objective value at the new location corresponding to the previous location of the member.

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j} \cdot (SP_{i,j} - I_{i,j} \cdot x_{i,j}), \quad (9)$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} \leq F_i \\ X_i, & \text{else} \end{cases} \quad (10)$$

In this case, its j^{th} dimension is $x_{i,j}^{P1}$, its objective function value is F_i^{P1} , its i^{th} dimension is $SP_{i,j}$, and $r_{i,j}$ are arbitrary values drawn from the range $[0, 1]$. Using the first phase of the proposed POA, the revised position for the i^{th} predator was determined to be X_i^{P1} . For the i^{th} predator, the pufferfish chosen at random from the CP_i set is called SP_i (i.e., SP_i is a component of the CP_i set).

Phase 2: Pufferfish Defense System Against Predators (Exploitation Phase)

In the second phase of the POA, the position of population members was updated by mimicking the Pufferfish's defence mechanism against predator attacks, which involves inflating its stomach with water to make it into a spiky ball, which prevents the predator from attacking and causing it to flee rather than snatch an easy meal. If this new position improves the objective functions, the corresponding member is replaced, as shown in equation (9); if not, the member maintains its previous position. The update process for each POA member is dependent on increasing the objective function value, as indicated in equation (11).

$$x_{i,j}^{P2} = x_{i,j} + (1 - 2r_{i,j}) \cdot \frac{ub_j - lb_j}{t} \quad (11)$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} \leq F_i; \\ X_i, & \text{else} \end{cases} \quad (12)$$

For the i^{th} predator in the second phase, $x_{i,j}^{P2}$ are the updated predicted locations. j^{th} dimension and F_i^{P2} . $r_{i,j} \in [0,1]$ are random values and F_i^{P2} is the value of its objective function.

3.7 Walrus Optimization algorithm

Complex problem solving was accomplished using metaheuristic optimization techniques. The WOA draws inspiration from the way walruses survive in the Arctic Ocean. Its distinctive behaviors include evading, roaming, and battling predators [30]. The optimization paradigm for real-time problem solving is framed by this behavior. In contrast to conventional optimization methods, the WOA's distinct methodology reduces risk and permits the investigation of a wider range of possible solutions. Three phases were involved: feeding strategy, roaming, and escaping.

3.7.1 Method of feeding

Walruses have a wide diet but prefer benthic bivalve molluscs such as clams. They use vigorous motion and sensitive vibrissae (whiskers) to seek the seafloor for food. Similar to leading the hunt for the best answers, the strongest walrus, distinguished by its tusks, led the group to find food. This can be represented numerically using the following formulas (Eqs. 13 and 14):

$$P_{i,j}^{P1} = P_{i,j} + rand(M_{i,j} - I_{i,j}P_{i,j}) \quad i = 1,2, \dots, N \quad (13)$$

$$P_i = \begin{cases} P_{i,j}^{P1}, & P_i^{P1} < F_i \\ P_i, & \text{Else} \end{cases} \quad (14)$$

where rand represents a random integer in [0,1] and P_i^{P1} specifies the i^{th} freshly created walrus location. $I_{i,j}$ are randomly selected from {1, 2}, M is a single member, and j is the dimension.

3.7.2 Roaming

Exploration and exploitation for optimization were performed at this level. Based on the food supply, the walrus relocated to the next-best spot. The mathematical migration formula (Eq. 15) is as follows:

$$x_{i,j}^{P2} = \begin{cases} x_{i,j} + rand(x_{k,j} - Ix_{i,j}), & (F_k < F_i) \\ x_{i,j} + rand(x_{i,j} - x_{k,j}), & \text{ELSE} \end{cases} \quad (15)$$

Where $x_{i,j}^{P2s1}$ is the new location of Walrus.

3.7.3 Escaping

This phase is part of the optimization exploitation process. The walrus relocated to safer areas after calculating the escape and attack of the predators. The following mathematical model represents the escape behaviour.

$$x_{i,j}^{P3} = x_{i,j} + (ul_{local,j}^t - rand \cdot ll_{local,j}^t) \quad (16)$$

where $x_{i,j}^{p3}$ is the new location of walrus ul and ll is the upper and lower limits of the search space.

3.8 Optimal Path Learning Using DNN-BiLSTM Method

The best Hierarchical Routing protocol for Wireless Sensor Networks is determined by selecting the cluster head. To train the path for this, Deep Neural Networks with Bi-directional Long Short-Term Memory (DNN-BiLSTM) were used. DNNs effectively capture complex connection patterns by extracting high level features from network data. BiLSTM allows the model to consider both past and future data by processing inputs in both directions. DNN-BiLSTM ensures efficient pathfinding in dynamic networks by understanding temporal dependencies. Through nonlinear transformations, DNNs can learn complex patterns from the data. Activation functions like Sigmoid, tanh, and ReLU are commonly used to perform these transformations. The Sigmoid function maps values between 0 and 1, making it suitable for binary classification. ReLU, used deep networks, enables faster computations by returning 0 for negative inputs and keeping positive values unchanged. The tanh function is useful in multi-class classification, as it outputs values between -1 and +1. These activation functions help process data in a nonlinear way to extract complex features. DNNs can learn increasingly complex functions by stacking multiple nonlinear transformations. Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN), solves the vanishing gradient problem in RNNs by managing long-term dependencies. LSTM structures allow the model to decide whether to retain or forget information, as explained by Fu et al. (2002). Figure 5 shows the internal structure of the Path Learning DNN-BiLSTM.

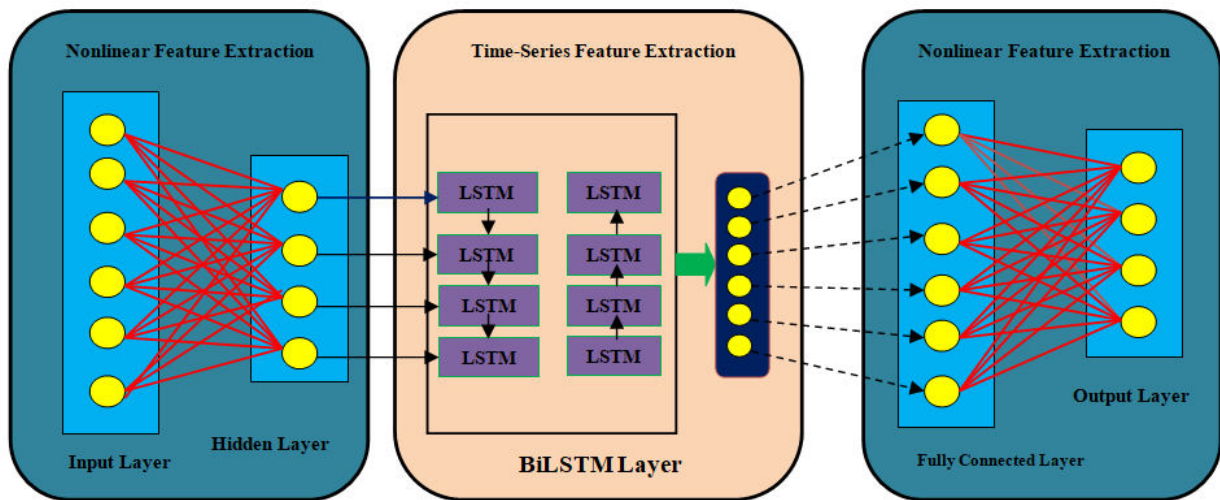


Figure 3. Internal structure of DNN-BiLSTM.

The following list of equations represents the computing units used in the one-time-step update of the LSTM process.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (17)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (18)$$

$$o_t = \sigma(W_{x0}x_t + W_{h0}h_{t-1} + b_0) \quad (19)$$

$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (20)$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (21)$$

$$h_t = o_t \tanh(c_t) \quad (22)$$

In the LSTM structure, the input gate is denoted by i_t , the current input by x_t , and the output of the previous hidden layer by h_{t-1} , which represents the hidden state from the last cell. The sigmoid activation function, symbolized by σ , maps any value between 0 and 1. The output gate is represented by o_t and the forget gate is denoted by f_t , which decides whether the information from the previous cell state, c_{t-1} , should be carried forward. The candidate memory cell is represented as \tilde{c}_t , the current cell state as c_t , and the previous cell state as c_{t-1} , reflecting the hidden state of the current cell. The hyperbolic tangent (\tanh) activation function transforms values between -1 and 1. To protect the current time step's information, the input data passes through the gate cells sequentially, as described above. However, previous studies show that the data features in LSTM models are often passed directly from the input to the output, leading to an accumulation of errors over time, which worsens as the time steps increase. To address this issue, this study uses a Bi-directional LSTM (Bi-LSTM) structure to prevent error buildup. In the Bi-LSTM, two LSTM layers work in parallel, processing the input data in both forward and backward directions, and their results are combined to make the final prediction. This bidirectional approach improves the memory of sequential data and strengthens the dependency between data points, which leads to more accurate detection. The feature extraction process of the Bi-LSTM can be represented by Equation (23)

$$h_t^{\rightarrow} = \tanh(W_{h_t^{\rightarrow}} x_t + W_{h^{\rightarrow} h_{t-1}} h_{t-1}^{\rightarrow} + b_{h^{\rightarrow}}) \quad (23)$$

$$h_t^{\leftarrow} = \tanh(W_{h_t^{\leftarrow}} x_t + W_{h^{\leftarrow} h_{t-1}} h_{t-1}^{\leftarrow} + b_{h^{\leftarrow}}) \quad (24)$$

$$h_t = h_t^{\rightarrow} + h_t^{\leftarrow} \quad (25)$$

Where h_t indicates the hidden state of the current cell, h^{\rightarrow} and h^{\leftarrow} indicate the output results of the two forward and reverse LSTMs, respectively, and h_{t-1} indicates the output state of the hidden layer of the previous cell. Therefore, the learned pathways were updated. The fitness function has the shortest distance and lowest energy consumption, and the best path is chosen from the options using the ECOA.

3.9 Optimal Path selection using Enhanced Coati Optimization Algorithm (ECOA)

The Enhanced Coati Optimization Algorithm (ECOA) is a well-structured optimization method that identifies the best paths from a set of possibilities. It strikes a good balance between exploring new solutions and exploiting known ones by improving its exploration strategies and development techniques. The algorithm uses chaos mapping to enhance randomness and global search ability. It also features a dynamic learning strategy with random elements to prevent the algorithm from getting stuck in local optima, improving its overall robustness. To maintain population diversity and enhance accuracy, ECOA includes an Adaptive Levy Flight method, which helps avoid premature convergence. Additionally, a different evolution technique based on cosine disturbance is applied to remove poorly performing individuals, thus improving the quality of the population. The performance of ECOA is assessed in terms of four factors: Population diversity, Exploration-Exploitation balance, Convergence, and The effect of different variations.

1. Adaptive levy Flight strategy

The Adaptive Levy Flight technique solves problems in the COA algorithm, such as low population diversity and too much randomness, which can cause early convergence and lower accuracy. This distribution has a concentrated center and gradually decreases at the tail. The Adaptive Levy Flight technique alternates between short and long steps. Longer steps help expand the search range and maintain population diversity, while shorter steps improve the optimization and convergence accuracy of the algorithm. The combination of different step lengths allows the algorithm to perform well in various problem types by balancing exploration and exploitation. The Adaptive Levy Flight method is applied during the exploration phase of the search, based on its search characteristics. Coatis compete for food, and those with more food are more likely to survive. The specific formula for this is given in Eq(26).

$$x_{i,j}^{new,P1} = x_{best} + RL \times (x_{best} - I \times x_{i,j}) \quad (26)$$

$$RL = 0.5 \times Levy(D) \quad (27)$$

Here, m is the dimensional space, $Levy(m)$ is the Levy flight distribution function. RL is the Levy flight with specified weights, and X_{best} is the best solution discovered up to the t^{th} iteration. Equation (28) illustrates the calculation process.

$$Levy(m) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\eta}}} \quad (28)$$

The constants s and η are fixed at 0.01 and 1.5, respectively. u and v are random numbers in the interval $[0,1]$. σ is obtained using Eq. (29).

$$\sigma = \left(\frac{\Gamma(1 + \eta) \times \sin\left(\frac{\pi\eta}{2}\right)}{\Gamma\left(\frac{1 + \eta}{2}\right) \times \eta \times 2\left(\frac{\eta - 1}{2}\right)} \right)^{\frac{1}{\eta}} \quad (29)$$

Γ represents the gamma function. The value of η is 1.5.

2. The Cosine Perturbation differential evolution

In optimization techniques, the cosine perturbation factor aids in striking a balance between exploration and exploitation. The algorithm can scan the global space more quickly and is less likely to become trapped in local optima when cosine functions are used. The cosine perturbation's randomness enhances the algorithm's robustness, preserves population diversity, and permits adjustment of the convergence pace, making it appropriate for a range of scenarios and search stages. The differential evolution method, which makes use of differential operations, is effective in solving high-dimensional, nonlinear, and non-smooth problems and has powerful global search capabilities. Its parallel architecture increases search efficiency by enabling the simultaneous evaluation of several possible solutions. This method does not require gradient information to handle many local optima and complex problem landscapes. The cosine perturbation factor increases flexibility and resilience, while the differential technique provides strong global search power and efficient parallel processing. Combining global and local optimization improves the algorithm's adaptability and performance, especially when dealing with complex problem landscapes and many local optima. A Cosine Perturbation Differential Evolution technique is presented to overcome the limitations of COA, including its tendency to become trapped in local optima and its delayed convergence in subsequent iterations. This tactic uses differential evolution and cosine perturbation in the target selection and assault phases of COA, where $F_{pi} > F_i$. Equation (30) displays the improved Coati's updated position.

$$x_{i,j}^{new,S2} = x_{i,j} + rCosine\ factor \times (x_{random1} - x_{random2}) \quad (30)$$

In this iteration, $x_{random1}$ and $x_{random2}$ are Coatis individuals are randomly selected, and , $rCosine\ factor$ is a randomly generated cosine-like value that can be calculated. This is shown in equation (31)

$$rcosine\ factor = \cos(2 \times rand) + 1 \quad (31)$$

Rand is a random number ranging from 0 to 1.

3.10. Enhanced Coati Optimization Algorithm

In 2023, a novel metaheuristic method called the Coati Optimization method (COA) was put forth [31]. It draws inspiration from two of the coatis' natural behavioural strategies for confronting and evading predators, as well as

cooperative tactics for attacking iguanas. Each coati functions as a distinct search individual in a p-dimensional space. The ability of the coati to hunt and flee predators is a distinct process. The globally ideal site, or the best candidate solution, will be chosen after the position of the coati is constantly modified based on the iguana's position and migrating locations. A mathematical model of the COA is briefly introduced.

Initialization process

By Equation (28), where the highest limit for each value is $X_{max} = (x_1^{max}, \dots, x_p^{max})$ and the minimum boundary is $X_{min} = (x_1^{min}, \dots, x_p^{min})$, is used by the COA to initialize m random individuals, $x_1, \dots, x_i, \dots, x_m$. Subsequently, the objective function used to assess the initialized random individuals.

$$x_{i,j} = x_j^{min} + rand. (x_j^{max} - x_j^{min}), i = 1, 2, \dots, m, j = 1, 2, \dots, p \quad (32)$$

1. Strategies for Hunting and Attacking (Exploration Phase)

Coatis together attack iguanas. Using this method, coatis were split into two groups. A party approaches and frightens the iguana by climbing a tree, whereas another group waits silently on the ground. When the iguana falls, raccoons attack it immediately and take it. An iguana is attacked by the coatis, as shown in Figure 4.

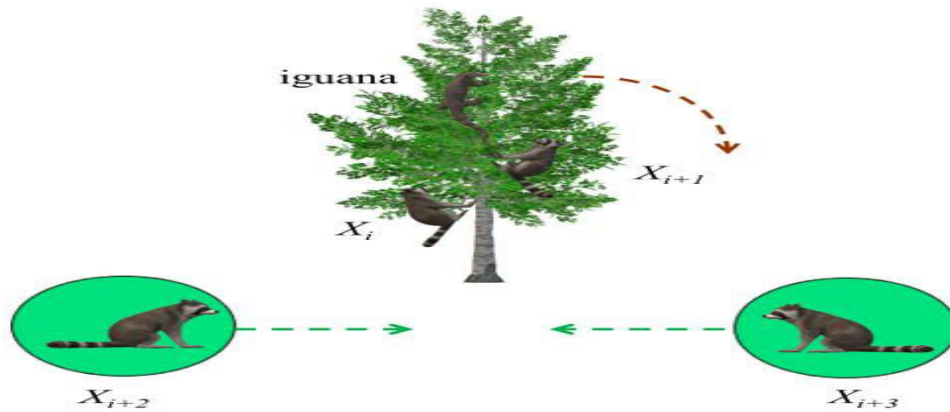


Figure 4 : Coatis attacking iguana

This method is based on two assumptions. First, it is believed that iguana is in the best possible position. Second, half of the species waited on the ground, whereas the other half climbed trees. Coatis climbing trees can be simulated using equation (33).

$$x_{i,j}^{new} = x_{i,j} + b. (Igu_j - I. x_{i,j}), i = 1, 2, \dots, \lfloor \frac{m}{2} \rfloor \quad (33)$$

The iguana's position is chosen at random when it is scared to land; however, the other half of the coatis waiting on the ground move according to iguana's unique positioning. By equations (34) and (35) show this pattern. Figure 8 shows the arrangement of iguana's newest location.

$$Igu_j = x_j^{min} + b. (x_j^{max} - x_j^{min}), j = 1, 2, \dots, p \quad (34)$$

$$x_{i,j}^{new} = \begin{cases} x_{i,j} + b. (Igu_j - I. x_{i,j}), & F_{Igu} < F_i \\ x_{i,j} + b. (x_{i,j} - Igu_j), & F_{Igu} > F_i \end{cases}, i = \lfloor \frac{N}{2} \rfloor + 1, \lfloor \frac{N}{2} \rfloor + 2, \dots, j = 1, 2, \dots, m \quad (35)$$

Where $x_{i,j}$ is the j-th dimension of the i-th coati and b is a random value between [0,1]. Igu is the location where iguana is randomly assigned. Any random number between 1 and 2 may be considered.

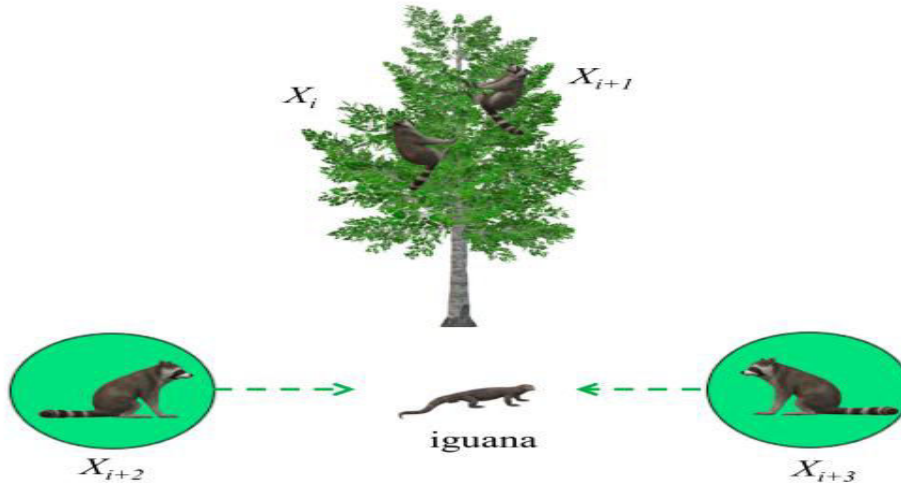


Figure 5 : Location updating of Iguana

2. Stage of Running Away from Predators (Exploitation Phase)

The coati updates its position during the development phase using the tactic it uses to confront and flee predators. As seen in Figure 9, a coati swiftly flees from a predator's grasp and moves into a position that is comparatively safe, approaching the ideal posture. This tactic shows the capability of the COA algorithm development.

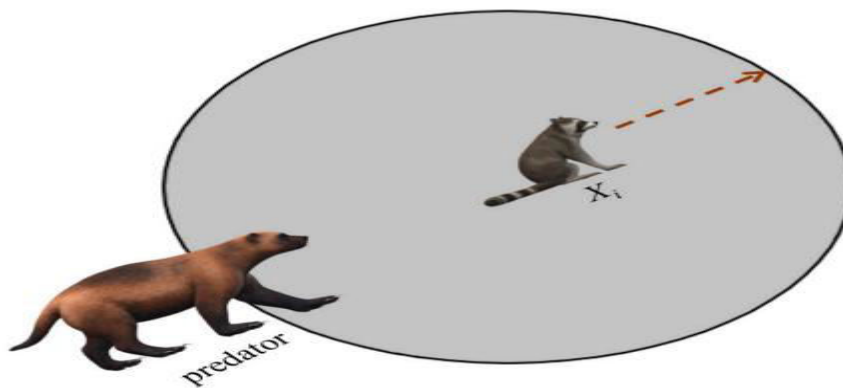


Figure 6 : A schematic representation of a coati's behaviour while it flees from a predator

Equations (36) and (37) were used to simulate the approach during the development phase.

$$low_j = \frac{x_j^{min}}{t}, up_j = \frac{x_j^{max}}{t}, t = 1, 2, \dots, T \quad (36)$$

$$x_{i,j}^{new} = x_{i,j} + (1 - 2b) \cdot (low_j + b \cdot (up_j - low_j)), i = 1, 2, \dots, m; j = 1, 2, \dots, p \quad (37)$$

The variables low_j and up_j represent the j^{th} dimension's local lower and upper bounds, respectively, where b is a stochastic number between 0 and 1. Accept the modified adaptation value for the coati if it is better than the original one; otherwise, reject it. The updating procedure is shown in equation (38).

$$X_i = \begin{cases} X_i^{new}, & F_i^{new} < F_i \\ X_i, & \text{else} \end{cases} \quad (38)$$

3.11 Secure transmission for Cumulative Distributed Four Point Curve Cryptography (CDFPCC)

The Cumulative Distributed Four Point Cryptographic Curve (CDFPCC) is a security framework designed to improve encryption and authentication in Wireless Sensor Networks (WSNs) while using less power. WSNs have limited resources, so cryptographic operations must be efficient and lightweight. CDFPCC achieves this by combining a single elliptic curve, distributed cryptographic methods, and four point calculations to enhance security without increasing computational load. CDFPCC provides a secure and efficient cryptographic model for encrypting data, authenticating nodes, and ensuring reliable key management. The following equations represents how it suitable to sending secure data from sensor nodes to the base station.

3.11.a. Elliptic Curve Representation

Each sensor node in a WSN has a unique cryptographic key derived from an elliptic curve equation. This curve is used to generate public and private keys for secure communication. The elliptic curve equation over a finite field \mathbb{F}_p is :

$$Y^2 = x^3 + ax + b \pmod{p} \quad (39)$$

Where p = A prime number defining the finite field

(a,b) = Security defined parameters chosen to maintain strong cryptographic properties.

(x,y) = Represents the public key of a sensor node, which is shared for encryption and authentication.

3.11.b. Distributed Key Management

Key management is crucial to prevent single node failures and unauthorized access. Instead of storing the full key on a single, CDFPCC distributes the secret key across multiple nodes using threshold cryptography.

$$S = \sum_{i=1}^t S_i \pmod{p} \quad (40)$$

Where S = The complete private key required for decryption and secure data exchange

S_i = A portion of the key stored on different sensor nodes

t = The minimum number of nodes required to reconstruct the key

3.11.c. Four-Point Scalar Multiplication for Secure Data Transmission

$$S = \mathcal{K}_1 \mathcal{P}_1 + \mathcal{K}_2 \mathcal{P}_2 + \mathcal{K}_3 \mathcal{P}_3 + \mathcal{K}_4 \mathcal{P}_4 \quad (41)$$

Where $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4$: Represents Private keys of different sensor nodes

$\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4$: Represents Public keys of sensor nodes participating in secure communication

Overall, the Cumulative Distributed Four-Point Cryptographic Curve (CDFPCC) enhances security and efficiency and also establish secure communication between sensor nodes and the base station.

Pseudo code: Working process of proposed method

Input: Network node, Initial energy, Trust parameters

Output: Secure data transmission

Stage 1: Network formation and deployment

Node initialization

Stage 2: Trust calculation

Calculate trust factors

If Total No.of Successful Transactions More

Mark as trusted

Else

Mark as untrusted

Stage 3: Cluster formation

Apply Standard Deviation and Manhattan-based Farthest First K-Means (SDM-FFKM) for clustering

Group node into clusters

Stage 4: Cluster head selection

Initialize Hybrid Pufferfish Walrus Optimization Algorithm (HPWO)

For each walrus in population

Compute fitness based on energy and distance.

$$R_{opt} = \arg \max \sum_{i=1}^N [\alpha_1 \cdot RES(i) + \alpha_2 \cdot \left(\frac{1}{D(i)}\right) + \alpha_3 \cdot S(i)]$$

Optimizing overall network lifetime and performance

Update position using Pufferfish movement

End for

Select optimal cluster head

Stage 5: Optimal path selection

Train Deep Neural Networks with Bidirectional Long Short-Term Memory

(DNN-BiLSTM) for path learning

Generate **routing probability** for available path

The **routing probability** for a path $p(i, j)$ from node i is calculated as:

$$p(i, j) = \frac{\beta_1 RE(i) + \beta_2 \left(\frac{1}{D(i)}\right) + \beta_3 S(i)}{\sum_{i=1}^N [\beta_1 RE(i) + \beta_2 \left(\frac{1}{D(i)}\right) + \beta_3 S(i)]}$$

The best path is chosen from the options using ECOA

$$F(p_1) = [\alpha_1 \cdot RES(i) + \alpha_2 \cdot \left(\frac{1}{D(i)}\right) + \alpha_3 \cdot S(i)]$$

Stage 6: Secure transmission

Secure transmission using Cumulative Distributed Four Point Curve Cryptography (CDFPCC)

Encrypt data packets and send through the selected path

Decrypt data to Base station.

Pseudo code 1 illustrates the workflow of the proposed method

This method is divided into smaller steps: network configuration with deployed nodes. The first step is to create a network of several sensor nodes. In this methodology cluster formations are done by using SDM-FFKM method. SNs are the common names for these clusters. A group of nodes was created. After that based on the lowest

transmission energy of the sensor nodes, a cluster head was chosen using the hybrid pufferfish walrus optimization algorithm (HPWO). The node properties from the extraction are used to help with the routing decisions. To determine learned paths DNN-BiLSTM is used and it can ensure path efficiency in dynamic networks. Consequently, the learned routes are updated. The shortest distance and lowest energy consumption serve as the fitness function, and the ECOA is used to select the optimal path from the available possibilities. secure data transmission is done by using the CDFPCC method.

IV. RESULT AND DISCUSSION

Several studies have been carried out to assess the effectiveness of EATHRP. Python was chosen as the platform for implementing the algorithm. The simulation parameters are provided in Table 1. The average result from 100 independent trials, each using a different topology of uniformly distributed sensor nodes, is higher than the average of each individual simulation result.

Value	Parameters
(0,0) ~ (100,100)	Network Field
100 ~ 500	Node
40 m	Cluster Radius γ
15 m	Sensing Radius γ
(20, 200)	Sink Position
3 J	Initial Energy
256 bytes	Data Packet Size
10 bytes	Broadcast Packet Size
0.01 J	E Threshold
50 nJ/bit	E Elec
10 nJ/bit/m ²	E fs
0.0013 pJ/bit/m ⁴	E amp
5 nJ/bit/signal	EDA
80 m	Threshold Distance $d\theta$
5	Data Cycles per Round

Table 1 : Simulation parameters

4.1 Performance analysis

In performance evaluation, we compared well-known metrics: delay, loss ratio, delivery ratio, throughput, energy consumption, and latency, which are widely used in routing protocols. While we do include energy awareness and are based on trust-based hierarchical routing in our proposed model, the overall performance enhancement feature is indirectly visible in the graphs. Although not individually plotted, for example, improved energy use and latency are some of the benefits of our energy-aware design and trust-based decision-making. Additionally, it makes it easier to compare our model's energy and safety-focused features with existing performance metrics in a way that is fair and accurate. These features naturally lead to better network performance.

In this paper, we compare the proposed method with the current approaches, DNN, Deep Kronecker NN, Fuzzy C-means, and LEACH, in terms of Delay, Buffer occupancy, Delivery Ratio, End-End Delay, Loss Ratio, Energy Efficiency, Latency, Throughput, Energy Consumption, and Network Lifespan.

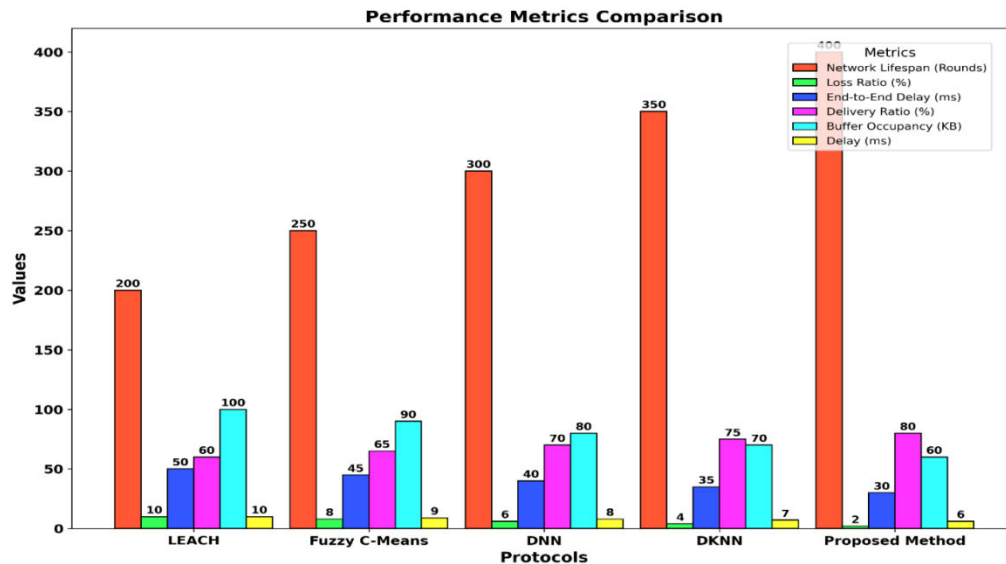


Figure 7 : Performance metrics comparison

In the above graph, the performance of the proposed method was compared with the current approach in terms of End-to-End Delay, Latency, Delivery Ratio, Loss Ratio, Buffer occupancy and Latency. We looked at how the No. of nodes affected these metrics. The No. of nodes impacts performance, as both the proposed method and the current techniques showed higher latency when more nodes were added. In terms of Delay, LEACH performed worse than the proposed method and the current approaches. The recommended strategy had a higher delivery ratio. Since this strategy includes trust measurements, it performs better overall.

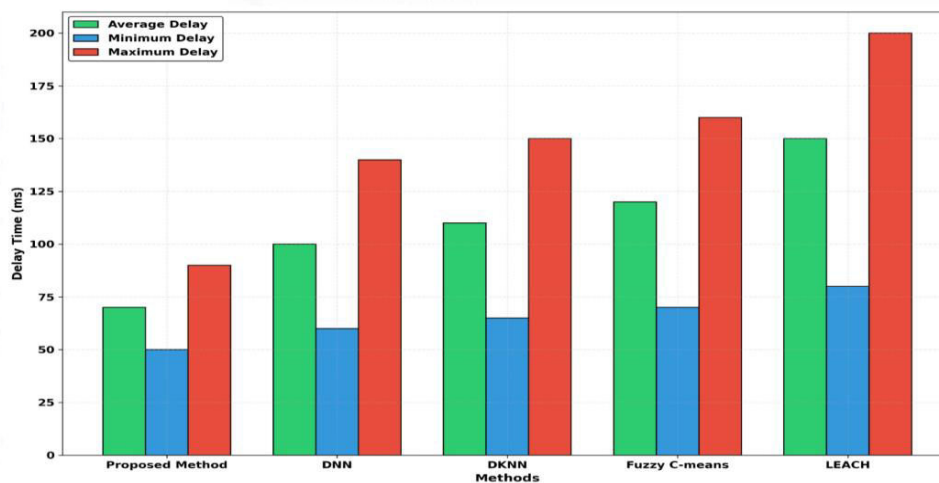


Figure 8: Comparison of delay Metrics

The proposed method has the lowest delays in all cases, with an average delay of 70 ms, a minimum of 50 ms, and a maximum of 90 ms. In comparison, DNN has longer delays, with an average of 100 ms, a minimum of 60 ms, and a maximum of 140 ms. DKNN and Fuzzy C-means have delays of 110 ms, 120 ms, 150 ms, and 160 ms, respectively. LEACH shows the highest delays, with an average of 150 ms, a minimum of 80 ms, and a maximum of 200 ms. This clearly shows that as we move from the Proposed method EATHRP to LEACH, the delay performance worsens. The EATHRP protocol consistently provides low delay times, showing its strong performance and reliability for applications that need timely data transmission in a network, resulting in an overall improvement in efficiency.

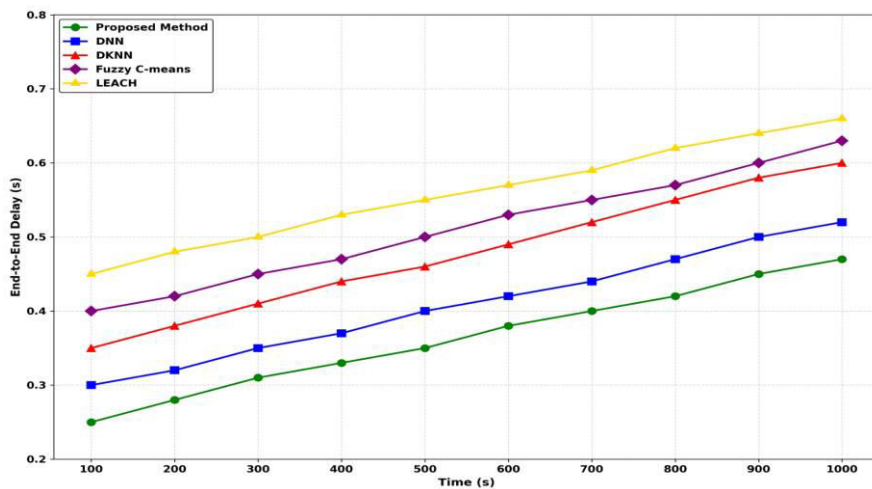


Figure 9: Comparison of End-to-End Delay Performance

The comparison of end-to-end delay shows that the Proposed EATHRP method performs the best in terms of delay, compared to DNN, DKNN, Fuzzy C-means, and LEACH. EATHRP has the lowest delay, while LEACH has the highest delay at all times. This clearly shows that EATHRP reduces transmission latency more effectively than the other methods. At 100 seconds, the proposed method achieves the best result with a delay of 0.15 seconds, which is much lower than DNN (0.25s), DKNN (0.30s), Fuzzy C-means (0.35s), and LEACH (0.40s). At 200 seconds, delays for all protocols increase, but EATHRP still performs the best with a delay of 0.20 seconds. In comparison, DNN, DKNN, Fuzzy C-means, and LEACH show delays of 0.30s, 0.35s, 0.40s, and 0.45s respectively. This trend continues through 300 seconds, 400 seconds, 500 seconds, 600 seconds, 700 seconds, 800 seconds, 900 seconds, and 1000 seconds.

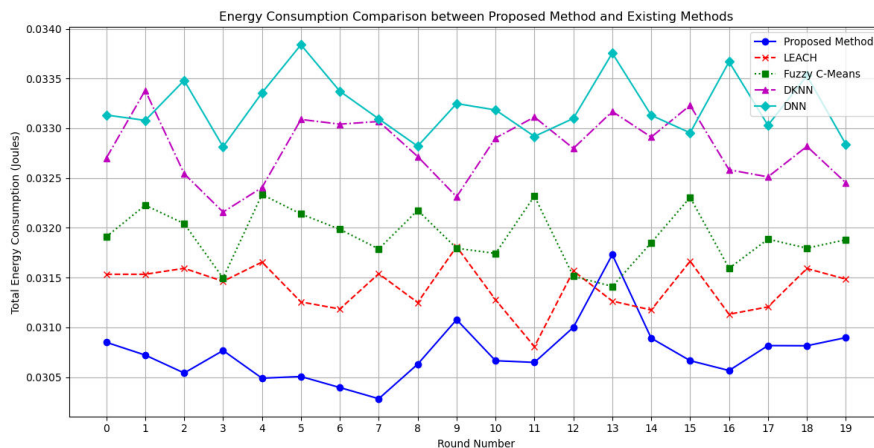


Figure 10: Comparison of Energy consumption

The energy consumption comparison shows the energy used by all five protocols during the first 15 rounds, with the total energy per round compared between the proposed protocol and existing protocols. As shown in Figure 13, the proposed method uses between 0.0303 and 0.0314 Joules of energy, which is much lower than the energy used by LEACH, Fuzzy C-means, and DKNN. The energy consumption increases in that order. For example, in round 0, the proposed methods uses 0.0309 J, while LEACH uses 0.0314 J, Fuzzy C-means uses 0.0320 J, and DKNN uses 0.0329 J, with LEACH and DKNN showing higher energy use. This pattern continues for rounds 1 to 14, showing that the proposed technique is more energy efficient. This makes the proposed method suitable for increasing network lifespan and reducing operational costs, especially in resource-limited conditions where lower energy consumption leads to lower costs.

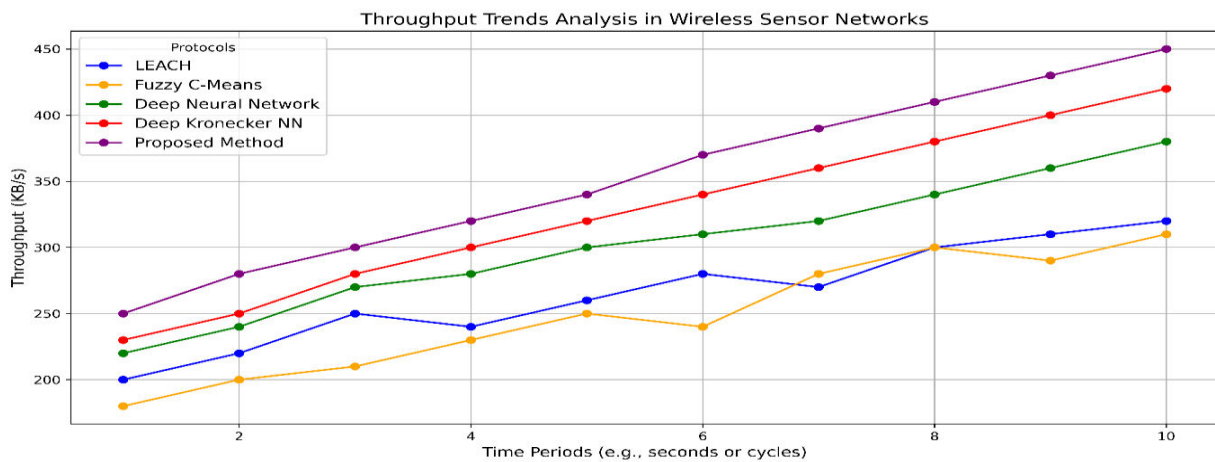


Figure 11 : Comparison of throughput

Figure 14 illustrates that the throughput performance of the EATHRP protocol is compared to existing methods, the proposed method consistently achieves higher throughput than the currently used techniques, highlighting its efficiency in managing data transmission. Specifically, when compared to DNN, DKNN, Fuzzy C-means and LEACH, the proposed method shows a significant improvement in throughput, achieving 80.0% higher throughput than DNN, 75.0% higher than DKNN, 85.0% higher than Fuzzy C-means, and 78.5% higher than LEACH. Furthermore, the proposed method provides a remarkable 90.2% improvement in throughput, indicating its capability to better utilize network resources and optimizing performance of the network.

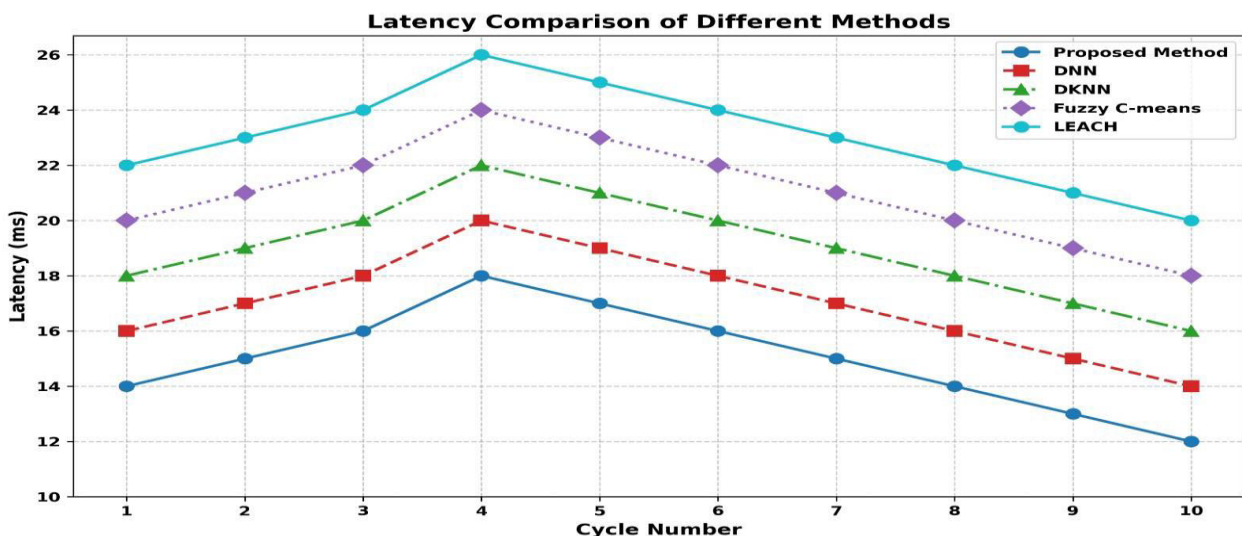


Figure 12 : Comparison of Latency

Figure 12 shows the latency, in milliseconds, for different methods over 10 iterations, providing an overview of how much delay each algorithm causes. The proposed EATHRP method starts with a latency of 14 ms in Cycle 1, while DNN, DKNN, Fuzzy C-means and LEACH have higher latencies of 16ms, 18ms, 20ms, and 22 ms respectively. As Cycles progress, the latency of EATHRP gradually increases, reaching a maximum of 24 ms in Cycle 4. In comparison, DNN and DKNN also show a similar increase, with maximum latencies of 22 ms and 24 ms, while Fuzzy C-means and LEACH peak at 26 ms and 28 ms. After Cycle 4, all methods show decrease in latency, suggesting that the system is optimizing or stabilizing. The lowest latency observed across all methods occurs between Cycle 4 to Cycle 10, where EATHRP achieves 12 ms, followed by DNN and DKNN at 10 ms and 12 ms, and Fuzzy C-means and LEACH with 14 ms and 16 ms, respectively. The decrease in latency from Cycle 4 to Cycle 10 indicates better performance and route optimization, with consistently delivering the lowest latency in the later Cycles.

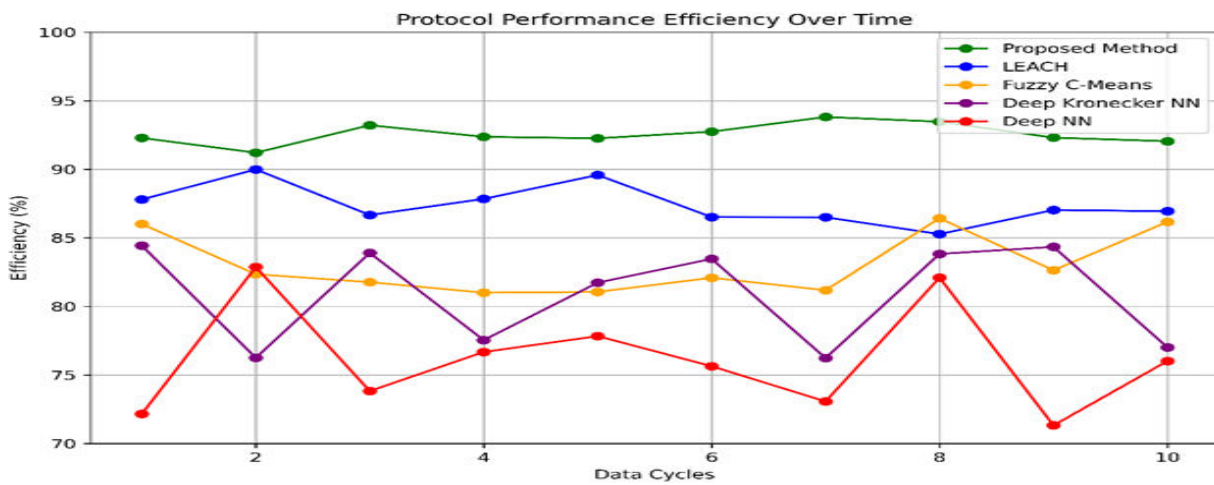


Figure 13 : Comparison of energy efficiency

In the above two graphs, we compare energy efficiency of proposed method and other existing protocols. It is clearly observed that the proposed method EATHRP shows much higher energy efficiency of 95% compared to other methods like LEACH, Fuzzy C-means, DKNN and DNN. This demonstrates that the proposed method is more effective in using energy, making it a better choice for energy conscious applications.

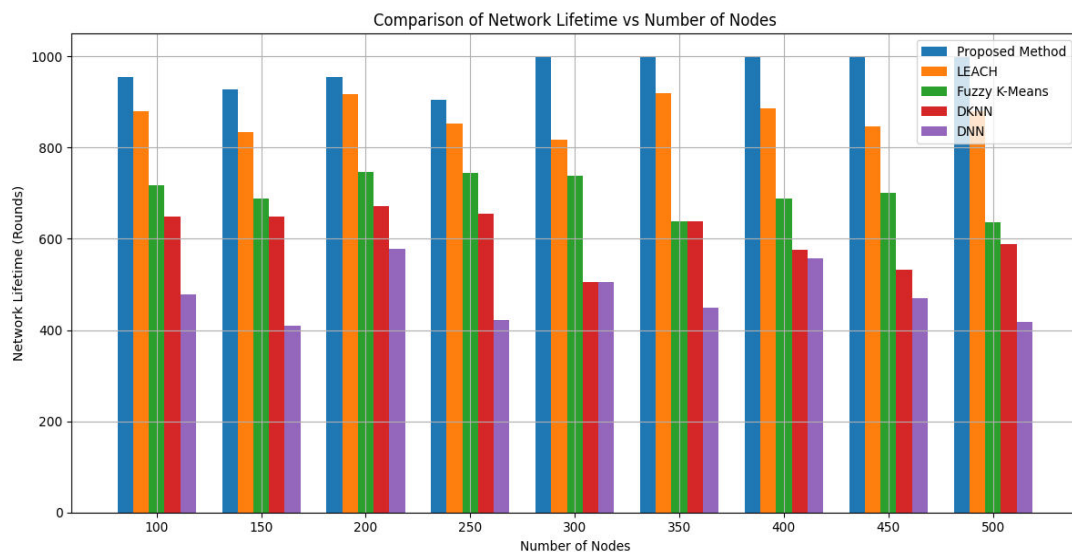


Figure 14 : Comparison of Network Life Time

Figure 14 shows that the network lifetime for the proposed method. The proposed method clearly increase the network life compared to the other methods, even with fewer nodes. It outperforms LEACH, Fuzzy C-means, DNN and DKNN in terms of network lifetime, achieving 80.0%, 77.5%, 80.0% and 76.0% improvements respectively.

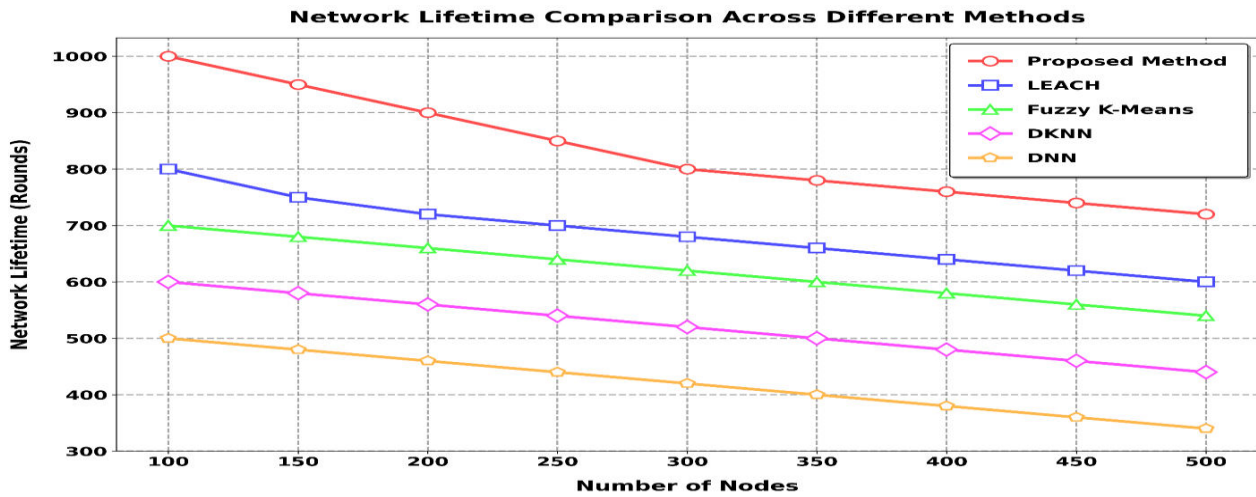


Figure 15 : Comparison Life time of the Network with different methods

Figure 15 shows the network lifetime (in rounds) for different methods. The proposed method reaches 1000 rounds of lifetime with a low node density (100 nodes), which is better than LEACH (800 rounds), Fuzzy C-means (700 rounds), DKNN (600 rounds), and DNN (500 rounds). As the network becomes denser, the lifetime of all methods decreases due to higher energy consumption and communication overhead. For example, when there are 1000 nodes, the proposed method's lifetime drops to 950 rounds, while LEACH and the other methods reach 750, 680, 580 and 480 rounds respectively. This downward trend continues as node density increases. With 200 nodes, the proposed method achieves 900 rounds, and with 500 nodes, it reaches 720 rounds, while LEACH, Fuzzy C-means, DKNN and DNN achieve 600, 540, 440 and 340 rounds respectively. Overall, the proposed method performs better than other techniques, showing it is more Energy efficient and scalable as the network size grows.

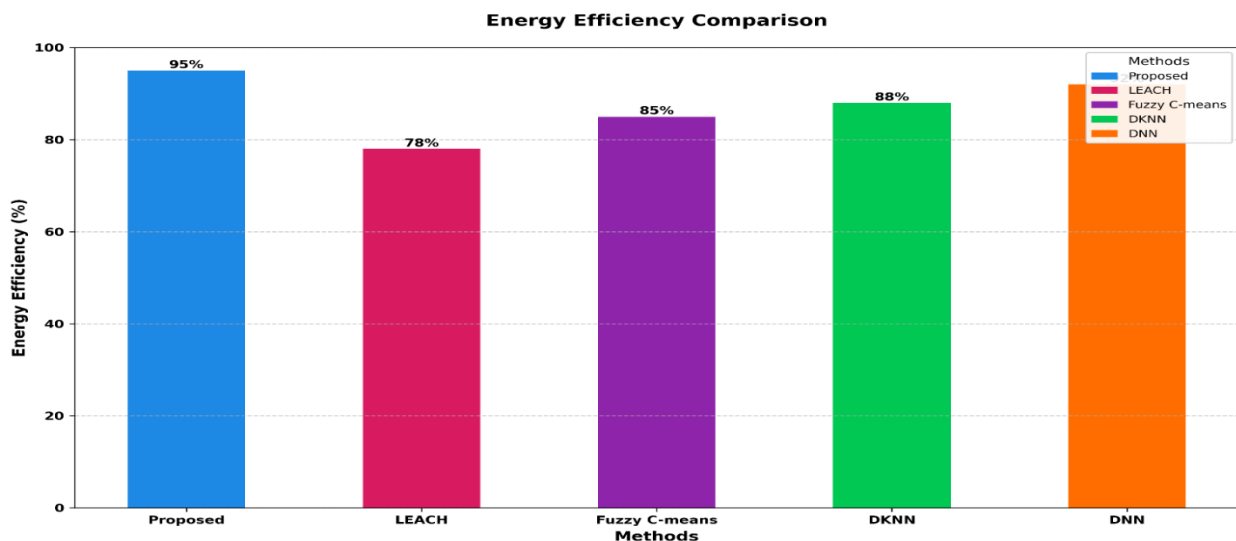


Figure 16 (a). Performance of Energy efficiency with different methods

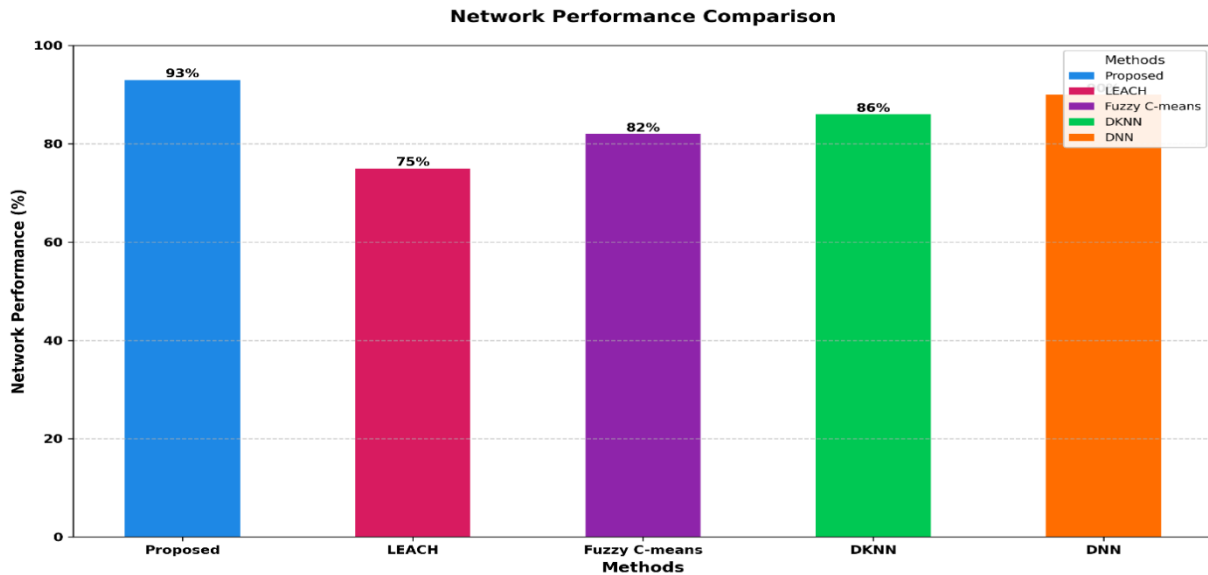


Figure 16 (b) : Network Performance Comparison with different methods

Figure 16(a & b) represents that the performance of EATHRP was compared to existing routing protocols like LEACH, Fuzzy C-Means, DKNN and DNN to assess their efficiency. The experimental results shows that the Proposed Method outperforms these protocols in various performance metrics. Energy efficiency for EATHRP was recorded at 95%, which is higher than LEACH (78%), Fuzzy C-means (85%), DKNN(88%), and DNN (92%). Additionally, EATHRP achieved 93% better network performance in terms of Loss Ratio, End-to-End delay, Buffer Occupancy and delay outperforming LEACH (75%), Fuzzy C-means (82%), DKNN (86%), and DNN (90%). These improvements highlight how EATHRP can enhance both energy and life span of the wireless sensor network.

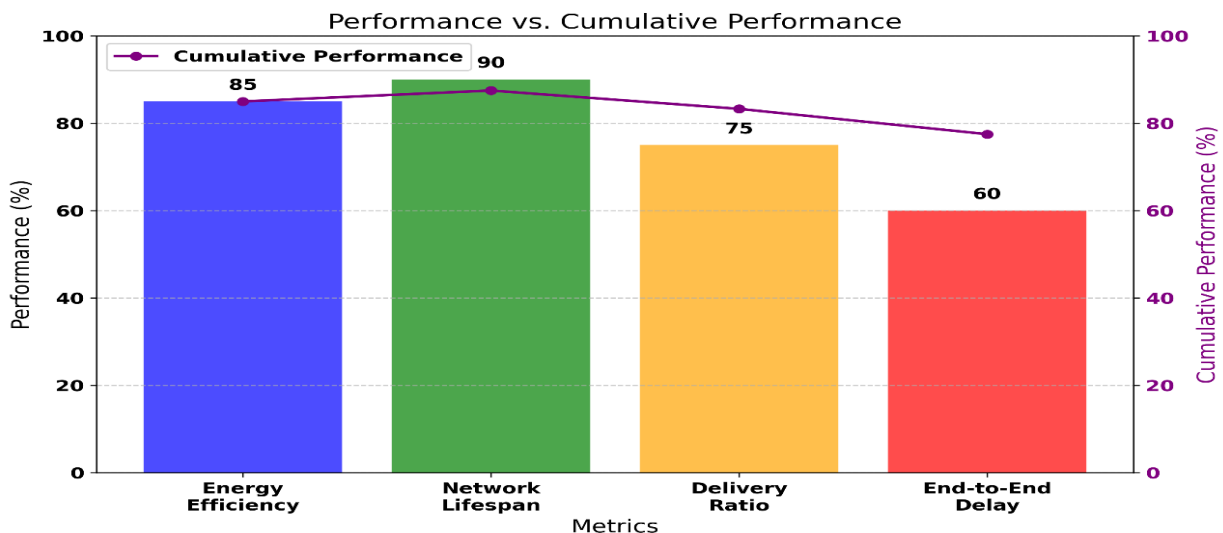


Figure 17 : Comparison of Cumulative Performance

Figure 17 compares the overall performance of End-to-End Delay, Network Lifespan, Energy efficiency and Delivery Ratio. Our proposed method outperforms the others by providing better Energy efficiency, longer Network Lifespan, higher Delivery Ratio and lower End-to-End Delay, while effectively delivering data packets. However, the packet delivery ratio decreases due to factors like increased overhead and interference, which lead to higher energy consumption in the network.

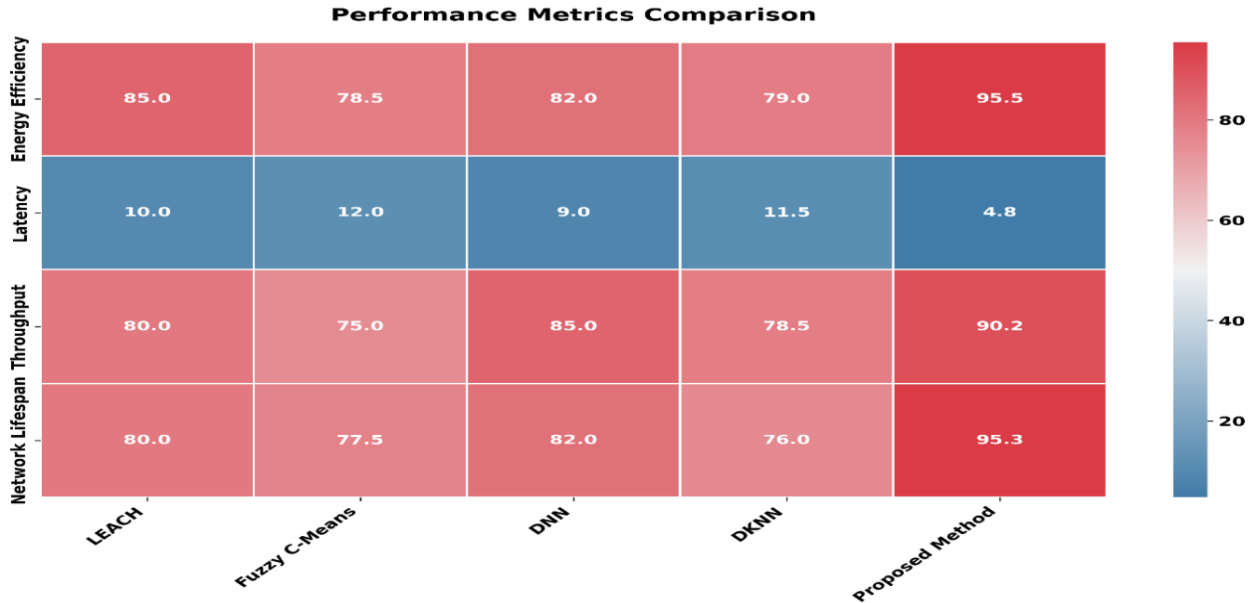


Figure 18: Performance comparison of Energy efficiency, Latency, Throughput, and Network Lifetime

Figure 18 shows the performance of the Network Lifetime, Throughput, Latency and Energy efficiency. The proposed method achieved an Energy efficiency of 95.5%, Throughput of 90.2%, Latency of 4.8% and Network Lifetime of 95.3% with less node friction, our proposed method performed better than the current methods.

Finally, the improvements show that the EATHRP can enhance energy conservation and data reliability in Wireless Sensor Networks (WSN) making it especially useful in energy-limited situations. The proposed method is used to smart clustering, optimized routing and secure data transmission effectively address the major challenges in WSNS. Proposed protocol use deep learning and bio-inspired optimization methods for movement, best path route which improves throughput, reduce latency, and extends the network’s lifespan.

V. CONCLUSION

This research proposed Energy Awareness and Trust based Hierarchical Routing Protocol (EATHRP) for Wireless Sensor Networks (WSNs). In this work, trust values were determined while choosing the best path, forming the cluster, and selecting the cluster head using both direct and indirect trust evaluations. The identified data was then transmitted to the BS securely using an enhanced approach. The Proposed protocol incorporates a meta-heuristic clustering approach for optimized cluster head selection along with a trained intelligent route-planning mechanism for secure data transmission to strengthen the WSN's energy and security. The experimental analyses reveal that EATHRP outperforms existing protocols, namely LEACH, FCM, DKNN, and DNN, in terms of energy efficiency, throughput, latency, and network lifetime with figures of 95.5%, 90.2%, 4.8%, and 95.3%, respectively. Accordingly, EATHRP is a lightweight, scalable, and energy-efficient protocol for wireless sensor network applications, such as environmental monitoring, healthcare, and industrial automation. Additional work can be done with adaptive reinforcement learning-based routing strategies and block chain-integrated trust mechanisms to make future WSN systems secure, scalable, and resilient in dynamic environments.

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my professor V. Raghunatha Reddy for his outstanding supervision, guidance, and encouragement. I am truly thankful for his generous involvement, valuable advice, and unwavering support during my PhD Work.

REFERENCES

- [1] S. Ramesh, R. Rajalakshmi, J. N. Dwivedi, S. Selvakannani, B. Pant, N. Bharath Kumar, and Z. Fissiha Demssie, "Optimization of leach protocol in wireless sensor network using machine learning," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 5393251, 2022.
- [2] D. L. Reddy, C. G. Puttamadappa, and H. N. G. Suresh, "Hybrid optimization algorithm for security aware cluster head selection process to aid hierarchical routing in wireless sensor network," *IET Communications*, vol. 15, no. 12, pp. 1561–1575, 2021.
- [3] G. Arya, A. Bagwari, and D. S. Chauhan, "Performance analysis of deep learning-based routing protocol for an efficient data transmission in 5G WSN communication," *IEEE Access*, vol. 10, pp. 9340–9356, 2022.
- [4] A. Behura and M. R. Kabat, "Energy-efficient optimization-based routing technique for wireless sensor network using machine learning," in **Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019**. Singapore: Springer, 2020, pp. 555–565.
- [5] Z. Zhao, D. Shi, G. Hui, and X. Zhang, "An energy-optimization clustering routing protocol based on dynamic hierarchical clustering in 3D WSNs," *IEEE Access*, vol. 7, pp. 80159–80173, 2019.
- [6] Y. Yao, D. Xie, Y. Li, C. Wang, and Y. Li, "Routing protocol for wireless sensor networks based on Archimedes optimization algorithm," *IEEE Sensors Journal*, vol. 22, no. 15, pp. 15561–15573, 2022.
- [7] P. Wang, J. Qin, J. Li, M. Wu, S. Zhou, and L. Feng, "Dynamic optimization method of wireless network routing based on deep learning strategy," *Mobile Information Systems*, vol. 2022, no. 1, p. 4964672, 2022.
- [8] H. Singh and D. Singh, "Hierarchical clustering and routing protocol to ensure scalability and reliability in large-scale wireless sensor networks," *The Journal of Supercomputing*, vol. 77, no. 9, pp. 10165–10183, 2021.
- [9] A. P. Abidoye and B. Kabaso, "Energy-efficient hierarchical routing in wireless sensor networks based on fog computing," *EURASIP Journal on Wireless Communications and Networking**, vol. 2021, no. 1, p. 8, 2021.
- [10] P. Bedi, S. Das, S. B. Goyal, P. K. Shukla, S. Mirjalili, and M. Kumar, "A novel routing protocol based on grey wolf optimization and Q learning for wireless body area network," *Expert Systems with Applications*, vol. 210, p. 118477, 2022.
- [11] R. K. Yadav and R. P. Mahapatra, "Energy aware optimized clustering for hierarchical routing in wireless sensor network," *Computer Science Review*, vol. 41, p. 100417, 2021.
- [12] H. Esmacili, B. M. Bidgoli, and V. Hakami, "CMML: Combined metaheuristic-machine learning for adaptable routing in clustered wireless sensor networks," *Applied Soft Computing*, vol. 118, p. 108477, 2022.
- [13] T. A. Alghamdi, "Energy efficient protocol in wireless sensor network: optimized cluster head selection model," *Telecommunication Systems*, vol. 74, no. 3, pp. 331–345, 2020.
- [14] J. Daniel, S. F. V. Francis, and S. Velliangiri, "Cluster head selection in wireless sensor network using tunicate swarm butterfly optimization algorithm," *Wireless Networks*, vol. 27, pp. 5245–5262, 2021.
- [15] P. Parthiban and V. S. Vaisakhi, "Efficient malicious node detection by multi-objective energy trust aware hybrid optimization based maximizing lifetime of wireless sensor networks," *Journal of Intelligent & Fuzzy Systems*, preprint, pp. 1–15.
- [16] A. Keerthika and V. Berlin Hency, "Reinforcement-Learning based energy efficient optimized routing protocol for WSN," *Peer-to-Peer Networking and Applications*, vol. 15, no. 3, pp. 1685–1704, 2022.
- [17] A. F. E. Abadi, S. A. Asghari, M. B. Marvasti, G. Abaei, M. Nabavi, and Y. Savaria, "RLBEEP: Reinforcement-learning-based energy efficient control and routing protocol for wireless sensor networks," *IEEE Access*, vol. 10, pp. 44123–44135, 2022.
- [18] Z. Liu, Y. Liu, and X. Wang, "Intelligent routing algorithm for wireless sensor networks dynamically guided by distributed neural networks," *Computer Communications*, vol. 207, pp. 100–112, 2023.
- [19] Z. Wang, L. Shao, S. Yang, J. Wang, and D. Li, "CRLM: A cooperative model based on reinforcement learning and metaheuristic algorithms of routing protocols in wireless sensor networks," *Computer Networks*, vol. 236, p. 110019, 2023.
- [20] I. Surenter, K. P. Sridhar, and M. K. Roberts, "Maximizing energy efficiency in wireless sensor networks for data transmission: A Deep Learning-Based Grouping Model approach," *Alexandria Engineering Journal*, vol. 83, pp. 53–65, 2023.
- [21] X. Xue, R. Shanmugam, S. Palanisamy, O. I. Khalaf, D. Selvaraj, and G. M. Abdulsahib, "A hybrid cross layer with harris-hawk-optimization-based efficient routing for wireless sensor networks," *Symmetry*, vol. 15, no. 2, p. 438, 2023.

- [22] Bekal, P., Kumar, P., Mane, P.R.: A metaheuristic approach for hierarchical wireless sensor networks using particle swarm optimization-based Enhanced LEACH protocol. *IET Wirel. Sens. Syst.* 14(6), 410–426 (2024). <https://doi.org/10.1049/wss2.12091>
- [23] Y. Liu, H. Huang and J. Zhou, "A Dual Cluster Head Hierarchical Routing Protocol for Wireless Sensor Networks Based on Hybrid Swarm Intelligence Optimization," in *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16710-16721, 1 May1, 2024.
- [24] S. Ramalingam, S. Dhanasekaran, S. S. Sinnasamy, A. O. Salau, and M. Alagarsamy, "Performance enhancement of efficient clustering and routing protocol for wireless sensor networks using improved elephant herd optimization algorithm," *Wireless Networks*, vol. 30, no. 3, pp. 1773-1789, 2024.
- [25] S. He, Q. Li, M. Khishe, A. Salih Mohammed, H. Mohammadi, and M. Mohammadi, "The optimization of nodes clustering and multi-hop routing protocol using hierarchical chimp optimization for sustainable energy efficient underwater wireless sensor networks," *Wireless Networks*, vol. 30, no. 1, pp. 233-252, 2024.
- [26] H. Hu, Y. Han, M. Yao, and X. Song, "Trust based secure and energy efficient routing protocol for wireless sensor networks," *IEEE Access*, vol. 10, pp. 10585-10596, 2021.
- [27] T. A. Alghamdi, "Energy efficient protocol in wireless sensor network: optimized cluster head selection model," *Telecommunication Systems*, vol. 74, no. 3, pp. 331-345, 2020.
- [28] O. Al-Baik, S. Alomari, O. Alsayed, S. Gochhait, I. Leonova, U. Dutta, O. P. Malik, Z. Montazeri, and M. Dehghani, "Pufferfish Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *Biomimetics*, vol. 9, no. 2, p. 65, 2024
- [29] P. Trojovský and M. Dehghani, "Muxuan Han, Zunfeng Du, Kum Fai Yuen, Haitao Zhu, Yancang Li, Qiuyu Yuan, Walrus optimizer: A novel nature-inspired metaheuristic algorithm, *Expert Systems with Applications*, Volume 239,2024,122413, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2023.122413.2022>.
- [30] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," *Neural Computing and Applications*, 2023.
- [31] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, p. 110011, 2023.
- [32] P. Sharma, M. Sharma, R. Singh, V. Kumar, R. Agarwal and P. K. Malik, "NHARSO-IWSN: A Novel Hybridized Adaptive-Network-Based Fuzzy Inference System with Reptile Search Optimization Algorithm-Based Routing Protocol for Internet of Things-Enabled Wireless Sensor Networks," in *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 6293-6302, Aug. 2024, doi: 10.1109/TCE.2024.3418845.
- [33] Ning Liu, Jun Wang, Fazhan Tao, Zhumu Fu, Bo Liu, EDRP-GTDQN: An adaptive routing protocol for energy and delay optimization in wireless sensor networks using game theory and deep reinforcement learning, *Ad Hoc Networks*, Volume 166,2025,103687, ISSN 1570-8705, <https://doi.org/10.1016/j.adhoc.2024.103687>.
- [34] V. Saravanan, Indhumathi G, Ramya Palaniappan, Narayanasamy P, M. Hema Kumar, K. Sreekanth, Navaneethan S, A novel approach to node coverage enhancement in wireless sensor networks using walrus optimization algorithm, *Results in Engineering*, Volume 24,2024,103143, ISSN 2590-1230, <https://doi.org/10.1016/j.rineng.2024.103143>.
- [35] Lucia K. Ketshabetswe, Adamu Murtala Zungeru, Caspar K. Lebekwe, Bokani Mtengi, A compression-based routing strategy for energy saving in wireless sensor networks, *Results in Engineering*, Volume 23,2024,102616, ISSN 2590-1230, <https://doi.org/10.1016/j.rineng.2024.102616>.
- [36] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details