International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(23s) 2026 pp. 941-965
https://doi.org/10.70917/ijcisim-2026-5656
Article

AN ENHANCED K-MEDOID AND HYBRID OPTIMIZATION ALGORITHM FOR IMPROVING CLUSTER-BASED ROUTING PROTOCOL IN WIRELESS SENSOR NETWORKS

K. Tamilarasi[1], M. Hemalatha[2]
[1] Department of Computer Science, Hindusthan College of Arts & Science, Coimbatore- 641028, India.
Email: tamilkrishna15phd@gmail.com
[2] Department of Information Technology, Hindusthan College of Arts & Science, Coimbatore- 641028, India.
Email: hemalathabalachandar02@gmail.com

Abstract: Wireless Sensor Networks (WSNs) offer a robust technology for sensing and transmitting data across large geographic areas. However, restricted power, communication, and processing capabilities in WSNs can have a substantial influence on the network's longevity. To overcome these restrictions and enhance the energy efficiency of WSNs, sensor nodes are typically connected in groups, and the shortest path is established. Over the past years, many studies have been developed to cluster the nodes and choose the Cluster Head (CH) nodes. Among them, an Adaptive Sailfish Optimization (ASFO) with K-medoids outperforms in clustering and CH selection by reducing distance and latency among nodes. However, the limitation of K-medoids is that it may have difficulty establishing clusters due to the random selection of medoid locations. The random medoid initialization provides inferior results and degrades clustering quality. Also, the ASFO is presently facing the challenge of addressing optimization problems over extended periods. Hence, this article proposes a new hybrid with parallel processing energy-efficient cluster-based routing algorithm for WSNs with high energy efficiency and network longevity. The key goal of this study is to optimize the CH selection and achieve energy efficiency in WSNs. This algorithm comprises two major phases: CH selection and data transmission. First, the sensor nodes in WSN are clustered using the Fisher Median Naive Sharding (FMNS)-based K-Medoids scheme. Then, a new combination of Dove Optimization Algorithm (DOA) with SFO called DSFOA is utilized to select the optimal CHs. Furthermore, the tree-based multi-hop routing scheme is utilized to determine the shortest route for energy-efficient data transfer. The entire proposed work is named as Fisher K-medoids with Amalgamation of Dove and Sailfish Optimization-based Energy-Efficient Clustering and Routing Algorithm (FKADSO-EECRA). Finally, simulation results exhibit that the FKADSO-EECRA achieves high Packet Delivery Ratio (PDR), throughput, and network lifetime while reducing End-to-End Delay (E2D), jitter, energy consumption, and routing overhead compared to the conventional cluster-based routing algorithms.
Keywords: WSN, Cluster-based routing, K-medoids, Sailfish optimization, Fisher median naive sharding, Dove optimization

1. INTRODUCTION

WSNs include small, inexpensive sensor nodes. One of the best methods to transmit information from remote locations to a central data processing station, known as Base Station (BS) is WSNs. For this purpose, these nodes self-organization is employed to establish a multi-hop network. [1]. Multimedia WSNs can gather additional data for intelligent buildings, low-cost healthcare, military monitoring, and monitoring of agricultural and industrial applications. They can also send multimedia data, including photographs and videos [2]. Larger sensor nodes might consume more resources, and multimedia data is usually more comprehensive. Researchers have tried to look into and resolve this problem [3]. One of the most important ways for nodes in WSNs to prolong their lives is energy saving. Energy is mostly used for packet transmission and reception [4].

Batteries are commonly used by sensor nodes in WSNs. The capacity of the battery becomes the most valuable resource in WSNs, yet charging the battery is complicated. Energy saving thus becomes a crucial concern in WSNs. To optimize network longevity and energy efficiency, a clustering algorithm has to be created. One of WSN's power management features is clustering, which divides the network into several clusters and designates a CH node in each cluster [5]. The CH lowers the overhead of the BS by aggregating the data that is obtained through every node and forward into BS. In resource-constrained scenarios, the WSN saves electricity since BS takes information via less nodes [6]. In WSNs, clustering methods help lower power usage. There are two stages in each cycle of the clustering algorithm's operation: cluster formation and maintenance.

Copyright: © 2026 by the authors

The nodes were arranged into discrete collections, known as clusters, along with CH. The CH gathers information from the sensor nodes and sends it to the receiver. Several data aggregation techniques are used to combine the information from every node in the cluster as well transmit it to the BS in the form of crucial information. The CH is necessary to sustain the energy reductions achieved by clustering approaches.

To manage data transmission through clustering, CH selection, and routing, several routing algorithms have been devised to extend the lifetime of WSNs [7]. The most common routing technique utilized in WSNs is referred to as Low-Energy Adaptive Clustering Hierarchy (LEACH) [8]. By reducing energy depletion through clustering and CH selection, it increases the network's longevity. It randomly selects several nodes to serve as CHs, which are subsequently switched among CMs to ensure equitable energy allocation. After receiving data packets from respective CMs, the CHs aggregate them to reduce unnecessary data transfers, lowering energy costs and increasing network throughput. It reduces transmission delays and collisions inside and across clusters by allocating data transfer times for all nodes using the Time Division Multiple Access (TDMA) scheduling policy. However, the primary disadvantages of LEACH include the random selection of CH based on probability, the neglect of residual energy in the CH selection process, and the reliance on single-hop data transfer to the BS.

Recent advancements in unsupervised Machine Learning (ML) techniques, such as clustering, have addressed the drawbacks of traditional routing algorithms in WSNs [9]. By minimizing the computational demands associated with CH selection and routing in WSNs, the clustering schemes, including K-means and K-medoids, enable multi-hop communication [10]. However, clustering and routing continue to be typical NP-hard issues, making it difficult for existing approaches to identify optimal solutions. Thus, swarm intelligence approaches like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and others have been developed with unsupervised ML algorithms to find optimum solutions for clustering and routing optimization [11]. For example, to select the CH using the saved energy, distance to the sink, node density, along with mean distances of CM nodes, a hybrid of the Multiple Criteria Decision Making (MCDM) and K-means algorithms was introduced in [12]. In a similar vein, a novel routing method was introduced in [13] to increase the WSN lifespan by integrating the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) technique with K-means clustering. In [14], K-means clustering with a firefly algorithm was used to dynamically group nodes and modify them based on load criteria. A hybrid K-means ant Lion optimization for Energy-efficient clustering-based Routing (KLionER) was suggested for WSNs in [15]. Similarly, K-means-based Bee Colony Optimization (K-BCO) was applied in [16] for heterogeneous WSN clustering. In [17], improved K-means clustering and adaptive distance threshold techniques were presented to reduce the energy usage in WSNs. However, the disadvantage of K-means clustering was its sensitivity to the original cluster centers, which resulted in imbalanced clusters, excessive energy depletion, and poor network performance. Furthermore, because CH selection is based on cluster centroids, it may result in longer intra-cluster distances and greater communication costs.

To solve this problem, a new algorithm as stated to the MCDM method with K-medoids clustering was developed in WSN in the presence of uncertainty [18]. The nodes were clustered using K-medoids scheme and TOPSIS technique was used to choose CHs. In [19], an ASFO algorithm with K-medoids has been presented to cluster the nodes and pick the optimal CHs. Additionally, the E-CERP was utilized for determining the shortest route and dynamically reducing network overhead. However, one of its limitations is that it may have difficulty establishing clusters due to the random selection of medoid locations. The random medoid initialization provides inferior results and degrades clustering quality. Additionally, the ASFO are presently facing the challenge of addressing optimization problems over extended time periods.

These issues should be addressed by novel hybrid algorithms with parallel processing, which offer excellent efficiency in optimization tasks. Therefore, this article proposes a new energy-efficient cluster-based routing algorithm in WSNs with high energy efficiency and network longevity. The key goal of this algorithm is to optimize the CH selection and achieve energy efficiency in WSNs. The contributions of this work include:

- In the clustering and CH selection phase, the FMNS-based K-Medoids is introduced to enhance the clustering performance. The FMNS is used to choose the medoid points for clustering. First, the naïve sharding splits the nodes into manageable subsets. The Fisher median determines the CH. Then, a new DSFOA is proposed to select the new optimal CH effectively.
- In the data routing phase, the tree-based multi-hop routing strategy are used to determine the best shortest route for energy-efficient data transfer while maintaining high communication link dependability and minimal packet latency.
- This entire work is called FKADSO-EECRA, and simulations are conducted under various scenarios, such as varying node densities and rounds, to comprehend the efficiency of FKADSO-EECRA.

The remaining portions have been provided as outlined below: Section 2 comprises a literature work. Section 3 provides a detailed overview of the FKADSO-EECRA in WSNs, while Section 4 presents the simulation results. Section 5 provides a summary of the study and outlines its potential future scope.

2. LITERATURE SURVEY

Numerous scholars have investigated routing and clustering strategies in WSNs. This section reviews some of recent algorithms in detail. A Novel Fuzzy Clustering and Routing Protocol (NFCRP) based on the improved PSO was presented in [20]. To determine the best cluster, a Fuzzy Inference System (FIS) was first utilized. The best route was then determined using the FIS optimized by the improved PSO. On the other hand, because node mobility effects were not taken into account, energy consumption remained high.

To generate optimum clusters and choose the best CHs for WSNs, a Distributed PSO with Fuzzy Clustering Protocol (DPFCP) was introduced [21]. However, the network's energy usage was still high. The Bacterial Foraging Optimization with Harmony Search Algorithm (BFO-HSA) was utilized to select CHs in [22]. They also utilized an Energy-efficient Routing utilizing Fuzzy Neural Network (ERFN) to choose the next hop for path creation. However, the PDR and throughput were poor.

By merging Harris Harks Optimization Clustering with Fuzzy Routing (HHOCFR), a novel protocol was presented in [23]. Using a novel encoding approach, ideal optimum clusters were created, and the best CHs were selected by the HHO scheme. Furthermore, the Fuzzy Logic (FL) was used to calculate the relay CH for each CH. To avoid frequent re-clustering, a novel adaptive cluster management technique was employed to rotate CHs inside clusters and initiate re-clustering in the network. However, the network lifespan and energy consumption were worse because other variables, like node and traffic density, were not taken into account.

By considering factors like distance, node energy, density, and capacity of nodes, Pravin et al. [24] proposed the Stochastic CH Selection Model (SCHSM) using GA to enhance energy efficiency in heterogeneous WSNs. However, the cluster lifespan was found to be low. The SFO and Spotted Hyena Optimization (SHO) were adopted to determine the best CH and route, respectively, in WSNs [25]. However, PDR and throughput continued to be poor. A hybrid Whale-Ant Optimization Algorithm (WAOA) was created for WSN in [26], which finds the best route between the source and CHs in the predetermined search space. However, the Packet Loss Ratio (PLR) remained high and requires further consideration when selecting a CH.

To choose the best CH in WSNs, an Improved Squirrel Search Algorithm (I-SSA) was applied in [27]. However, the network lifespan was short, and energy consumption was considerable. A Modified ACO Algorithm (MACOA) was utilized to boost energy efficiency in WSNs [28]. First, energy use was simultaneously optimized while ensuring short route lengths, bandwidth, and reliability using a multi-objective heuristic algorithm. They then developed an adaptive pheromone decay policy that dynamically adjusts according to network parameters, including node energy and connection reliability, to select energy-efficient routes.

To avoid congestion for WSNs, a Hybrid Trust-based Congestion-aware Cluster Routing (HTCCR) approach was introduced [29]. This protocol used hybrid K-Harmonic Means (KHM) and Enhanced Gravitational Search Algorithm (EGSA) to determine the node probability. The nodes with the highest fitness scores were then chosen to serve as CHs. Additionally, packet forwarding is executed according to the designated priority level was effectively controlled by combining priority-based data delivery. However, to select the optimal CH and route, other characteristics were required. In [30], intelligent routing was combined with uneven clustering to create a novel energy-efficient protocol termed FOAEAUC-SARP for WSNs. The best CHs were chosen by the Fox Optimization Algorithm (FOA), whereas the best route was determined by the Snake Algorithm (SA). Conversely, energy efficiency and throughput were low. The aforementioned algorithms are summarized in Table 1 based on their merits, demerits, and performance metrics.

Table 1. Summary of Current Clustering and Routing Algorithms in WSNs

Ref. No.	Algorithms	Benefits	Drawbacks	Performance metrics
[20]	NFCRP	It achieved a maximum network lifetime and throughput.	Energy consumption remained high due to the lack of considering node's mobility factors.	**No. of rounds = 100:** Alive nodes = 100 **No. of nodes = 100:** Network throughput = 14 x10^7; Energy consumption = 12 J

[21]	DPFCP	It can increase the network lifetime and throughput.	Network energy consumption remained high.	**No. of nodes = 100:** Network throughput = $8x10^8$ **No. of rounds = 2000:** Alive nodes = 5
[22]	BFO-HSA and ERFN	It achieved better network lifetime and delay.	Throughput and PDR were low.	**No. of nodes = 500:** Energy consumption = 100 mJ; Network lifetime = 6000 rounds; Throughput = 0.68 Mbps; E2D = 4.5 ms; PDR = 81%; PLR = 5%
[23]	HHOCFR	It has a high network throughput.	Other factors like node density, traffic density was not considered, which leads to degrade the network lifetime and energy use.	**No. of nodes = 100:** Network throughput = $4.45x10^8$ **No. of rounds = 1600:** Energy usage = 100 J
[24]	SCHSM using GA	It increased PDR and reduced transmission delay significantly.	Cluster lifetime was low.	**Node density = 150:** Packet Delivery Ratio (PDR) = 410000 kbps; Overhead = 0.4; Cluster lifetime = 0.5 ms; Transmission delay = 3 ms; Residual energy = 9.86 J
[25]	SFO-SHO	It has a lower delay and latency, as well as higher throughput.	Network lifetime and PDR remained low.	**Node count = 1000:** Network lifetime = 650 sec; Mean remaining energy = 0.15 J; No. of dead nodes = 750; Throughput = 635 kbps; PDR = 0.87; Delay = 152 ms; Latency = 3.5 sec
[26]	WAOA	It efficiently increased the network lifetime.	Packet drop ratio was high	First node died = 520 rounds; Half of nodes died = 1380 rounds; Last node died = 14000 rounds; Packet drop ratio = 17.5%; Routing overhead = 18%
[27]	I-SSA	It reduced E2D significantly.	Energy consumption was high and network lifetime was low.	**No. of nodes = 1000:** PDR = 88%; Throughput = 0.5 bps; Energy usage = 210 J;

				E2D = 10 ms
[28]	MACOA	It improved network longevity and throughput effectively.	Path optimization remained low efficient, which necessitates ML scheme for further enhancements.	**No. of nodes = 100:** Network lifetime = 1150 rounds; Network stability = 1110 rounds; Mean throughput = 320 Mbps
[29]	HTCCR	It increased mean throughput and reduced delay significantly.	It requires additional parameters such as node centrality, node density, delay, etc., for CH and path selection.	**Traffic rate = 20 pps:** E2D = 200 ms; Mean throughput = 105 kbps
[30]	FOAEAUC-SARP	It has high energy efficiency and network lifespan.	Throughput remained low.	No. of packets to BS = 17266; First node died = 1550 rounds; Half of node died = 1750 rounds

From the literature, it can be inferred that most existing algorithms still suffer from high energy usage, low throughput, and high E2D. This is due to the ineffective selection of CHs based on random initialization and the lack of network factors for optimization, such as node density and CH stability. Additionally, existing metaheuristic optimization algorithms still struggle with slow convergence speed and take more time for large-scale WSNs. As a result, this study resolves these issues by integrating FMNS-based K-Medoids for efficient clustering, a hybrid DSFOA for optimal CH selection, and tree-based multi-hop routing strategy for energy-efficient data transfer. This results in higher energy efficiency and network longevity with minimum delay and overhead.

3. PROPOSED METHODOLOGY

This section briefly explains the FKADSO-EECRA for WSNs. Figure 1 depicts a schematic representation of this study.

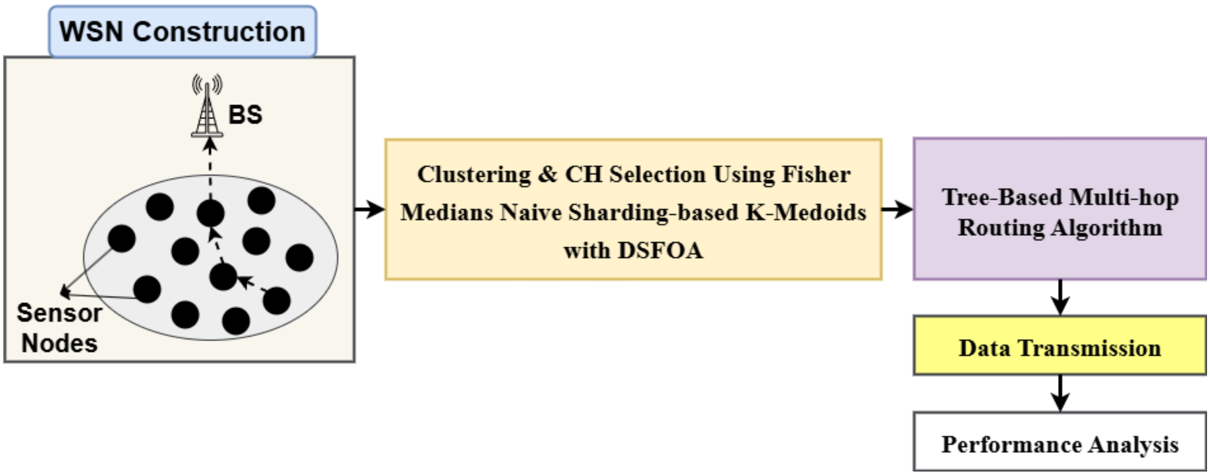

Figure 1. Schematic Illustration of the FKADSO-EECRA

3.1 Network Model

The WSN setup consists of a BS and multiple sensor nodes. By grouping nodes into clusters, it reduces the overall network's energy usage. The CM nodes in each cluster continuously observe the environment and transmit sensory information to the CH. One of the CM nodes is always selected to be the CH node. The CH is responsible for collecting data from each CM node and relaying it to the BS. Avoiding a direct interaction between sensors and BS is made easier by this clustering process. Figure 2 shows the network model.

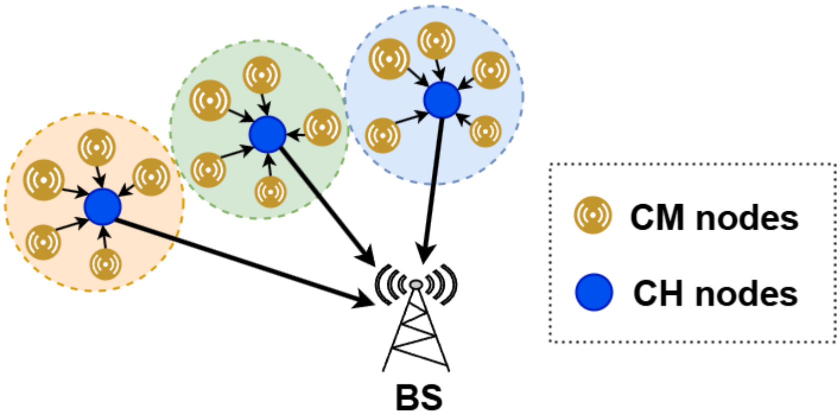

Figure 2. Network Model

.

3.2 Energy Model

A node's energy is directly proportional to the propagation distance d when the threshold distance (d_0) is larger than d. The M-bit data packet transmission energy is represented by Eq. (1):

$$E_{TX}(M,d) = E_{elec} \times M + \varepsilon_{amp} \times M \quad (1)$$

$$\text{Where } \varepsilon_{amp} = \begin{cases} \varepsilon_f \times d^2, & d \leq d_0 \\ \varepsilon_{mp} \times d^4, & d > d_0 \end{cases} \quad E_{R\_CH} = E_s \times \delta_{CH} \times \mathfrak{b} \quad (2)$$

In Eqns. (1)-(2), E_{TX} represents the overall amount of energy needed to transmit data, E_{elec} refers to the dissipation of energy per bit, ε_f denotes n the free space model, the energy that is used for amplification, ε_{mp} denotes as energy spent in the multipath model, and M indicates the number of bits. The energy usage of the receiver is characterized as

$$E_{RX}(M) = E_{elec} \times M \quad (3)$$

In Eq. (3), E_{RX} refers to the overall energy needed to receive M bits. Each CH collected information packets from its CMs, aggregates them, and sends to the BS. So, the amount of energy utilized by the CH is determined as:

$$E_{CH} = \begin{cases} n_{CM}(E_{elec} + E_{DA}) \times M\varepsilon_f d_{CH}^2, & d_{CH} \leq d_0 \\ n_{CM}(E_{elec} + E_{DA}) \times M\varepsilon_{mp} d_{CH}^4, & d_{CH} > d_0 \end{cases} \quad (4)$$

In Eq. (4), n_{CM} denotes as sum amount of CMs established a connection with that CH, E_{DA} is the energy spent by the CH for aggregating information from the CMs, and d_{CH} is the distance between CH and BS.

3.3 Network Factors Used for Clustering and CH Selection

Remaining Energy (RE): Reducing energy consumption is critical since it influences how long a network lasts, and the value is computed by summing the current energy of each chosen CH. Because the total quantity of energy is efficiently maximized, taking the reciprocal ensures that each goal function is perfectly balanced. This study looks for CHs that maximize energy usage across the network by incorporating the leftover energy from each CH into the objective function. The following is the reciprocal of the total energy of all chosen CHs:

$$f_1 = \max\left(\frac{1}{\sum_{j=1}^{m}(E_{CH_j})}\right) \quad (5)$$

In Eq. (5), m denotes as total amount of CHs and E_{CH_j} is the current energy of j^{th} CH.

Distance between CM and CH Nodes: Intra-cluster distance is the gap between each CM and CH in a certain cluster. The average intra-cluster distance factor, f_2, is computed to reduce the distance among the CMs and their respective CHs.

$$f_2 = min\left(\sum_{j=1}^{m}\left(\frac{1}{n_{CM}} D_{CM-CH_j}\right)\right) \quad (6)$$

In Eq. (6), n_{CM} represent the number of CMs in the j^{th} cluster and D_{CM-CH_j} is the Euclidean distance between CM and j^{th} CH.

Distance between CH and BS: The amount of sensor nodes attached to the CH is divided by the distance among the CH and the BS to determine the mean sink distance. The BS distance factor, represented by f_3, accounts for the distance among the CHs and the BS.

$$f_3 = min\left(\sum_{j=1}^{m}\left(\frac{1}{n_{CM}} D_{CH_j-BS}\right)\right) \tag{7}$$

In Eq. (7), D_{CH_j-BS} represents the Euclidean distance among j^{th} CH and BS.

CH Stability: Because more nodes inside the cluster lead to more energy dissipation, it is crucial to maintain a balance in cluster size. Clusters of various sizes, both small and large, may be formed to optimize energy utilization through the implementation of sensor nodes. The necessity of equal-sized clusters in the network is shown by the CH stability f_4 in Eq. (8).

$$f_4 = max\left(\sum_{c=1}^{C}\frac{N}{m} - ClusterSet(N_c)\right) \tag{8}$$

In Eq. (8), N denotes as total amount of nodes in the entire network, C represents the number of clusters, and $\mathrm{ClusterSet}(N_c)$ indicates the number of nodes in cluster c.

Cluster Reliability (CR): It is an average measure across all clusters and calculated in Eq. (9).

$$f_5 = max\left(\frac{1}{C}\sum_{c=1}^{C}\left(1 - \frac{ClusterSet(N_c)}{ClusterSet(N_c)_{max}+1}\right)\right) \tag{9}$$

In Eq. (9), C denotes as total amount of clusters and $ClusterSet(N_c)_{max}$ represents the maximum quantity of nodes in cluster c.

Node Centrality (NC): The centrality of node (n_i) is determined by Eq. (10).

$$f_6 = max\left(\frac{\sum_{i,k=1}^{N}\left(D_{n_i-n_k}\right)}{|Neighbor(n_i)|}\right), k \neq i \tag{10}$$

In Eq. (10), $|Neighbor(n_i)|$ represents the set of neighbors for node n_i within its communication range R and $D_{n_i-n_k}$ is the distance between node n_i and its neighbor n_k.

Node Density (ND): The CH area's increased node density contributes to energy conservation and protection. The main cause of the nodes' frequent energy depletion is intra-cluster transmission. As a result, a reliable CH node is chosen from the available nodes. The average distance between adjacent nodes within a cluster, or node density f_7, is determined using Eq. (11).

$$f_7 = max\left(\frac{\sum_{i=1}^{N}\left(D_{n_i-n_k}\right)}{N} \times \frac{1}{\sum_{j\in m} D_{CH_j-BS}}\right) \tag{11}$$

Node Degree (NDeg): It is calculated using Eq. (12).

$$f_8 = max\left(\frac{\sum_{c=1}^{C}\sum_{j\in NeighborCH(CH_j),j=1,\dots m}|ClusterSet(N_c)|}{\sum_{c=1}^{C}\Psi(CH_c)}\right) \tag{12}$$

In Eq. (12), C represents as total amount of clusters, $\Psi(CH_c)$ indicates the quantity of hops from CH in the c^{th} cluster to the BS, $|ClusterSet(N_c)|$ denotes as number of nodes in cluster c, and $NeighborCH(CH_j)$ is the set of neighbor CHs of CH_j.

Delay: Network latency, which is closely related to the sum quantity of nodes in a collection, must be kept to a minimum to select an effective CH. The ideal cluster size must be attained by limiting the quantity of nodes in every cluster, which means that the latency may grow as the number of cluster members increases. The fitness of delay f_9 is computed using Eq, (13).

$$f_9 = min\left(\sum_{i,j\in N}\left(\frac{1}{\sigma-\rho} + \frac{J}{\mu} + \frac{d_{ij}}{\beta}\right)\right) \tag{13}$$

In Eq. (13), σ is the service rate, ρ is the new packets entry rate, J is the packet size, μ is the link bandwidth, d_{ij} is the physical link length of nodes i to j, and β is the speed of propagation using M/M/1 queuing theory.

Communication Overhead: Packet size and distance both influence communication overhead. This is defined by the packet size to hop count required for the PDR.

$$f_{10} = min\left(\sum_{i=1}^{N} \frac{S_i \times H_i}{DR_i}\right) \quad (14)$$

In Eq. (14), S_i is the packet size of i^{th} node, H is the number of hops needed by i^{th} node for transmission, and DR is the PDR of i^{th} node.

As a result, these criteria may be merged into a fitness function of each node is represented as:

$$f(x) = \omega_1 f_1 + \omega_2 \frac{1}{f_2} + \omega_3 \frac{1}{f_3} + \omega_4 f_4 + \omega_5 f_5 + \omega_6 f_6 + \omega_7 f_7 + \omega_8 f_8 + \omega_9 \frac{1}{f_9} + \omega_{10} \frac{1}{f_{10}} \quad (15)$$

In Eq. (15), $\omega_1, \dots, \omega_{10}$ are the weight coefficients of respective network factors. This guarantees balancing energy efficiency and network performance.

3.4 Fisher Median Naive Sharding-Based K-Medoids with Dove Sailfish Optimization Algorithm for Clustering and CH Selection

After initializing N sensor nodes, the nodes are then arranged into collections, with each cluster allocated a CH for data exchange. This study leverages the FMNS-K-medoids clustering scheme, which divides N nodes into η clusters based on the network factors such as RE, distance between CMs and CH (d_{CM-CH}), NC, ND, NDeg, and CR. These factors for each node can be represented as:

$$\mathcal{X} = \{x_1, \dots, x_z\} \quad (16)$$

In Eq. (16), z represents the sum quantity of network factors considered for clustering. The value of η is calculated by

$$\eta = \sqrt{\frac{N}{2}} \quad (17)$$

In Eq. (17), the medoid point (i.e., node) in every cluster, which corresponds to the cluster's centroid point is called first CH node. The steps involved in this FMNS-K-medoids scheme are described below.

Step 1: In addition to initializing η clusters, the CH (C_η) is selected using the FMNS computation. The FMNS is determined by the summation value of all node factors. Based on this value, data points are divided into the number of shards. The shard's median is then determined; moreover, the group of medians is considered as the CH. So, the CH (C_η) is calculated by

$$C_\eta = \frac{\sum_{z=1}^{z} \hbar\epsilon(\mathcal{X})}{\hbar(\sigma^z)} \quad (18)$$

In Eq. (18), $\hbar$ represents the total amount of input factors, $\epsilon(\mathcal{X})$ denotes as median of the factors, and σ^z is the standard variance of the corresponding factor.

Step 2: The distance between the CH factor and each nodes' factor is calculated by the Euclidean distance as:

$$d(C_\eta, \gamma) = \sqrt{\sum_1^N (\gamma - C_\eta)^2} \quad (19)$$

In Eq. (19), $d(C_\eta, \gamma)$ is the distance between C_η and node factor $\mathcal{X}$.

Step 3: Based on $d(C_\eta, \gamma)$, each node is allocated to the closest centroid (C_η). After, the new position of the centroid point $(C_{\eta p})$ is determined by the DSFOA, which is described in the below section.

Step 4: At last, each node is grouped into the closest centroid points $(C_{\eta p})$. The grouped clusters nodes are presented as:

$$\eta_{clus} = \{\eta_1, \eta_2, \dots, \eta_C\} \quad (20)$$

In Eq. (20), η_{clus} is the grouped cluster nodes and η_C is the number of clusters.

i. Dove Optimization Algorithm

People frequently witness doves scavenging for crumbs in plazas, while all individual birds actively seek food. Some doves find contentment, while others remain unsatisfied [31]. Doves that do not locate adequate crumbs will continue to fly in search of more food. It is observed that fed doves will amass the majority of the available crumbs. Following observations on dove foraging behavior, this approach determines optimization using the fitness function $\left(\boldsymbol{f}(\boldsymbol{x})\right)$, determined in Eq. 15. Each network factor range $\boldsymbol{x}$ indicates a site where crumbs occur, and $\boldsymbol{f}(\boldsymbol{x})$ reflects the number of crumbs found at $\boldsymbol{x}$. The best solution is the one with the most crumbs.

Step 1: Set the dove population (network factor range) and distribute them around the solution area. The doves adhere to a fixed population of $\boldsymbol{N}$. The space can accommodate various distributions of doves, although experts promote constructing a uniform distribution over rectangular region (network factor region covering minimum $(\boldsymbol{l})$ and maximum ranges $(\boldsymbol{u})$).

Step 2: Configure the quantity of epochs, $\boldsymbol{e} = \boldsymbol{0}$, and establish the level of satiety, $\boldsymbol{s_d^e}$ for dove $\boldsymbol{d}$, $\boldsymbol{d} = \boldsymbol{1}, \dots, \boldsymbol{N}$. There are two approaches to initialize the position vector (initial centroid position $\boldsymbol{C_\eta}$) $\boldsymbol{x} \subset \boldsymbol{R^M}$ of dove $\boldsymbol{d}$. The easiest approach is to randomize $\boldsymbol{x}$ around the solution space. The alternative option is to use the lattice initialized approach. The following steps are outlined below:

Step 2.1: Initialization of the cells located at the four corners: The weight vectors corresponding to the four corners of the network are set as:

$$\boldsymbol{x_{1,1}} = (\boldsymbol{l_1}, \dots, \boldsymbol{l_z})^{\boldsymbol{T}} \tag{21a}$$

$$\boldsymbol{x_{A,B}} = (\boldsymbol{u_1}, \dots, \boldsymbol{u_z})^{\boldsymbol{T}} \tag{21b}$$

$$\boldsymbol{x_{1,B}} = \left(\boldsymbol{l_1}, \dots, \boldsymbol{l_{\lfloor z/2 \rfloor}}, \boldsymbol{u_{\lfloor z/2 \rfloor+1}}, \dots, \boldsymbol{u_z}\right)^{\boldsymbol{T}} \tag{21c}$$

$$\boldsymbol{x_{A,1}} = \left(\boldsymbol{u_1}, \dots, \boldsymbol{u_{\lfloor z/2 \rfloor}}, \boldsymbol{l_{\lfloor z/2 \rfloor+1}}, \dots, \boldsymbol{l_z}\right)^{\boldsymbol{T}} \tag{21d}$$

In Eq. (21a) to (21d), $(\boldsymbol{l_1}, \dots, \boldsymbol{l_z})$ and $(\boldsymbol{u_1}, \dots, \boldsymbol{u_z})$ are the lower and upper limits of network factors $\boldsymbol{x_1}, \dots, \boldsymbol{x_z}$, respectively.

Step 2.2: Initialization of the cells along the four boundaries: The values of the cells at the four boundaries are initialized as follows Eq. (22a) to Eq. (22d):

$$\boldsymbol{x_{1,j}} = \frac{x_{1,B}-x_{1,1}}{B-1}(\boldsymbol{j}-\boldsymbol{1}) + \boldsymbol{x_{1,1}} = \frac{j-1}{B-1}\boldsymbol{x_{1,B}} + \frac{B-j}{B-1}\boldsymbol{x_{1,1}}, \boldsymbol{j} = \boldsymbol{2}, \dots, \boldsymbol{B}-\boldsymbol{1} \tag{22a}$$

$$\boldsymbol{x_{A,j}} = \frac{x_{A,B}-x_{A,1}}{B-1}(\boldsymbol{j}-\boldsymbol{1}) + \boldsymbol{x_{A,1}} = \frac{j-1}{B-1}\boldsymbol{x_{A,B}} + \frac{B-j}{B-1}\boldsymbol{x_{A,1}}, \boldsymbol{j} = \boldsymbol{2}, \dots, \boldsymbol{B}-\boldsymbol{1} \tag{22b}$$

$$\boldsymbol{x_{i,1}} = \frac{x_{A,1}-x_{1,1}}{A-1}(\boldsymbol{i}-\boldsymbol{1}) + \boldsymbol{x_{1,1}} = \frac{i-1}{A-1}\boldsymbol{x_{A,1}} + \frac{A-i}{A-1}\boldsymbol{x_{1,1}}, \boldsymbol{i} = \boldsymbol{2}, \dots, \boldsymbol{A}-\boldsymbol{1} \tag{22c}$$

$$\boldsymbol{x_{i,B}} = \frac{x_{A,B}-x_{1,B}}{A-1}(\boldsymbol{i}-\boldsymbol{1}) + \boldsymbol{x_{1,B}} = \frac{i-1}{A-1}\boldsymbol{x_{A,B}} + \frac{B-i}{B-1}\boldsymbol{x_{1,B}}, \boldsymbol{i} = \boldsymbol{2}, \dots, \boldsymbol{A}-\boldsymbol{1} \tag{22d}$$

Step 2.3: Initialization of the remaining cells is performed by initializing the weight vectors of the corners. The remaining cells are initialized sequentially from the top to the bottom and from the left to the right, as detailed below Eq. (23):

For $\boldsymbol{i} = \boldsymbol{2}, \dots, \boldsymbol{X}-\boldsymbol{1}; \boldsymbol{j} = \boldsymbol{2}, \dots, \boldsymbol{Y}-\boldsymbol{1}$,

$$\boldsymbol{x_{i,j}} = \frac{x_{A,j}-x_{i,j}}{A-1}(\boldsymbol{i}-\boldsymbol{1}) + \boldsymbol{x_{1,j}} = \frac{i-1}{A-1}\boldsymbol{x_{A,j}} + \frac{A-i}{A-1}\boldsymbol{x_{1,j}}$$

$$= \frac{i-1}{A-1}\left(\frac{j-1}{B-1}\boldsymbol{x_{A,B}} + \frac{B-j}{B-1}\boldsymbol{x_{A,1}}\right) + \frac{A-i}{A-1}\left(\frac{j-1}{B-1}\boldsymbol{x_{1,B}} + \frac{B-j}{B-1}\boldsymbol{x_{1,1}}\right)$$

$$= \frac{\left((j-1)(i-1)x_{A,B}+(j-1)(A-i)x_{1,B}+(B-j)(i-1)x_{A,1}+(B-j)(A-i)x_{1,1}\right)}{(B-1)(A-1)} \tag{23}$$

Here, for the next generation, a new population with optimal positions $\boldsymbol{x_{i,j}}$ is obtained by the SFO, which is discussed in the Section 3.3.2. This ensures the better exploitation and efficiency of DSFOA.

Step 3: The fitness function $\boldsymbol{f}\left(\boldsymbol{x_j^e}\right)$ for all doves, where $\boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{N}$ is calculated at epoch $\boldsymbol{e}$ as a total number of crumbs located at the position of the $\boldsymbol{d^{th}}$ dove.

Step 4: The dove $\mathbf{d_j^e}$ positioned nearest to the highest quantity of crumbs by applying the maximum criterion at $\boldsymbol{e}$ as follows Eq. (24):

$$\boldsymbol{d_j^e} = \mathbf{argmax}\left\{\boldsymbol{f}\left(\boldsymbol{x_j^e}\right)\right\}, \boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{N} \tag{24}$$

Step 5: Each doves' satiety degree is updated as Eq. (25):

$$S_j^e = \lambda S_j^{e-1} + e^{\left(f(x_j) - f\left(x_{d_f}\right)\right)}, j = 1, \dots, N \quad (25)$$

Step 6: The dove d_s^e exhibiting the highest level of satiety (fitness) is selected based on the maximum degree of satisfaction:

$$d_s^e = \underset{1 \le j \le N}{\operatorname{argmax}} \{S_j^e\}, j = 1, \dots, N \quad (26)$$

The dove selected in accordance with Eq. (26) is the one that exhibits optimal foraging efficiency, making it the model for imitation by other doves within the flock.

Step 7: Each doves' location vector is updated as follows Eq. (27) to Eq. (29):

$$x_j^{e+1} = x_j^e + \eta \beta_j^e \left(x_{d_s}^e - x_j^e\right) \quad (27)$$

$$\text{Where } \beta_j^e = \left(\frac{S_{d_s}^e - S_j^e}{S_{d_s}^e}\right)\left(1 - \frac{\left\|x_j^e - x_{d_s}^e\right\|}{\max\_distance}\right) \quad (28)$$

$$\max\_distance: \max_{1 \le j \le N} \left\|x_j - x_i\right\| \quad (29)$$

Step 8: Go to Step 3 and the number of iterations is increased by one, i.e., $e = e + 1$ until the termination condition is satisfied. The termination condition is defined as follows Eq. (30):

$$e \le maximum\ number\ of\ epochs \quad (30)$$

ii. Sailfish Optimization

The SFO method is modelled after sailfish group hunting, in which members alternate assaults on schooling sardine prey [32]. The fundamental motivation for this algorithm is the attack-change tactic used in sailfish group hunting. This model incorporates two distinct sets of prey and predator populations to simulate group hunting behaviour strategies. It breaks down the collective protection of grouped prey through assault alternation. Prey movements can be modified throughout the search space, allowing hunters to capture the appropriate prey, thereby enhancing their fitness levels. The mathematical model for this algorithm is provided in the following sections.

3.4.2.1. Initialization

The SFO algorithm functions as a metaheuristic that is based on population dynamics. The variables of this method are the sailfish's location in the search space, and it is assumed that the sailfish represent potential suggestions. Consequently, the population (dove's initial locations) is produced at random over the solution space. The sailfish's changing location vectors allow it to hunt in one, two, three, or hyperdimensional space. The current location of the i^{th} member at the k^{th} searching bout in a d-dimensional search space is $x_{i,k} \in \mathcal{R}(i = 1, 2, \dots, z)$. It has been suggested that matrix SF might save the location of every sailfish.

$$SF_{location} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{z,1} & x_{z,2} & \cdots & x_{z,d} \end{bmatrix} \quad (31)$$

In Eq. (31) and Eq. (32), z indicated as amount of sailfish and d represents the number of decision parameters, and $x_{i,j}$ denotes as value of j^{th} dimension of i^{th} sailfish. Also, the fitness of all sailfishes is determined as:

$$fitness\ value\ of\ sail\ fish = f(sailfish) = f(x_1, x_2, \dots, x_z) \quad (32)$$

To evaluate all sailfishes, below matrix represents the fitness value for all solutions.

$$SF_{Fitness} = \begin{bmatrix} f\left(x_{1,1}\ x_{1,2}\ \cdots\ x_{1,d}\right) \\ f\left(x_{2,1}\ x_{2,2}\ \cdots\ x_{2,d}\right) \\ \vdots \\ f\left(x_{z,1}\ x_{z,2}\ \cdots\ x_{z,d}\right) \end{bmatrix} = \begin{bmatrix} F_{x_1} \\ F_{x_2} \\ \vdots \\ F_{x_z} \end{bmatrix} \quad (33)$$

In Eq. (33), f determines the fitness function and $SF_{Fitness}$ stores the fitness value of every sailfish. The other significant incorporator in the SFO system is the school of sardines. The sardine group is thought to be swimming in the search space as well. Thus, sardines' location and fitness values are applied in the following:

$$S_{location} = \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,d} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ S_{n,1} & S_{n,2} & \cdots & S_{n,d} \end{bmatrix} \tag{34}$$

In Eq. (34), $\boldsymbol{n}$ represents the number of sardines and $S_{i,j}$ indicates the value of j^{th} dimension of i^{th} sardine. Also, the fitness of all sardines is determined as:

$$S_{Fitness} = \begin{bmatrix} f(S_{1,1}\ S_{1,2} \cdots S_{1,d}) \\ f(S_{2,1}\ S_{2,2} \cdots S_{2,d}) \\ \vdots \\ f(S_{n,1}\ S_{n,2} \cdots S_{n,d}) \end{bmatrix} = \begin{bmatrix} F_{S_1} \\ F_{S_2} \\ \vdots \\ F_{S_n} \end{bmatrix} \tag{35}$$

In Eq. (35), $\boldsymbol{f}$ is the fitness function, defined in Eq. (15) and $S_{Fitness}$ stores the fitness value of every sardine. Sardines along with sailfish are key factors in determining the best solutions. Sardines work together to locate the ideal location in this region, while sailfish are the primary element dispersed across the search space in this SFO. Sailfish can eat sardines while searching the search space, and they will adjust their location if they find a better solution than what they have found so far.

3.4.2.2 Elitism

When search agents' positions are updated, sometimes good solutions are lost, and unless elitist selection is used, these positions may be weaker than the previous ones. Copying the unaltered fittest answer or solution to the following generation is elitism. The optimal location of the sailfish is recorded in every iteration and is regarded as an elite in the SFO algorithm as well. Since the elite sailfish is the fittest one yet discovered, it might have an impact on the sardines' ability to accelerate and navigate during the attack. Furthermore, as already noted, during group hunting, sardines will sustain injuries from cutting motions with the sailfish's rostrum.

Therefore, the damaged sardine's location is likewise recalled in every iteration, and the sailfish may choose this sardine as the ideal target for cooperative hunting. Elite sailfish along with injured sardines with the greatest fitness at the i^{th} iteration are referred to as $X^i_{elite\_SF}$ and $X^i_{injured\_S}$, respectively. These roles can significantly affect SFO performance and save time spent rediscovering ideas that have already been rejected.

3.4.2.3 Attack-Change Mechanism

In reality, sailfish typically target a school of prey when none of their companions are attacking. In other words, by coordinating their attacks in time, sailfish can increase the likelihood of hunting success. Prey are pursued and herded by sailfish. Without direct communication, sailfish herd by adjusting their positions based on the locations of other hunters within the prey school. The attack-change mechanism of sailfish during group hunting is demonstrated by the SFO approach. At the i^{th} iteration of the SFO algorithm, the sailfish's new position $X^i_{new\_SF}$ changes as follows:

$$X^i_{new\_SF} = X^i_{elite\_SF} - \lambda_i \left(rand(0,1) \times \left(\frac{X^i_{elite\_SF} + X^i_{injured\_S}}{2} \right) - X^i_{old\_SF} \right) \tag{36}$$

In Eq. (36), $X^i_{elite\_SF}$ indicates the location of elite sailfish generated until now, $X^i_{injured\_S}$ measures the optimal location of injured sardine generated so far, $X^i_{old\_SF}$ represents the present position of sailfish, $rand(0,1)$ refers to an arbitrary value from 0 to 1, and λ_i is the coefficient at the i^{th} iteration computed by:

$$\lambda_i = 2 \times rand(0,1) \times PD - PD \tag{37}$$

Prey Density (PD) in Eq. (37), represents as total amount of prey at all iterations. The parameter is critical for adjusting the position of sailfish in relation to the prey school, as the quantity of prey diminishes during group hunting activities by sailfish. It is established as:

$$PD = 1 - \left(\frac{N_{SF}}{N_{SF} + N_S} \right) \tag{38}$$

In Eq. (38), N_{SF} and N_S represents the amount of sailfish and sardines in every iteration, correspondingly. Here, N_{SF} expressed as $N_S \times PP$, where *PP* represents the proportion of the sardine population constituting the initial sailfish population., because the initial number of sardines is often greater than the initial number of sailfish.

The fluctuation range of λ, as determined by Eq. (37), is between -1 and 1, but it is contingent upon the quantity of prey. In other words, the quantity of λ will approach -1 or 1 with regard to $rand(0,1)$ in Eq.

(37) as $\boldsymbol{PD}$ increases. While $\boldsymbol{rand(0,1) > 0.5}$, the $\boldsymbol{\lambda}$ variable will tend to 1; while $\boldsymbol{rand(0,1) < 0.5}$, it will tend to -1, and when $\boldsymbol{rand(0,1) = 0.5}$, it can be 0. Sailfish divergence and convergence near prey schools may be analytically modeled by varying their $\boldsymbol{\lambda}$ and updating their positions. This emphasizes the strong global exploration. Thus, The SFO algorithm enables sailfish to employ two-way navigation for the purpose of updating their location in relation to the prey school.

3.4.2.4 Hunting and Catching Prey

Group hunting often results in sardines' scales being removed by sailfish bills, causing injuries. At the beginning, the ability to catch prey is enhanced in sailfish due to their increased energy levels, allowing sardines to escape quickly. Over time, the power of attacks decreases, reducing energy stores in prey and affecting their ability to detect directional information. Eventually, sardines rapidly become ensnared after breaking away from the shoal.

Each sardine must adjust its location related to the sailfish's optimal posture and attack power at each iteration to replicate this procedure. The new location of the sardine $\boldsymbol{X^i_{new\_S}}$ during the $\boldsymbol{i^{th}}$ iteration of the SFO algorithm is defined by

$$\boldsymbol{X^i_{new\_S} = r \times \left(X^i_{elite\_SF} - X^i_{old_S} + AP\right)} \tag{39}$$

In Eq. (39), $\boldsymbol{X^i_{elite\_SF}}$ represents the optimal location of elite sailfish generated so far, $\boldsymbol{X^i_{old_S}}$ refers to the present location of sardine, $\boldsymbol{r}$ indicates an arbitrary value from 0 to 1, and $\boldsymbol{AP}$ refers to the number of sailfish's attack power at every iteration, which is defined by

$$\boldsymbol{AP = A \times \left(1 - (2 \times Itr \times \varepsilon)\right)} \tag{40}$$

In Eq. (40), the power attack value is reduced linearly from A to 0 with the help of the coefficients $\boldsymbol{A}$ and $\boldsymbol{\varepsilon}$. An essential part of the suggested strategy is Eq. (39), which demonstrates how the sardines move relative to the sailfish after each assault. In a phenomenon that is essentially a strong local search, this equation lets sardines evade the quickest sailfish (elite) by drawing on data from their immediate surroundings. Consequently, this approach permits a reasonable compromise between the search space's exploration and exploitation.

Furthermore, the quantity of sardines that update their locations and the number of displacements rely on the strength of the sailfish attack $(\boldsymbol{AP})$. The potency of a sailfish's assault decreases over time. Decreasing the intensity of sailfish's attack strength may help the search agents converge more adaptively. Based on the $\boldsymbol{AP}$ value, the count of sardines updating their location $(\boldsymbol{\alpha})$ and their quantity of parameters $(\boldsymbol{\beta})$ are determined as:

$$\boldsymbol{\alpha = N_S \times AP} \tag{41}$$

$$\boldsymbol{\beta = d_i \times AP} \tag{42}$$

In Eq. (41) and Eq. (42), $\boldsymbol{d_i}$ refers to the number of parameters at $\boldsymbol{i^{th}}$ iteration. In instances where the strength of the sailfish's tap is recorded as low $(\boldsymbol{AP < 0.5})$, updates are applied exclusively to $\boldsymbol{\alpha}$ sardines that possess $\boldsymbol{\beta}$ sardine variables. All sardine positions will be updated if the sailfish tap strength is high $(\boldsymbol{AP > 0.5})$. By adjusting the $\boldsymbol{AP}$ and $\boldsymbol{r}$ parameters, SFO is possible to produce a more stochastic optimization behavior and avoid iterative stagnation of local optima.

During the last phase of the hunting process, an injured sardine that has detached from the shoal will be swiftly captured. This algorithm posits that the sardine captures prey when its fitness exceeds that of the comparable sailfish. The sailfish's position supersedes the latest location of the pursued sardine, thereby enhancing the probability of successfully capturing fresh prey. This representation is as follows:

$$\boldsymbol{X^i_{SF} = X^i_S,\ if\ f(S_i) < f(SF_i)} \tag{43}$$

In Eq. (43), $\boldsymbol{X^i_S}$ and $\boldsymbol{X^i_{SF}}$ are the present location of sardine and sailfish at $\boldsymbol{i^{th}}$ iteration, respectively. The SFO technique generates the positions of sailfish and sardines randomly. In each round, the location of every sailfish is adjusted in relation to an injured sardine along with exceptional sailfish. Furthermore, chosen sardines and elite sailfish update sardine positions based on the power of the sailfish attack. After updating the sailfish and sardine positions, the objective function evaluates them. If any sardine becomes fitter than any other sailfish, the sailfish adjusts its place to match the associated sardine. Additionally, the positions of elite sailfish along with injured sardines are modified with each algorithm cycle. The targeted sardine will be removed from the population. The stages undergo iterative updates until the end criterion is satisfied.

Algorithm 1: DSFOA for CH Selection

Input: $\boldsymbol{N}$ sensor nodes

Output: Optimal CH node and centroid for each cluster

1. **Begin**
2. Construct the WSN using $\boldsymbol{N}$ nodes and locate all nodes randomly;
3. Split $\boldsymbol{N}$ nodes into the number of clusters $(\boldsymbol{\eta})$;
4. Each cluster has $\boldsymbol{\eta_c}$ nodes and each node is associated with its closest CH;
5. Choose the initial CH $(\boldsymbol{C_\eta})$ by choosing the group of medians using Eq. (18);
6. Calculate the distance $\boldsymbol{d(C_\eta, \gamma)}$ by the CH using Eq. (19);
7. Apply the DSFOA to choose the new CH and centroid of the clustered nodes;
8. Initialize DOA parameters;

   Dove population size $\boldsymbol{N}$

   Maximum number of epochs $\boldsymbol{e_{max\_DOA}}$

   Satiety level $(\boldsymbol{S_j})$ for each dove $\boldsymbol{j}$
9. $\boldsymbol{for(each\ dove\ population)}$
10. $\boldsymbol{for(each\ epoch\ e_{DOA} = 1{:}\, e_{max\_DOA})}$
11. Evaluate fitness function for each dove using Eq. (15);
12. Find the dove with the maximum fitness (nearest to the best solution) using Eq. (24);
13. Update satiety degree for each dove using Eq. (25);
14. Select the dove with maximum satiety using Eq. (26);
15. Update each dove's location vector using Eq. (27);
16. $\boldsymbol{end\ for}$
17. Use the best solution from DOA as the initial population for sailfish and sardines;
18. Initialize SFO parameters;

    Sailfish population size $\boldsymbol{N_s}$, and sardine population size $\boldsymbol{N_d}$

    Set parameters $\boldsymbol{A = 4}$, $\boldsymbol{\varepsilon = 0.001}$

    Maximum number of iterations $\boldsymbol{T_{max\_SFO}}$;
19. $\boldsymbol{for(each\ iteration\ T_{SFO} = 1{:}\, T_{max\_SFO})}$
20. Evaluate fitness for all sailfish and sardines using Eq. (15);
21. Find the elite sailfish $(\boldsymbol{X_{elite}})$ and injured sardine $(\boldsymbol{X_{injured}})$ with the best fitness;
22. Determine $\boldsymbol{\lambda_i}$ using Eq. (37);
23. Update sailfish positions using Eq. (36);
24. Determine $\boldsymbol{AP}$ using Eq. (40);
25. $\boldsymbol{if(AP < 0.5)}$
26. Determine $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ using Eqns. (41) and (42), respectively;
27. Choose a set of sardines based on the value of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$;
28. Update the position of chosen sardine using Eq. (39);
29. $\boldsymbol{else}$
30. Update the position of all sardines using Eq. (39);
31. $\boldsymbol{end\ if}$
32. Evaluate fitness for each sardine using Eq. (21);

```
    if(there is a better solution in sardine population)
      Replace a sailfish with an injured sardine using Eq. (43);
      Eliminate the hunted sardine from population;
      Update the elite sailfish and injured sardine;
    end if
  end for
end for
Return the global best solution i.e., optimal CH and centroid for each cluster;
End
```

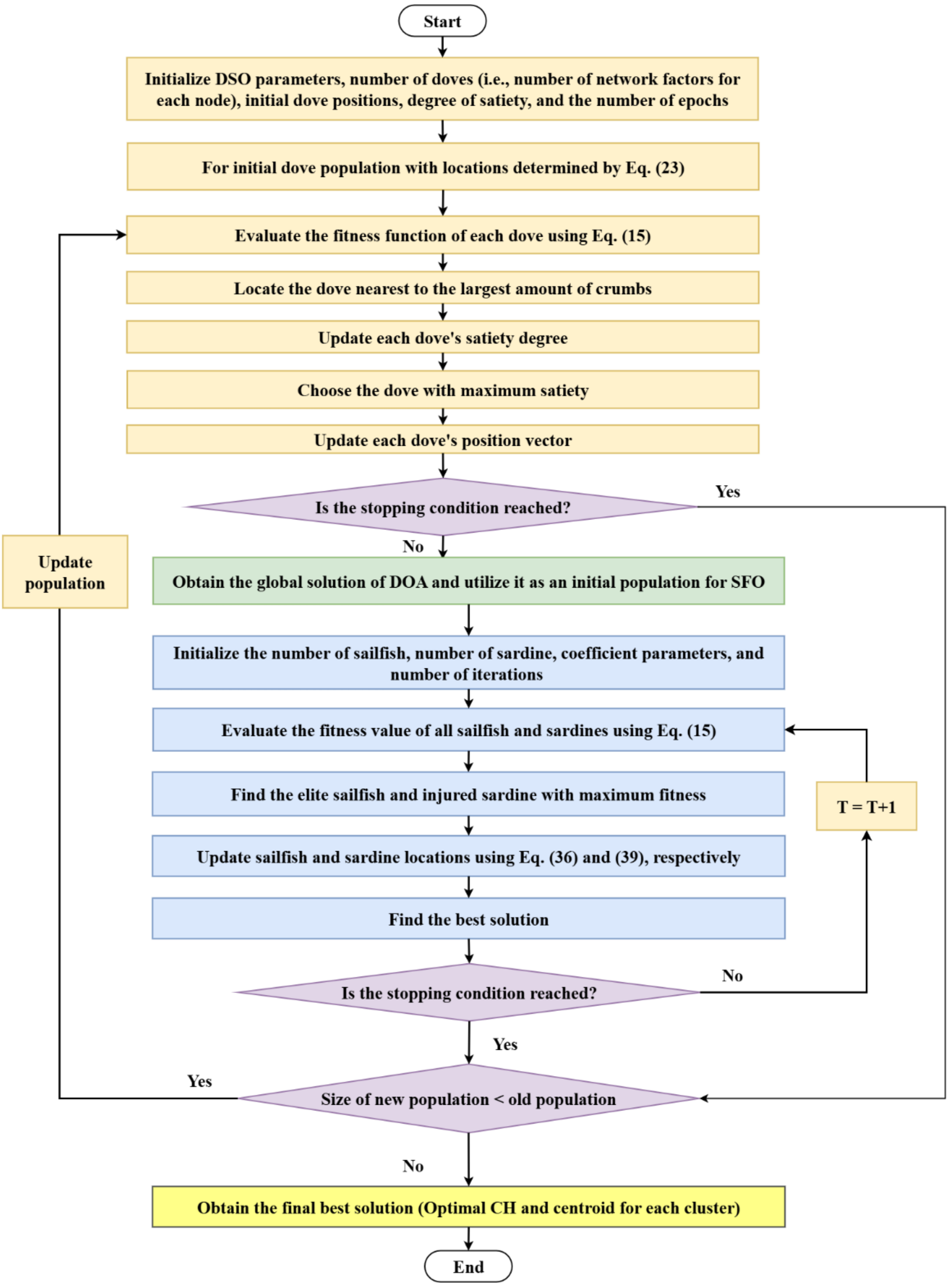

Figure 3. Flowchart of DSFOA for Optimal CH Selection

Thus, this FMNS-K-medoids with DSFOA technique is used to construct clusters and select optimal CHs due to its ability to discover accurate centroids, which results in reduced power consumption, low packet delay, and superior sensor nodes.

iii. Data Transmission

Once the CH for each cluster is found using the DSFOA, the tree-based multi-hop routing is used to reliably transfer information from the CMs to the BS. This strategy requires the arrangement of nodes in a hierarchical tree structure, placing the BS node at the root with the CMs extending from it. The algorithm for this tree-based multi-hop routing is given below.

Algorithm 2: Data Transmission Using Tree-Based Multi-hop Routing

1. **Begin**
2. Allocate a distinct ID to all nodes in the network;
3. Designate the BS as a root node;
4. Utilize the FMNS-K-medoids with DSFOA to group nodes into clusters and choose the best node with the maximum fitness value as CHs;
5. Transmit the data from the CH nearest to the BS (called direct CH) directly;
6. Determine the Euclidean distance between the direct CH and other CHs;
7. Designate the direct CH as a parent node to the closest CHs;
8. For each cluster, select the CH that is nearest to as its direct CH in the tree;
9. Each node transmits the data to its parent node, which forwards it to the BS;
10. Transmit the data to the BS via the tree structure, with each node transmitting the data to its parent node until it reaches the root node;
11. **End**

4. SIMULATION OUTCOMES

This phase evaluates the effectiveness of the suggested FKADSO-EECRA in WSNs, and the results are compared with existing algorithms, including HHOCFR [23], SFO-SHO [25], MACOA [28], HTCCR [29], and FOAEAUC-SARP [30]. A system configuration consisting of an i7 processor, 8GB RAM, 1TB HDD, and Windows 10 64-bit OS is used to simulate the proposed and current algorithms in Python 3.12.4. The simulation can depend on data transfer, node availability, and sensing range. Table 2 provides the simulation parameters.

Table 2. Parameters Settings

Parameters	Values
Simulation area	500×500 m^2
No. of sensor nodes	500
No. of rounds	2000
No. of clusters	8
Initial energy	0.5 J
Transceiver energy, E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
Data aggregation energy	5 nJ/bit
BS coordinates	510, 510
Packet transfer rate	1 packet/sec
Packet size	512 bytes

4.1 Performance Evaluation Metrics

Performance metrics used for evaluation are energy consumption, network lifespan, E2D, PDR, throughput, PLR, jitter, and routing overhead.

- **Energy Consumption:** The total energy utilized by all sensor nodes during each round is specified.
- **Network Lifetime:** It is the total amount of rounds or amount of time the network takes to complete the data transmission.
- **E2D:** The term denotes the duration required for data transmission from the source node to the sink node.
- **Jitter:** It refers to the variance on E2D observed by receiving nodes.
- **PDR:** The metric refers to the effective delivery rates of packets transmitted over the network channel from the source node to the sink node. The calculation is performed as follows Eq. (44):

$$\boldsymbol{PDR} = \frac{\boldsymbol{Total\ number\ of\ packets\ received\ succesfully}}{\boldsymbol{Total\ number\ of\ packets\ sent}} \times \mathbf{100} \tag{44}$$

- **Throughput:** The data volume transmitted across the network at maximum speed Eq. (45).

$$\boldsymbol{Throughput} = \frac{\boldsymbol{Amount\ of\ packets\ transmitted\ (bits)}}{\boldsymbol{Time\ taken\ to\ transmit\ the\ packet\ (s)}} \tag{45}$$

- **PLR:** The metric is defined as the quantity of packets that are lost during the transmission process between the source and sink nodes.
- **Routing Overhead:** It represents the quantity of control packets sent for path discovery and maintenance.

4.2 Performance Analysis for Different Node Densities & Simulation Rounds

4.2.1 PDR

This section evaluates the performance of FKADSO-EECRA for varying node counts, such as 100, 200, 300, 400, and 500, with a fixed simulation round of 1000.

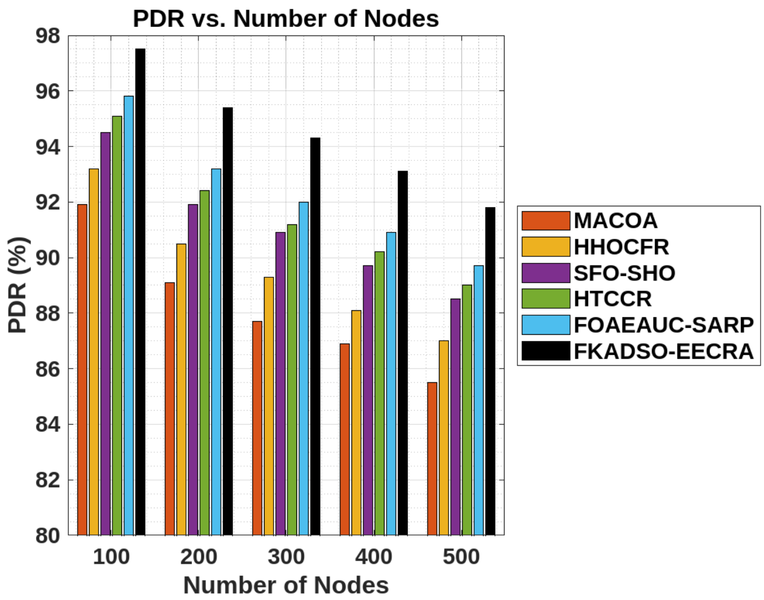

Figure 4. PDR vs. Node Densities

Figure 4 displays the PDR for several cluster-based routing techniques with varying node counts. In contrast to the other algorithms, the FKADSO-EECRA increases the PDR by transmitting the data through the most effective CH nodes and energy-efficient paths. For 500 nodes, it increases the PDR by 7.37%, 5.52%, 3.73%, 3.15%, and 2.34% in comparison to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, respectively. As a result, it is evident that when the number of nodes is changed, the FKADSO-EECRA raises the PDR.

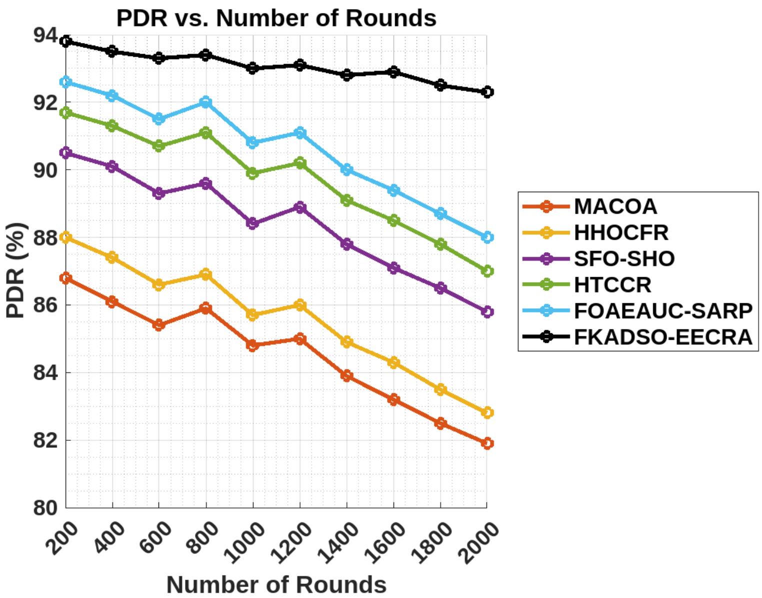

Figure 5. PDR vs. Rounds

This section evaluates the performance of FKADSO-EECRA for varying simulation rounds, between 200 and 2000 for 500 nodes. The PDR for a number of cluster-based routing strategies with different rounds is shown in Figure 5. By using the most efficient CH nodes and energy-efficient paths to transfer the data, the FKADSO-EECRA algorithm raises the PDR effectively. In comparison to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, it raises the PDR by 12.7%, 11.47%, 7.58%, 6.09%, and 4.89% for 2000 rounds, respectively.

4.2.2 Throughput

The performance of cluster-based routing methods with different node counts is displayed in Figure 6. By choosing suitable CHs and pathways according to different node characteristics, the suggested FKADSO-EECRA improves throughput over earlier methods. Regarding 500 nodes, it enhances throughput by 7.37%, 5.52%, 3.73%, 3.15%, and 2.34% in comparison to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, respectively. To conclude, FKADSO-EECRA's throughput grows in direct proportion to the number of nodes.

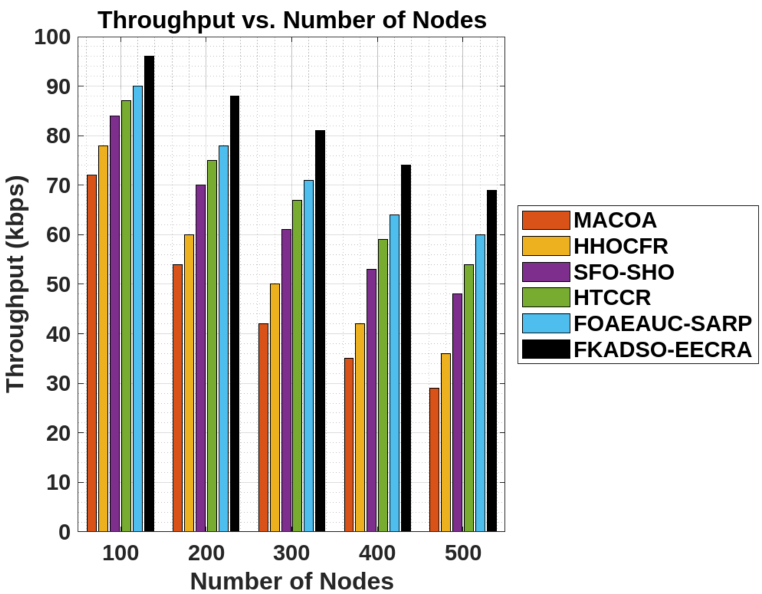

Figure 6. Throughput vs. Node Densities

Figure 7 shows how well cluster-based routing techniques work over many simulation cycles. Throughput is improved over previous techniques by the FKADSO-EECRA, which selects appropriate CHs and pathways based on various node characteristics. Compared to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, it improves throughput by 52.17%, 48.94%, 27.27%, 19.66%, and 14.75% for 2000 rounds, respectively. Therefore, it can be shown that the throughput of FKADSO-EECRA grows as the number of rounds increases.

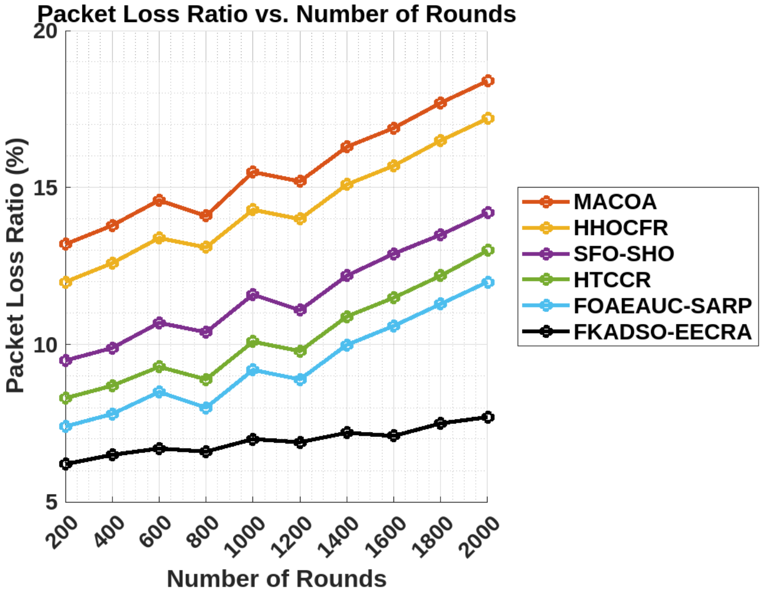

Figure 7. PLR vs. Rounds

4.2.3 Packet Loss Rate

The PLR for several cluster-based routing methods with different node counts is shown in Figure 8. By selecting the best CHs and path for dependable data transfer, the FKADSO-EECRA lowers the PLR compared to other algorithms. The PLR is reduced by 43.45%, 36.92%, 28.7%, 25.45%, and 20.39% for 500 nodes in comparison to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, respectively. Therefore, it can be shown that the PLR of FKADSO-EECRA decreases as the number of nodes rises.

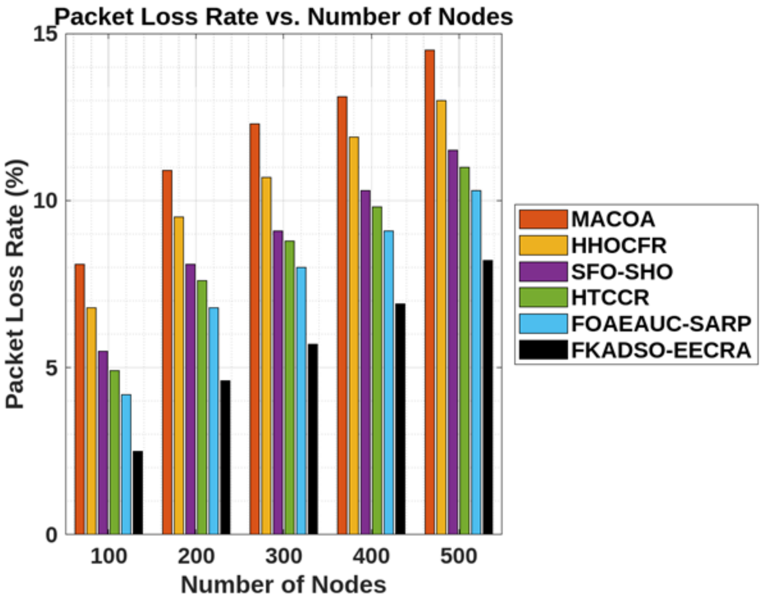

Figure 8. PLR vs. Node Densities

Figure 9 displays the PLR for a number of cluster-based routing techniques with various rounds. In comparison to previous algorithms, the FKADSO-EECRA reduces the PLR by choosing the optimal CHs and route for reliable data transmission. Compared to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, the PLR is decreased by 58.15%, 55.23%, 45.77%, 40.77%, and 35.83% for 2000 rounds, respectively. Consequently, it can be demonstrated that, with an increasing number of rounds, the PLR of FKADSO-EECRA falls.

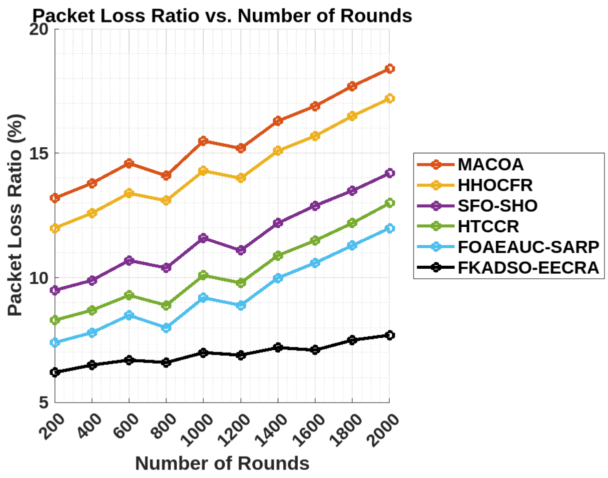

Figure 9. PLR vs. Rounds

4.2.4 Energy Consumption

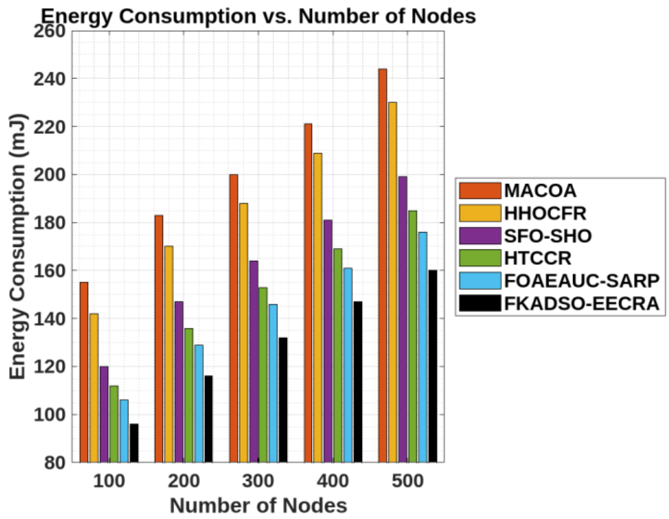

Figure 10. Energy Consumption vs. Node Densities

The energy usage of several cluster-based routing techniques with varying node density is shown in Figure 10. For various nodes, the FKADSO-EECRA uses less energy than other algorithms. Compared to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, the energy consumption for 500 nodes is 34.43%, 30.43%, 19.6%, 13.51%, and 9.09% lower, respectively. Thus, increasing the number of nodes effectively lowers energy usage.

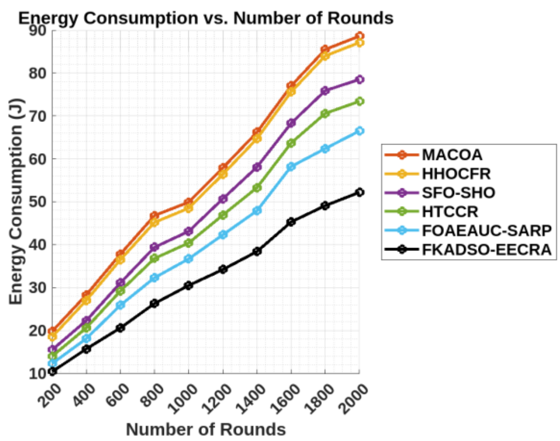

Figure 11. Energy Consumption vs. Rounds

Figure 11 displays the PLR for a number of cluster-based routing techniques with various rounds. In comparison to previous algorithms, the FKADSO-EECRA reduces the PLR by choosing the optimal CHs and route for reliable data transmission. Compared to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, the PLR is decreased by 58.15%, 55.23%, 45.77%, 40.77%, and 35.83% for 2000 rounds, respectively. Consequently, it can be demonstrated that, with an increasing number of rounds, the PLR of FKADSO-EECRA falls.

4.2.5 Network Lifetime

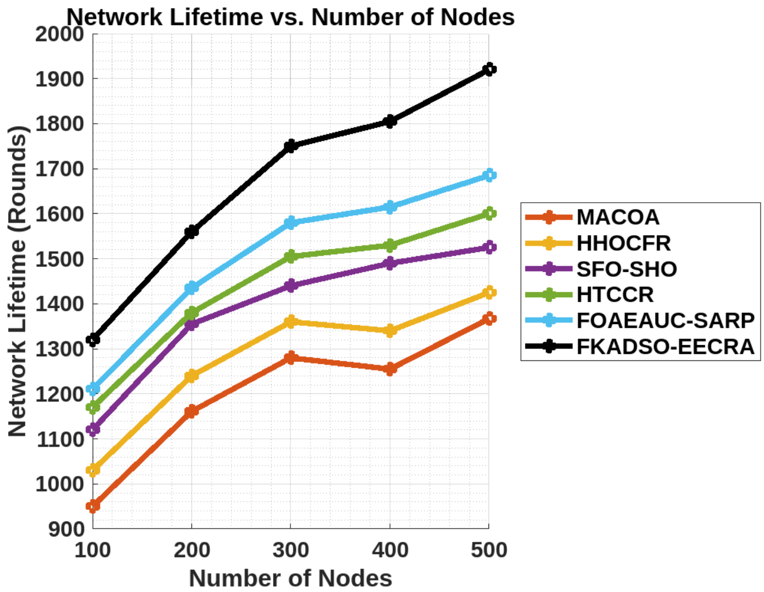

Figure 12. Network Lifetime vs. Node Densities

The network lifespan for several cluster-based routing methods with different node counts is displayed in Figure 12. By lowering the energy consumption of sensor nodes within data transfer, the FKADSO-EECRA outperforms other algorithms in terms of network longevity. For 500 nodes, it increases the network longevity by 40.45%, 34.74%, 25.9%, 20%, and 13.95% in comparison to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP. Therefore, it's obvious the duration of a network increases dramatically as the number of nodes increases.

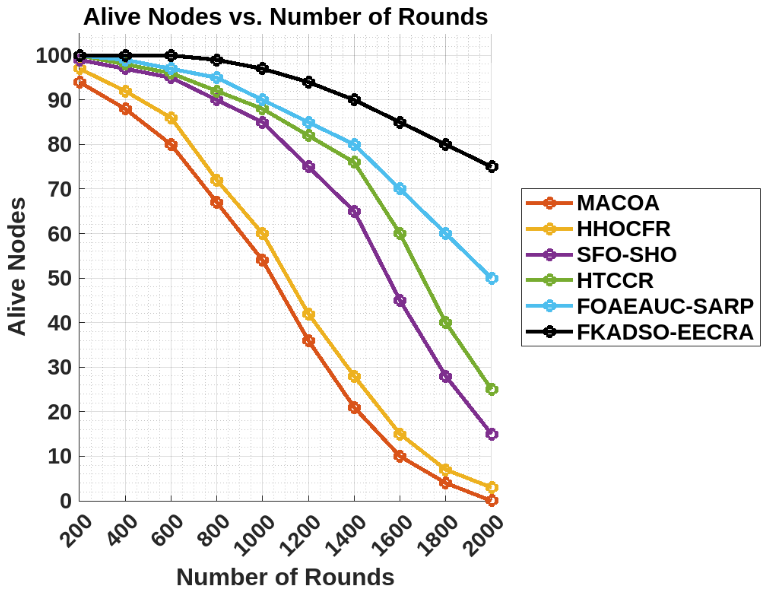

Figure 13. Alive Nodes vs. Rounds

Figure 13 shows the network longevity in terms of live nodes for a number of cluster-based routing techniques with various rounds. The FKADSO-EECRA has more live nodes than the others because it reduces the energy use of nodes during data transfer. In contrast to MACOA, HHOCFR, SFO-SHO, HTCCR, and

FOAEAUC-SARP, which have 0, 3, 15, 25, and 50 living nodes, it contains 75 alive nodes for 2000 rounds, which is more than the others. Consequently, it is clear that as the number of cycles grows, a network's lifespan increases significantly.

4.2.6 Routing Overhead

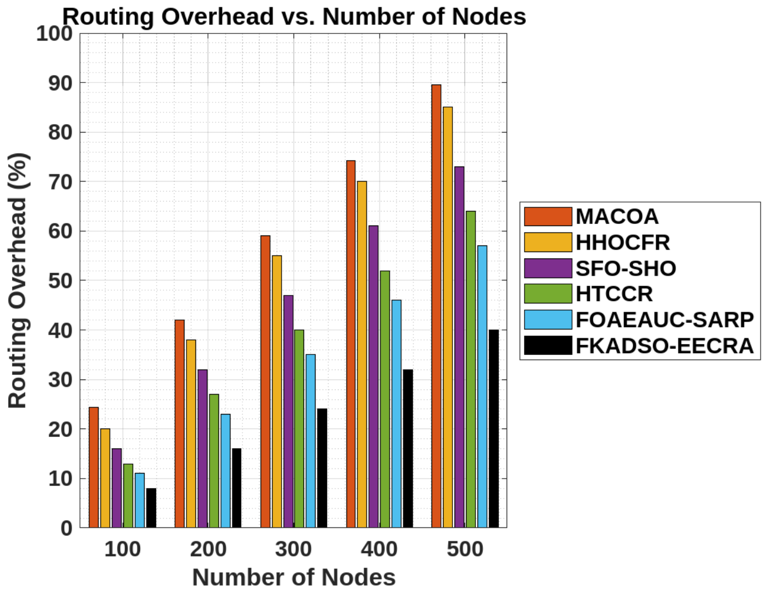

Figure 14. Routing Overhead vs. Node Densities

The routing overhead outcomes for several cluster-based routing algorithms with different node counts are presented in Figure 14. Compared to existing algorithms, the FKADSO-EECRA reduces the routing overhead, meaning that more network resources may be saved for data transfer. In comparison to MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, the routing overhead of FKADSO-EECRA for 500 nodes is decreased by 55.31%, 52.94%, 45.21%, 37.5%, and 29.82%. As a result, it is shown that when the number of nodes is greatly increased, the routing overhead has been reduced.

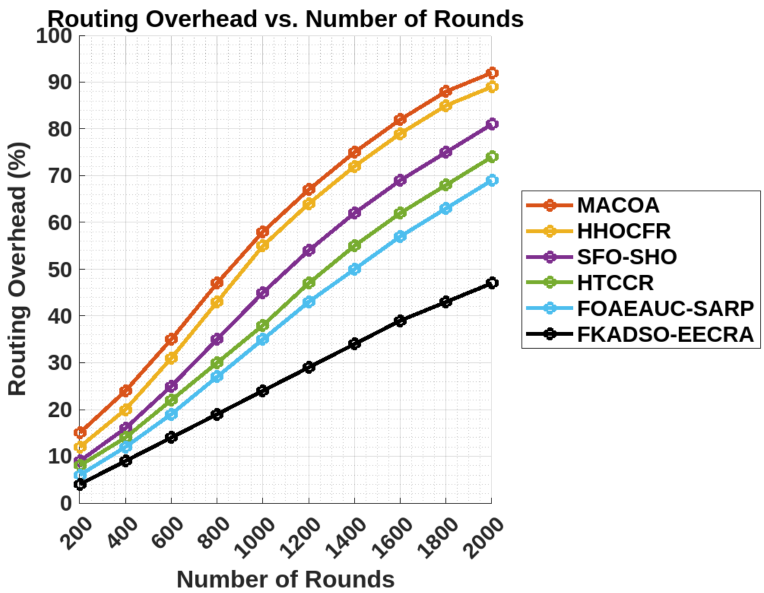

Figure 15. Routing Overhead vs. Rounds

Figure 15 displays the routing overhead results for a number of cluster-based routing methods with various rounds. The FKADSO-EECRA considerably lowers the routing overhead as compared to current methods. Routing overhead is reduced by 48.91%, 47.19%, 41.98%, 36.49%, and 31.88% for 2000 rounds compared to MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP. Consequently, it is demonstrated that the routing overhead decreases when the number of simulation rounds is significantly increased.

4.2.7 E2D

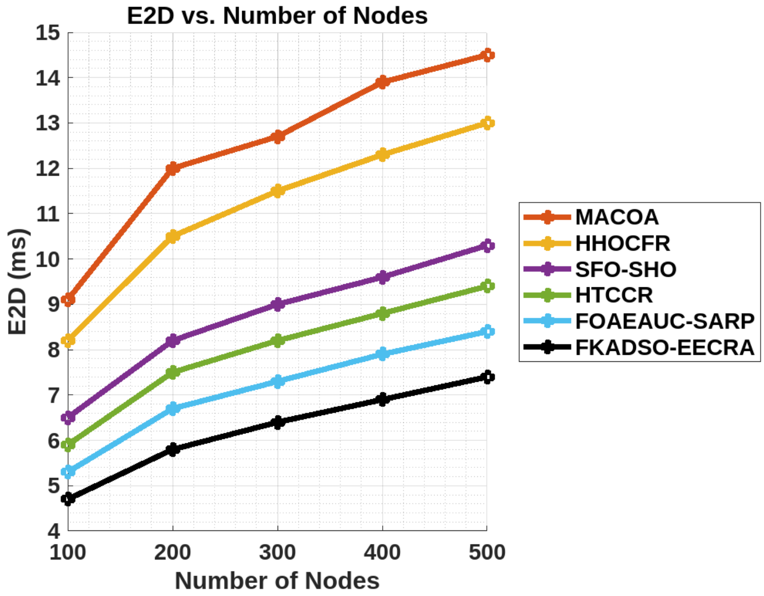

Figure 16. E2D vs. Node Densities

The E2D for a number of cluster-based routing methods with varying node counts is shown in Figure 16. By employing the delay factor as one of the input factors to optimize CHs and routes, the proposed FKADSO-EECRA reduces E2D in comparison to other methods. In comparison to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, the E2D is 48.97%, 43.08%, 28.16%, 21.28%, and 11.9% lower for 500 nodes, respectively. As a result, it is observed that an increase the number of nodes occurs, the delay of data packet transfer between the source and the BS decreases.

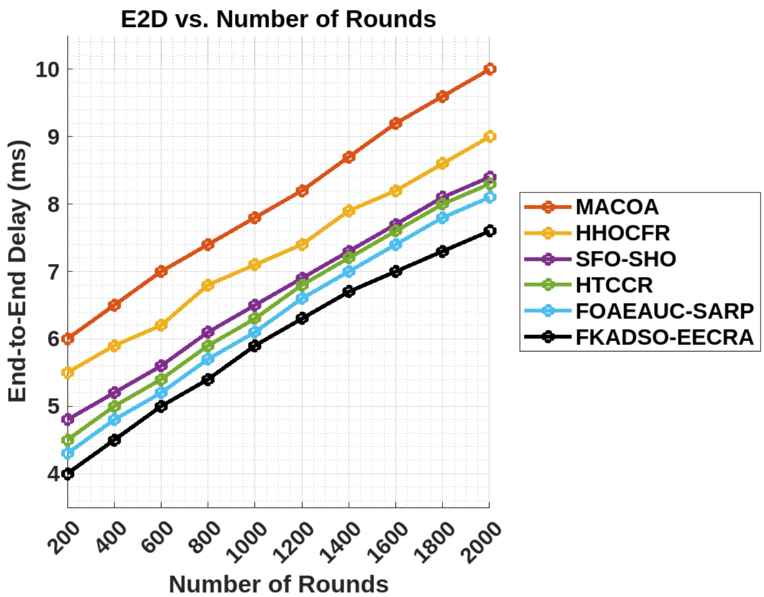

Figure 17. E2D vs. Rounds

Figure 17 displays the E2D for many cluster-based routing techniques with different rounds. The suggested FKADSO-EECRA lowers E2D compared to alternative approaches by using the delay factor as one of the input elements to optimize CHs and routes. For 2000 rounds, the E2D is 24%, 15.56%, 9.52%, 8.43%, and 6.17% lower than the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, respectively.

4.2.7 Jitter

Figure 18 displays the jitter for many cluster-based routing techniques with different node counts. When compared to alternative techniques, it is evident that the FKADSO-EECRA lowers jitter. Compared to the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, the jitter is 55.56%, 46.67%, 33.33%, 27.27%, and 20% lower for 500 nodes, respectively.

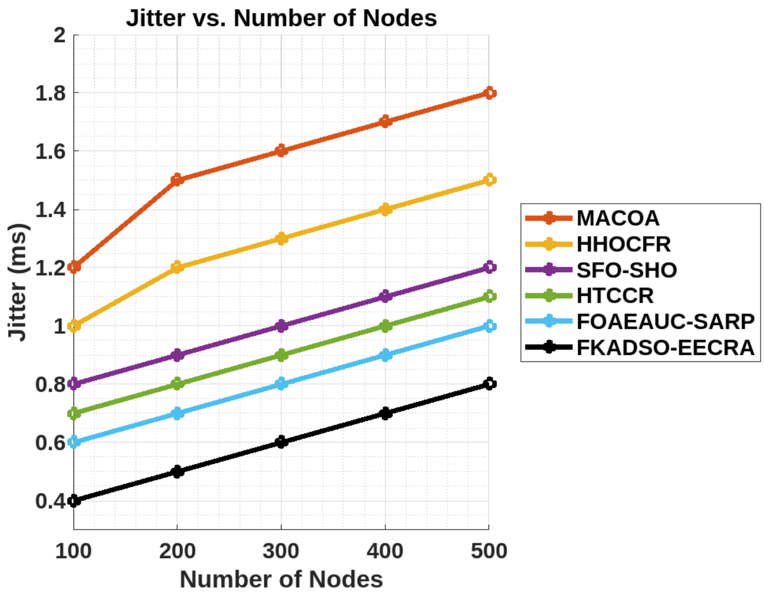

Figure 18. Jitter vs. Node Densities

Jitter for several cluster-based routing strategies with varying node counts is also shown in Figure 19. The FKADSO-EECRA reduces jitter when compared to other methods. For 2000 rounds, the jitter is 57.14%, 53.85%, 31.82%, 23.08%, and 16.67% lower than the MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP, respectively.

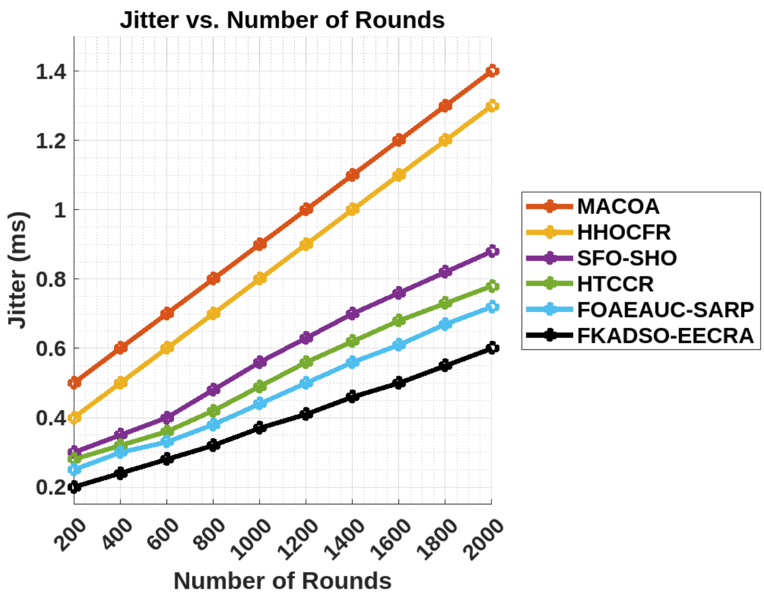

Figure 19. Jitter vs. Rounds

According to these findings, the proposed FKADSO-EECRA outperforms previous algorithms by optimizing CH selection based on various node factors, whereas previous algorithms only considered node residual energy and distance. Furthermore, instead of utilizing a single metaheuristic method, a hybrid DSFOA in a parallel computing platform greatly improves the convergence efficiency of identifying the optimum CH. Thus, the FKADSO-EECRA may be used in WSNs to achieve great energy efficiency and network lifetime while reducing latency and overhead.

5. CONCLUSION

The novel FKADSO-EECRA has been introduced in this work to maximize CH selection and attain energy efficiency in WSNs. First, the FMNS-based K-Medoids technique was used to cluster the nodes. To choose the best CHs based on the many node characteristics and attain high convergence efficiency, the DSFOA was then used. Additionally, the shortest path for energy-efficient data transfer was determined using the tree-based multi-hop routing strategy. Extensive simulations proved that the FKADSO-EECRA achieved a 91.8% PDR, 69 kbps throughput, 8.2% PLR, 160 mJ energy consumption, 1920 rounds of network lifespan, 7.4 ms

E2D, 0.8 ms jitter, and 40% routing overhead for 500 nodes in 1000 rounds. Similarly, it obtained a 92.3% PDR, 140 kbps throughput, 7.7% PLR, 52.2 J energy usage, 75 living nodes, 7.6 ms E2D, 0.6 ms jitter, and 47% routing overhead for 500 nodes in 2000 rounds, outperforming algorithms such as MACOA, HHOCFR, SFO-SHO, HTCCR, and FOAEAUC-SARP.

Mrs.K.Tamilarasi serves as the Assistant Professor in the Department of Artificial Intelligence and Data Science at Hindusthan College of Technology at Salem. She holds M.C.A., M.Phil. degree, all completed at Bharathiyar University with overall 11 years of teaching experience. She has contributed significantly to academia since 2009. Her leadership continues to foster growth and innovation within the department.

Dr.M.Hemalatha currently working as a Professor in Hindusthan College of Arts & Science in Department of Information Technology and has received the BSc(Mathematics) from the Manonmaniam Sundharanar University, MCA from the Madurai Kamaraj University, M.Phil from the Mother Teresa Women's University and Ph.D from Periyar University. She has published many papers in International Journals/UGC Journals and International conferences/Seminars and published many books. She received “Best Senior Faculty Award” from Scientific International Publishing House and “Distinguished Professor Award” from Alpha International Publication.

References:

1. Kumari, S., & Tyagi, A. K. (2024). Wireless sensor networks: an introduction. *Digital Twin and Blockchain for Smart Cities*, 495–528. https://doi.org/10.1002/9781394303564.ch21.
2. Trigka, M., & Dritsas, E. (2025). Wireless sensor networks: from fundamentals and applications to innovations and future trends. *IEEE Access*, 13, 96365–96399. https://doi.org/10.1109/ACCESS.2025.3572328
3. Daousis, S., Peladarinos, N., Cheimaras, V., Papageorgas, P., Piromalis, D. D., & Munteanu, R. A. (2024). Overview of protocols and standards for wireless sensor networks in critical infrastructures. *Future Internet*, 16(1), 33. https://doi.org/10.3390/fi16010033.
4. Kocherla, R., Chandra Sekhar, M., & Vatambeti, R. (2023). Enhancing the energy efficiency for prolonging the network life time in multi-conditional multi-sensor based wireless sensor network. *Journal of Control and Decision*, 10(1), 72–81. https://doi.org/10.1080/23307706.2022.2057362.
5. Prasad, V. K., & Periyasamy, S. (2023). Energy optimization-based clustering protocols in wireless sensor networks and internet of things: survey. *International Journal of Distributed Sensor Networks*, 2023(1), 1362417. https://doi.org/10.1155/2023/1362417.
6. Lonkar, B., Kuthe, A., Charde, P., Dehankar, A., & Kolte, R. (2025). Optimal hybrid energy-saving cluster head selection for wireless sensor networks: an empirical study. *Peer-to-Peer Networking and Applications*, 18(4), 182. https://doi.org/10.1007/s12083-025-02002-y.
7. Gaidhani, A. R., & Potgantwar, A. D. (2024). A review of machine learning-based routing protocols for wireless sensor network lifetime. *Engineering Proceedings*, 59(1), 231. https://doi.org/10.3390/engproc590100231. https://doi.org/10.3390/engproc2023059231.
8. Akhtar, F., Ahmed, I., Youssef, A. F., Smaili, I. H., Ahmed, M. M. R., & El-Rifaie, A. M. (2025). Advance zonal rectangular low energy adaptive clustering hierarchy algorithm for optimal routing in wireless sensors network. *PLoS One*, 20(5), e0321938. https://doi.org/10.1371/journal.pone.0321938.
9. Priyadarshi, R. (2024). Exploring machine learning solutions for overcoming challenges in IoT-based wireless sensor network routing: a comprehensive review. *Wireless Networks*, 30(4), 2647–2673. https://doi.org/10.1007/s11276-024-03697-2.
10. Merah, M., Aliouat, Z., Harbi, Y., & Batta, M. S. (2023). Machine learning-based clustering protocols for internet of things networks: an overview. International Journal of Communication Systems, 36(10), e5487. https://doi.org/10.1002/dac.5487.
11. Del-Valle-Soto, C., Rodríguez, A., & Ascencio-Piña, C. R. (2023). A survey of energy-efficient clustering routing protocols for wireless sensor networks based on metaheuristic approaches. *Artificial Intelligence Review*, 56(9), 9699–9770. https://doi.org/10.1007/s10462-023-10402-w.
12. Sen, S., Sahoo, L., & Ghosh, S. L. (2024). Lifetime extension of wireless sensor networks by perceptive selection of cluster head using K-means and Einstein weighted averaging aggregation operator under uncertainty. *Journal of Industrial Intelligence*, 2(1), 54–62. https://doi.org/10.56578/jii020105.
13. Sahoo, L., Sen, S. S., Tiwary, K., Moslem, S., & Senapati, T. (2024). Improvement of wireless sensor network lifetime via intelligent clustering under uncertainty. *IEEE Access*, 12, 25018–25033. https://doi.org/10.1109/ACCESS.2024.3365490.

14. Chen, R. (2025). Optimizing wireless sensor network topology with node load consideration. *Virtual Reality & Intelligent Hardware*, 7(1), 47–61. https://doi.org/10.1016/j.vrih.2024.08.003.
15. Rekha, & Garg, R. (2024). K-LionER: meta-heuristic approach for energy efficient cluster based routing for WSN-assisted IoT networks. *Cluster Computing*, 27(4), 4207–4221. https://doi.org/10.1007/s10586-024-04280-2
16. Modey, P., Abdul-Salaam, G., Freeman, E., Acheampong, P., Brown-Acquaye, W. L., Agbehadji, I. E., & Millham, R. C. (2024). K-means based bee colony optimization for clustering in heterogeneous sensor network. *Sensors*, 24(23), 7603. https://doi.org/10.3390/s24237603.
17. Rahman, A. A., & Hamim, S. I. (2025). Improved k-means clustering and adaptive distance threshold for energy reduction in WSN-IoTs. *Array*, 100436. https://doi.org/10.1016/j.array.2025.100436.
18. Sen, S., Sahoo, L., Tiwary, K., Simic, V., & Senapati, T. (2023). Wireless sensor network lifetime extension via k-medoids and MCDM techniques in uncertain environment. *Applied Sciences*, 13(5), 3196. https://doi.org/10.3390/app13053196.
19. Cherappa, V., Thangarajan, T., Meenakshi Sundaram, S. S., Hajjej, F., Munusamy, A. K., & Shanmugam, R. (2023). Energy-efficient clustering and routing using ASFO and a cross-layer-based expedient routing protocol for wireless sensor networks. *Sensors*, 23(5), 2788. https://doi.org/10.3390/s23052788.
20. Yuebo, L., Haitao, Y., Hongyan, L., & Qingxue, L. (2023). Fuzzy clustering and routing protocol with rules tuned by improved particle swarm optimization for wireless sensor networks. *IEEE Access*, 11, 128784–128800. https://doi.org/ 10.1109/ACCESS.2023.3332914.
21. Wang, C. (2023). A distributed particle-swarm-optimization-based fuzzy clustering protocol for wireless sensor networks. *Sensors*, 23(15), 6699. https://doi.org/10.3390/s23156699.
22. Gopalan, S. H., Takale, D. G., Jayaprakash, B., & Raj, V. P. (2024). An energy efficient routing protocol with fuzzy neural networks in wireless sensor network. *Ain Shams Engineering Journal*, 15(10), 102979. https://doi.org/10.1016/j.asej.2024.102979.
23. Jing, D. (2024). Harris harks optimization based clustering with fuzzy routing for lifetime enhancing in wireless sensor networks. *IEEE Access*, 12, 12149–12163.https://doi.org/ 10.1109/ACCESS.2024.3354276.
24. Pravin, R. A., Murugan, K., Thiripurasundari, C., Christodoss, P. R., Puviarasi, R., & Lathif, S. I. A. (2024). Stochastic cluster head selection model for energy balancing in IoT enabled heterogeneous WSN. *Measurement: Sensors*, 35, 101282. https://doi.org/10.1016/j.measen.2024.101282.
25. Roberts, M. K., Thangavel, J., & Aldawsari, H. (2024). An improved dual-phased meta-heuristic optimization-based framework for energy efficient cluster-based routing in wireless sensor networks. *Alexandria Engineering Journal*, 101, 306–317. https://doi.org/10.1016/j.aej.2024.05.0787.
26. Kumar, N., Singh, K., & Lloret, J. (2024). WAOA: A hybrid whale-ant optimization algorithm for energy-efficient routing in wireless sensor networks. *Computer Networks*, 254, 110845. https://doi.org/10.1016/j.comnet.2024.110845.
27. Alshammri, G. H. (2025). Enhancing wireless sensor network lifespan and efficiency through improved cluster head selection using improved squirrel search algorithm. *Artificial Intelligence Review*, 58(3), 79. https://doi.org/10.1007/s10462-024-11088-4.
28. Tawfeek, M. A., Alrashdi, I., Alruwaili, M., Jamel, L., Elhady, G. F., & Elwahsh, H. (2025). Improving energy efficiency and routing reliability in wireless sensor networks using modified ant colony optimization. *EURASIP Journal on Wireless Communications and Networking*, 2025(1), 22. https://doi.org/10.1186/s13638-025-02449-w.
29. Bhukya, S. N., & Annavarapu, C. S. R. (2025). Hybrid reliable clustering algorithm with heterogeneous traffic routing for wireless sensor networks. *Sensors*, 25(3), 864. https://doi.org/10.3390/s25030864.
30. Vijayaragavan, P., Saravanan, V., Suresh, C., Manikavelan, D., Maheshwari, A., Vijayalakshmi, K., & Narayanamoorthi, R. (2025). FOAEAUC-SARP: A novel energy-efficient protocol integrating unequal clustering and intelligent routing for sustainable wireless sensor networks. *Results in Engineering*, 25, 103806. https://doi.org/10.1016/j.rineng.2025.103806
31. Su, M. C., Chen, J. H., Utami, A. M., Lin, S. C., & Wei, H. H. (2022). Dove swarm optimization algorithm. *IEEE Access*, 10, 46690–46696. https://doi.org/ 10.1109/ACCESS.2022.3170112.
32. Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 80, 20–34. https://doi.org/10.1016/j.engappai.2019.01.001.