Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Hybrid GA-DL Framework for Energy-Efficient Clustering and Lifespan Extension in RWSNs

Yen-Wu Ti
Xiamen Ocean Vocational College
Rei-Heng Cheng
Xiamen Ocean Vocational College
Songlin Wei
weisonglin@xmoc.edu.cn
Xiamen Ocean Vocational College
Wenfeng Huang
Xiamen Ocean Vocational College
Chih-Min Yu
Chung Yuan Christian University

Article

Keywords: Clustering, Genetic Algorithm, Deep Learning

Posted Date: July 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-7317318/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on October 29th, 2026.
See the published version at https://doi.org/10.1038/s41598-026-77479-5.

A Hybrid GA-DL Framework for Energy-Efficient Clustering and Lifespan Extension in RWSNs

Yen-Wu Ti[1], Rei-Heng Cheng[1], Songlin Wei[1,*], Wenfeng Huang[2], and Chih-Min Yu[3]

[1]School of Information Engineering, Xiamen Ocean Vocational College, Xiamen, 361100, China
[2]School of Ocean Engineering, Xiamen Ocean Vocational College, Xiamen, 361100, China
[3]Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan, 320314, Taiwan
[*]weisonglin@xmoc.edu.cn

ABSTRACT

The limited energy capacity of sensor nodes remains a central constraint on the operational longevity of Wireless Sensor Networks (WSNs)—a challenge that is more critical in Rechargeable Wireless Sensor Networks (RWSNs), where sustaining long-term functionality depends on delaying the First Node Death (FND), a key point that often signals the beginning of network problems. While prior research has extensively explored heuristic algorithms coupled with energy-aware routing mechanisms to reduce transmission energy consumption and extend network lifetime, these methods typically require repeated iterations to converge. Such computational demands render them unsuitable for real-time applications. To overcome these limitations, we introduce a hybrid optimization framework that integrates Genetic Algorithms (GA) with Deep Learning (DL). This framework can run quickly to improve energy efficiency and extend network lifetime in RWSNs. The proposed approach builds upon an energy-aware clustering architecture that incorporates a lightweight version of the Residual Neural Network (ResNet) model. This clustering strategy promotes a more balanced distribution of energy consumption across the network, delaying early node failures caused by localized energy depletion. Based on this setup, we create an optimization scheme using a novel version of GA that helps the ResNet model—simplified for use on regular CPUs—learns an optimal CH selection policy. The GA component employs a custom fitness function designed to maximize the time to FND latency. The synergy between the clustering method and the GA-DL hybrid selection mechanism enables rapid and effective CH assignment without the iterative overhead typical of heuristic algorithms. Our experimental results demonstrate that the proposed scheme not only reduces energy consumption and extends the time to FND but also achieves near real-time performance. This makes it a promising alternative to traditional heuristic-based approaches, particularly in latency-sensitive RWSN applications. Together, these contributions mark a significant step toward the development of scalable, efficient, and resilient RWSN architectures.

keywords: Clustering, Genetic Algorithm, Deep Learning

Introduction

Wireless Sensor Networks (WSNs) have become foundational infrastructure across a broad range of applications, including environmental monitoring, industrial automation, smart healthcare, and military reconnaissance[1,2]. Typically composed of spatially distributed sensor nodes with limited computing power, memory, and energy reserves, these networks face an inherent challenge: energy constraints. Such limitations directly impact the operational lifespan and overall service quality of WSNs[3]. Rechargeable Wireless Sensor Networks (RWSNs) have emerged as a promising solution that enables sustainable operation through charging technologies.[4] However, practical deployment of RWSNs introduces new challenges, especially in efficiently managing energy and maximizing the effective life of the network. Delaying the time to First Node Death (FND) is important because it often means that coverage is less reliable and monitoring is less accurate. Therefore, once an FND occurs, the operator may need to dispatch personnel to recharge. This means that delaying the time to FND can effectively save costs.

Clustering has been shown to be an effective way to manage energy in WSNs[5]. By grouping nodes into clusters and appointing a Cluster Head (CH) to aggregate and forward data to the sink, clustering can significantly save energy by reducing the need for long-distance transmissions. However, most existing clustering strategies oversimplify the problem and fail to account for minor variations in energy use between different nodes. These simplifications may lead to poor clustering results and CH selections, ultimately limiting the ability to balance energy loads and extend the network's lifespan.

To address these limitations, heuristic methods such as Genetic Algorithms (GA) have been widely explored for clustering optimization[6]. However, heuristic methods are inherently iterative processes that begin with an initial guess and gradually refine their solutions through repeated computations. These strategies can work well, but they may require a long time to reach a stable output. This computational overhead poses a significant limitation in many RWSN applications, particularly those that

demand continuous and real-time communication. In such settings, pausing operations to await the convergence of a heuristic algorithm is impractical, making these approaches unsuitable for latency-sensitive or time-critical scenarios.

To solve this problem, we need to develop more efficient computational techniques that can match or even beat the effectiveness of heuristic algorithms while running much faster. To achieve this, we provide a hybrid optimization framework that combines a GA with Deep Learning (DL). This approach not only uses less energy and extends the time to FND, which increases the lifespan of the entire network, but it also provides rapid clustering guidance, making it an excellent alternative for real-time RWSN applications.

However, the performance of any GA-based optimization method is critically dependent on the design of its fitness function. An effective fitness function must capture the trade-offs between competing objectives, such as the number of CHs, transmission distances, and realistic energy consumption models. If functions are not built properly, they may mislead the optimization process and cause undesirable results. We deal with this problem by developing a carefully designed, energy-aware fitness function based on the physical principles of radio transmission energy consumption. We use our improved DL model to learn the GA's output, which helps the RWSN find suitable CHs more quickly during clustering.

DL methods have gained attention in the RWSN domain for tasks like routing optimization and anomaly detection.[7] However, using well-known deep learning models like Residual Neural Network (ResNet), Recurrent Neural Networks (RNNs), and Transformers to select CHs in wireless sensor networks often requires a lot of processing power. These models often demand substantial computational resources, making them less suitable for users with limited computing power.

In this work, we propose a comprehensive approach that integrates precise energy modeling, efficient clustering strategies, and a lightweight ResNet model. This novel approach replaces the previously time-intensive heuristic algorithms, enabling swift execution on standard consumer-grade computers and substantially prolonging the operational lifespan of RWSNs. The main contributions of this study are as follows:

- Energy-Aware Clustering Based on Exact Analytical Models: We begin with a rigorous analytical framework that models the energy consumption characteristics of radio transmissions in RWSNs. This model serves not only as a quantitative tool to evaluate energy usage under varying network conditions, but also as a theoretical foundation for deriving optimal cluster configurations—including the ideal number of clusters.

- GA Fitness Function Informed by Probabilistic and Quantitative Analysis: We introduce a novel fitness function for the GA, rooted in the exact energy model. Importantly, we incorporate probabilistic analysis, assuming uniformly random node deployment, to guide the assignment of weights across evaluation criteria such as CH count and transmission distance. This hybrid design enables the GA to more effectively identify CH sets that can extend the number of rounds in which the time to FND.

- A Lightweight ResNet Model for CH Selection: We provide a framework that enables deep learning models to understand RWSN deployment scenarios, allowing them to address problems that arise in these specific situations. Furthermore, we made the structure of ResNet's residual blocks and the fully connected layers at the end of the model much simpler. The improved architecture reduced both training and execution times by significantly reducing the network depth and number of parameters without affecting decision-making performance.

Based on the above, we propose a hybrid scheme that combines the optimization efficiency of GA with the exact modeling capacity of deep learning. By employing a trained deep learning model, which is faster than heuristic approaches. We can rapidly provide clustering guidance for RWSNs to balance energy consumption and effectively postpone the time to FND.

The remainder of this article is organized as follows. Section 2 reviews related work in the field. Section 3 outlines the RWSN model employed in this study. Section 4 details the GA framework, while Section 5 introduces our proposed hybrid approach that integrates genetic algorithms with deep learning. Section 6 presents the experimental results and compares the performance of our method with existing techniques. Finally, Section 7 concludes the paper and highlights the main findings.

Related works

In recent years, there has been a lot of research on extending the lifespan of WSNs. A significant portion of these results utilize heuristic algorithms to optimize routing.[8] Al-Shalabi et al.[9] address the issue of direct transmission in WSNs and present a new strategy to reduce power consumption and maximize network lifespan. To discover the optimal path, the method employs a GA and a new fitness function. The proposed method is tested and compared with other relevant methods using the MATLAB simulator. Tan et al.[10] propose an enhanced LEACH-M protocol for aerial sensor networks to address communication failures caused by sensor node movement. Based on the existing LEACH and multi-hop communication routing (MCR) protocols, the authors introduce a solution that allows outlier nodes to discover new clusters and return to their original clusters quickly. The protocol is compared with the conventional LEACH-M and MCR protocols.

Nandan et al.[11] improved the use of GAs for CH selection in heterogeneous WSNs tailored for Internet of Things applications. Their approach incorporates a fitness function that evaluates four key parameters: residual node energy, distance to the sink, local node density, and node capability. This design aims to optimize energy efficiency, extend network's lifespan, and mitigate the hotspot problem commonly observed in dense deployments. Rani et al.[12] proposed a dynamic clustering framework that integrates evolutionary algorithms for IoT-oriented WSNs. Their method selects CHs based on residual energy levels and employs a comprehensive fitness function that accounts for inter-cluster distance, distance to the base station, energy consumption, transmission count, and the standard deviation of energy usage. The aim is to make networks more scalable and last longer by making clustering decisions that are more flexible and sensitive to energy usage. Verma et al.[13] developed an improved clustering strategy for WSNs using GAs, prioritizing nodes with the highest residual energy for CH roles. Their multi-objective fitness function balances energy availability, communication distance, network load, and node density. The goal is to reduce energy consumption and maximize the network's lifespan through more efficient CH selection.

Shokouhifar and Hassanzadeh[14] address a GA-based energy-efficient routing protocol for WSNs. The LEACH-LPR protocol considers variables such as residual energy, number of neighborhood nodes, and distance from the sink. Depending on the particular application, the GA's fitness function can be defined. A brief review of hierarchical routing protocols for WSNs is also provided, along with a discussion of the protocol's energy consumption model. Baranidharan and Santhi[15] introduce GAECH, a novel clustering method designed to improve the longevity of WSNs by equitably distributing energy usage across nodes. The method employs a GA algorithm and includes a new fitness function that looks at items such as the amount of energy that is used, how much energy is used by different clusters, dispersion CHs, and how much energy CHs use.

Rahmanian et al.[16] use a GA in the LEACH-C routing protocol to provide a unique method to cluster WSNs and identify. They compare the outcomes using the simulated annealing technique and suggest two new fitness functions. They also provide an overview of WSNs and emphasize the importance of consuming as little power as possible. Furthermore, they explain the LEACH and LEACH-C methods, as well as the issue with LEACH's CH determination. The suggested method selects the best CHs using an evolutionary process, resulting in effective clustering. Mollanejad et al.[17] investigate the application of a dynamic base station relocation (DBSR) method in WSNs to optimize energy consumption and the network's lifespan. They present a genetic way to solve the challenge of where to place the sink. The algorithm looks at energy and distance factors to find the best place for the sink. The authors present the network model and assumptions, detail the proposed method, and provide simulation and performance evaluation results. Darji and Shah[18] discuss energy harvesting in WSNs using machine learning methods, particularly GAs. The goal is to create a machine learning-based routing system that uses energy obtained from the environment rather than from batteries, since traditional routing methods are not appropriate for WSNs with energy-harvesting capabilities. They demonstrate how the suggested GA increases network lifetime and energy efficiency by comparing it with the LEACH protocol. They also explain how GAs operate and give an outline of the LEACH methodology. In the proposed study, fuzzy logic and GAs are applied to the network, and nodes are clustered using GAs.

Sanapala and Duggirala[19] use the improved spider monkey optimization (ISMO) algorithm to construct an optimized energy-efficient routing strategy for WSNs. The LEACH method is used to find the best CHs in the network, and ISMO is used to find the best routing path from the CHs to the sink. ISMO is optimized using a combination of fitness functions, such as residual energy, distance, and traffic routing. The method is evaluated in comparison with existing methods. Panhwar et al.[20] discuss the importance of optimizing energy consumption in WSNs to increase node lifetime. With the deployment of multiple sinks, they improved the fitness function for optimization in WSNs. In addition, they discuss related research on security analysis, clustering, and the detection of malevolent nodes in WSNs. The GA and its operators, including selection, crossover, and mutation, are described. Viet et al.[21] provide an upgrade of the LEACH protocol in WSNs by combining clustering with the K-means method and selecting nodes as CHs with the slime mold algorithm (SMA). They propose an enhanced LEACH approach employing a fitness function and the SMA after providing an outline of the LEACH protocol and its shortcomings. Their simulations demonstrate that SMA-LEACH beats other algorithms, such as PSO-LEACH and BA-LEACH, in terms of energy efficiency.

Edla et al.[22] investigate energy conservation in WSNs and suggest a routing strategy based on particle swarm optimization to improve WSN lifetime. They emphasize the difficulties of routing in WSNs, namely that gateways are limited by energy, computing power, and memory, and offer a novel fitness function that takes into account the number of relay nodes, the distance between the gateway and the sink, and the network's relay load factor. Experiments are used to validate the suggested method, which is then compared with other existing algorithms. Cheng et al.[23] propose a combined weighting model (OCW) and a heuristic approach for LEACH protocol optimization in WSNs. They address issues such as randomization and local optima in the LEACH clustering method. The OCW technique dynamically adjusts the weights of CHs based on their importance in various locations, taking into account characteristics such as residual energy, density, and distance to the sink node. MATLAB simulation experiments are used to validate the effectiveness of the proposed strategy.

Hassan et al.[24] discuss the issue of the enormous load on CHs in WSNs and propose a model to divide the load on CHs into smaller portions and distribute them to the cluster's nodes. Each member node transfers a specific amount of energy

to the CH node using the wireless power transfer strategy. The authors' contributions include demonstrating the efficacy of CH-load reduction in improving energy efficiency, eliminating the centralized nature of the CH load, and employing efficient optimization techniques. Sinde et al.[25] present a solution for extending the lifetime of WSNs through the use of an enhanced LEACH protocol and angle sector-based energy-aware time division multiple access (TDMA) scheduling. By selecting ideal CHs, the authors address the issue of energy consumption in WSNs. They also offer a cluster size management mechanism and an energy-aware TDMA scheduling algorithm that divides the CH's coverage area into 24 sectors. Sun et al.[26] investigate the issue of energy-constrained WSN data routing. To discover the best path for data transmission in WSNs, they propose a new routing technique based on ant colony optimization. To reduce network energy consumption and extend network lifetime, the algorithm takes into account aspects such as node communication transmission distance, transmission direction, and residual energy. The authors also compare their algorithm with other algorithms that are already in use.

The aforementioned in CH selection for WSNs have largely relied on heuristic algorithms. In recent years, there has been growing interest in applying deep learning techniques to assist in CH selection. However, as this remains a nascent area of study, the number of published results is still limited, and even fewer works have explored the integration of deep learning with heuristic-based methods. Pandiyaraju et al.[27] proposed a deep learning-based clustering strategy for WSNs using a recurrent neural network (RNN)-based model, termed DL-Clustering. Their approach dynamically identifies CHs by learning node characteristics such as residual energy, proximity to the sink, and local node density. Similarly, Surenther et al.[28] combined an improved K-means clustering algorithm with RNN and Long Short-Term Memory (LSTM) architectures. In their two-stage process, K-means initially groups nodes based on energy and spatial proximity, after which an RNN extracts relevant node features and passes them to an LSTM module. The LSTM learns temporal patterns to predict the likelihood of a node being selected as a CH.

Debasis et al.[29] approached CH selection by partitioning the WSN deployment region into smaller sub-regions and employing an artificial neural network (ANN) to identify a single CH within each. Only nodes meeting a predefined energy threshold were considered, and the ANN assigned a fitness score based on four criteria: residual energy, number of detected events, distance from the base station, and local node density. Ravi et al.[30] addressed the issue of redundant data transmission in IoT-enabled WSNs by developing a clustering-based dependable data aggregation scheme. Their method employs an Optimized Whale Swarm Algorithm for initial clustering and CH selection, complemented by a hybrid deep learning model combining CNN and LSTMs to enhance decision accuracy and reliability. Akram et al.[31] proposed a machine learning-based clustering and routing framework aimed at prolonging WSN lifetime. Their model uses an ANN to autonomously select CHs and optimize routing paths, incorporating factors such as residual energy and node-to-sink distance. The model is adaptive and retrained across various deployment scenarios to improve energy efficiency and overall network performance.

Among these studies, only Ravi et al.[30] incorporate spatial data into a deep learning framework, but their methodology lacks a clear explanation of how the spatial information of WSN is converted into a CNN-compatible training data set. To the best of our knowledge, the present study is the first to introduce a structured and practical framework that enables a deep learning model to extract spatial features from WSN topology. Moreover, our scheme is the first to combine heuristic algorithms with radio energy consumption analysis to create a hybrid clustering strategy. This is a novel contribution to the area.

The Model of Wireless Sensor Networks

A typical RWSN consists of three fundamental components: sensor nodes, a base station (often referred to as a sink), and communication links.[32] Sensor nodes are distributed throughout the monitored area, tasked with detecting physical phenomena and sending this information to neighboring nodes or directly to the sink. These nodes also process collected data locally and transmit the resulting information to other nodes within the network. The sink functions as a centralized hub, aggregating and processing the data transmitted by the nodes. It also manages all sensing and monitoring tasks throughout the network, facilitating data collection and sharing. To improve energy efficiency, the sink is typically placed centrally within the network deployment area, minimizing the distance radio signals must travel and, thus, conserving energy.[33] As illustrated in Figure 1, the RWSN deployment considered in this study is represented as a square area, with the sink placed in its center.

With nodes having finite energy supplies, energy consumption is a significant consideration in the operation of WSNs.[34] The following formula can be used to calculate the energy used to transport a packet to a node or sink at distance d[35]:

$$E_{elec} \times l + E_{efs} \times l \times d^2, \forall\, d \leq \sqrt{\frac{E_{efs}}{E_{amp}}} \tag{1}$$

$$E_{elec} \times l + E_{efl} \times l \times d^4, \forall\, d > \sqrt{\frac{E_{efs}}{E_{amp}}} \tag{2}$$

where E_{efs} represents the energy consumption of the amplifier when transmitting signals to a shorter distance, E_{efl} represents the energy consumption of the amplifier when transmitting signals to a longer distance, E_{amp} denotes the energy per bit for

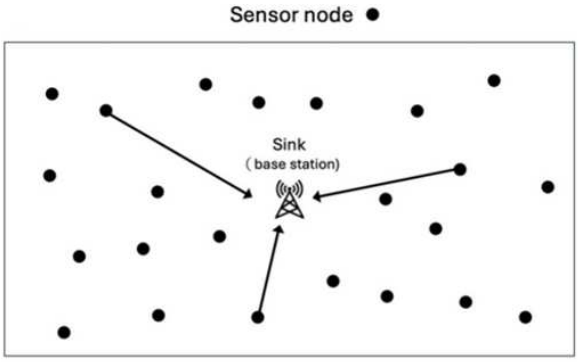


Figure 1. Basic structure of RWSN.

amplification, and l is the length of the packet. The amount of energy consumed during packet reception is $E_{elec} \times l$. The energy consumption of a wireless sensor to receive l bits of information is given by

$$l \times E_{elec} + l \times E_{BF}, \tag{3}$$

where E_{BF} denotes the energy expended in beamforming. The transmission distance has a direct effect on energy consumption. Optimal routing path selection can reduce energy consumption while ensuring sufficient coverage. Based on the foregoing, the routing protocols used have a significant impact on the energy consumption and dependability of RWSNs. Routing protocols are commonly used settings that take into account application needs, network capacity, and mobility patterns. Currently, widely used protocols such as LEACH and other hierarchical routing algorithms optimize energy consumption by grouping nodes into clusters and using CHs for data aggregation and transmission.[36]

Similar to LEACH and other widely adopted hierarchical routing algorithms, our proposed clustering protocol organizes sensor nodes into clusters and designates certain nodes as CHs to handle data aggregation and transmission, thereby improving energy efficiency. However, in contrast to existing clustering strategies, our approach determines the theoretically optimal number of clusters by analytically deriving it from the RWSN energy consumption model. Figure 2 illustrates the operational model of the RWSN under our proposed clustering mechanism.

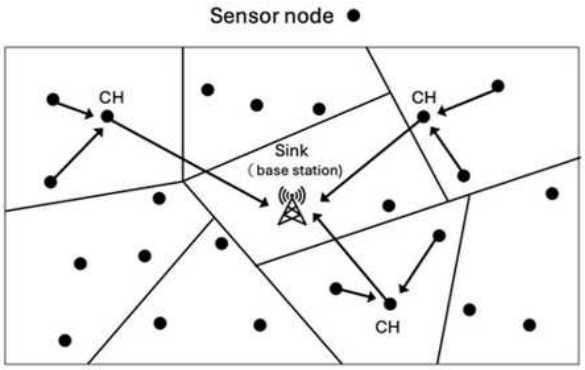


Figure 2. Operational model of our proposed clustering mechanism.

We consider an RWSN deployed over a square region, with nodes randomly distributed within the area. Given this network configuration, we derive an analytical expression for the total energy consumption of the RWSN. The number of clusters, k, is an important variable in the equation since the energy use of regular nodes is different from that of CHs. We can determine the value of k that minimizes the overall energy usage by differentiating this energy equation for k and solving for the critical point. The operational logic of our proposed clustering protocol is as follows:

- The network is divided into clusters, and each cluster contains a group of sensor nodes that are closer to each other than to nodes in other clusters. Initially, each node decides independently and probabilistically what its job will be, either as a CH or a regular member node, under the direction of the sink. The sink node makes all the clustering decisions and then broadcasts the cluster assignments to all the sensor nodes in the network.

- Randomized CH selection: Each node decides whether to become a CH or a regular node based on a predetermined probability threshold. The probability of any node being selected as a CH is determined by the number of nodes and the size of the map, the details of which will be explained in subsequent sections.
- CH rotation: In each round, CHs are chosen based on a probabilistic process, so nodes alternately become CHs. Nodes can share the energy cost of sending messages to the sink with the rotation mechanism, thereby extending the time of FND. Assume there are N nodes in the RWSN, and k of them are selected as CHs. Once a node has been selected as a CH, it is excluded from CH selection for the next $\lceil \frac{N}{K} \rceil$ rounds.
- Data aggregation: The CHs are responsible for gathering data from the regular nodes in each cluster. Before sending information to a sink, CHs compress the data. Energy is conserved when redundant data transmission is reduced by data aggregation.

The total energy consumption of the RWSN comprises three main components: E_c, the energy consumed by the k CHs in receiving data from member nodes within their respective clusters; E_s, the energy required for these CHs to transmit aggregated data to the sink; and E_i, the energy expended by regular nodes to transmit data to their assigned CHs. The total energy consumption of the entire network is denoted by

$$k \cdot (E_s) + (N-k) \cdot E_c + (N-k) \cdot E_i. \tag{4}$$

where N is the number of all nodes. The suggested protocol avoids configurations where any single cluster spans too much space in order to limit excessive energy use within individual clusters. This is because larger clusters would increase the distances between nodes within the same cluster, which would increase node energy consumption.

Assuming a uniform random distribution of all nodes in the sensing field, we define d_i as the average distance between a node and its corresponding CH. The term E_i is given by

$$E_{elec} \times\ l + E_{efs} \times\ l \times\ d_i^2, \tag{5}$$

as defined in Equation (1). Among the k CHs, on average,

$$\frac{(\sqrt{\frac{E_{efs}}{E_{amp}}})^2 \cdot \pi}{L^2} \cdot k = \frac{\pi \cdot E_{efs}}{L^2 \cdot E_{amp}} \cdot k \tag{6}$$

CHs are located at a distance from the sink smaller than $\sqrt{\frac{E_{efs}}{E_{amp}}}$, and $(1 - \frac{\pi \cdot E_{efs}}{L^2 \cdot E_{amp}}) \cdot k$ CHs are located at a distance from the sink greater than $\sqrt{\frac{E_{efs}}{E_{amp}}}$, therefore

$$E_s \approx (E_{elec} \times\ l + E_{efs} \times\ l \times\ d_s^2) \frac{\pi \cdot E_{efs}}{L^2 \cdot E_{amp}} \cdot k + (E_{elec} \times\ l + E_{efl} \times\ l \times\ d_l^4)(1 - \frac{\pi \cdot E_{efs}}{L^2 \cdot E_{amp}}) \cdot k. \tag{7}$$

Estimating E_i is particularly challenging, as it requires determining d_i, which in turn depends on calculating the minimum radius necessary to completely cover a square region using k identical circles. Since there is no general closed-form solution for this geometric problem,[37], we are limited to using other methods to estimate the approximate value of d_i. As previously described, the RWSN is assumed to operate within a square region of side length L. If this area is divided into k smaller squares, each with a side length of $\frac{L}{\sqrt{k}}$, and a random sensor node is selected to be the CH from each square, then we assume that the expected value of d_i is the average distance between two randomly chosen points within a square. This average distance, a well-studied quantity in the literature, can be denoted as $\frac{L \cdot (2+\sqrt{2}+5 \cdot ln(1+\sqrt{2}))}{15 \cdot \sqrt{k}}$.[38] Accordingly, the approximate energy consumption E_i can be expressed as

$$E_{elec} \times\ l + E_{efs} \times\ l \times\ (\frac{L \cdot (2+\sqrt{2}+5 \cdot ln(1+\sqrt{2}))}{15 \cdot \sqrt{k}})^2. \tag{8}$$

Consequently, Equation (4) can be reformulated as

$$\text{Equation (7)} + (N-k) \cdot [(l \times E_{elec} + l \times E_{BF}) + \text{Equation (8)}]. \tag{9}$$

To identify the value of k that minimizes Equation (9), we take its derivative and solve for the critical point by setting it to zero. However, it is important to note that Equation (9) provides only an approximate expression for the total energy consumption

of each node that transmits l bits of data within the RWSN. As such, the value of k obtained through this analysis may not represent the optimal solution in practice. However, experimental results demonstrate that the proposed algorithm significantly prolongs the time to FND, thus increasing the network lifespan compared to existing clustering approaches.

Since l doesn't affect the optimal value of k, we can divide Equation (9) by l before differentiating. The result of this differentiation is:

$$-(E_{elec}+E_{BF})+(E_{elec}+E_{efs}\cdot\ d_s^2)\frac{\pi\cdot E_{efs}}{L^2\cdot E_{amp}}+(E_{elec}+E_{efl}\cdot\ d_l^4)(1-\frac{\pi\cdot E_{efs}}{L^2\cdot E_{amp}})-\frac{N\cdot L^2\cdot(2+\sqrt{2}+5\cdot ln(1+\sqrt{2}))^2}{225\cdot k^2}. \quad (10)$$

To determine the optimal number of clusters in an RWSN with nodes that are distributed evenly, we set the derivative of Equation (9) with respect to k to zero and solved for $k = CHnum(L,N)$. This solution matches the network configuration that minimizes energy consumption. Calculating shows that the optimal number of CHs is:

$$CHnum(L,N)=\sqrt{\frac{N\cdot L^2\cdot(2+\sqrt{2}+5\cdot ln(1+\sqrt{2}))^2}{225\cdot[-(E_{elec}+E_{BF})+(E_{elec}+E_{efs}\cdot\ d_s^2)\frac{\pi\cdot E_{efs}}{L^2\cdot E_{amp}}+(E_{elec}+E_{efl}\cdot\ d_l^4)(1-\frac{\pi\cdot E_{efs}}{L^2\cdot E_{amp}})]}}. \quad (11)$$

However, it is important to note that Equation (9) holds only when the intra-cluster communication distance d_i is less than $\sqrt{\frac{E_{efs}}{E_{amp}}}$ meters.[39] This constraint arises from Equation (11), which is used to estimate the optimal cluster number. If d_i exceeds $\sqrt{\frac{E_{efs}}{E_{amp}}}$ meters, Equation (8) becomes inapplicable, resulting in Equation (9) being invalid. Therefore, there must be a lower bound on the value of k. In other words, if k is too small, the cluster coverage area may be too large, which significantly increases the likelihood that the distance between the nodes and their CH will be greater than the communication limit $\sqrt{\frac{E_{efs}}{E_{amp}}}$ meters.

Currently, no closed-form solution guarantees the number of clusters needed to keep all intra-cluster communication distances below this threshold.[37] Since our clustering strategy first selects CHs and then groups nearby nodes, and we don't want any node to be more than $\sqrt{\frac{E_{efs}}{E_{amp}}}$ meters away from its CH, we assume each cluster is a circular region with a maximum radius of $\sqrt{\frac{E_{efs}}{E_{amp}}}$ meters.

While no exact method exists to determine how many such circles are required to cover a square deployment area, we adopt an approximation using regular hexagonal tiling. In this model, the plane is divided into repeating equilateral hexagons, each circumscribed by a circle. To ensure full coverage, the radius R of each circle must reach the furthest vertex of its corresponding hexagon. The coverage density—defined as the ratio between the area of the circle and that of the hexagon—is given as[40]

$$\delta=\frac{\pi R^2}{\frac{3\sqrt{3}}{2}R^2}=\frac{2\pi}{3\sqrt{3}}\approx 1.209. \quad (12)$$

Assuming k such circles are required to cover a square area of side length L, the total area of these circles is $N\pi R^2$. This total should approximately equal the area of the square multiplied by the coverage density, i.e., $N\pi R^2 \approx L^2 \times \frac{2\pi}{3\sqrt{3}}$. From this, we derive a theoretical lower bound for k, denoted

$$N\approx\frac{2}{3\sqrt{3}}\cdot\left(\frac{L}{R}\right)^2\approx 0.385\cdot\left(\frac{L}{R}\right)^2. \quad (13)$$

Therefore, our clustering algorithm first employs Equation (13) to compute the lower bound of k, and then applies Equation (11) to determine its optimal value. If the result from Equation (11) exceeds the lower bound, it is adopted as the final cluster count. Alternatively, we use the lower bound, i.e.,

$$k=\begin{cases}CHnum(L,N), & if\ CHnum(L,N)>0.385\cdot\left(\frac{L}{R}\right)^2\\ 0.385\cdot\left(\frac{L}{R}\right)^2, & otherwise\end{cases}.$$

Once the value of k is determined, the probability of each node being selected as a CH during the selection process is set to $\frac{k}{N}$. Based on the above, a comprehensive flowchart of our proposed clustering protocol is presented in Figure 3.

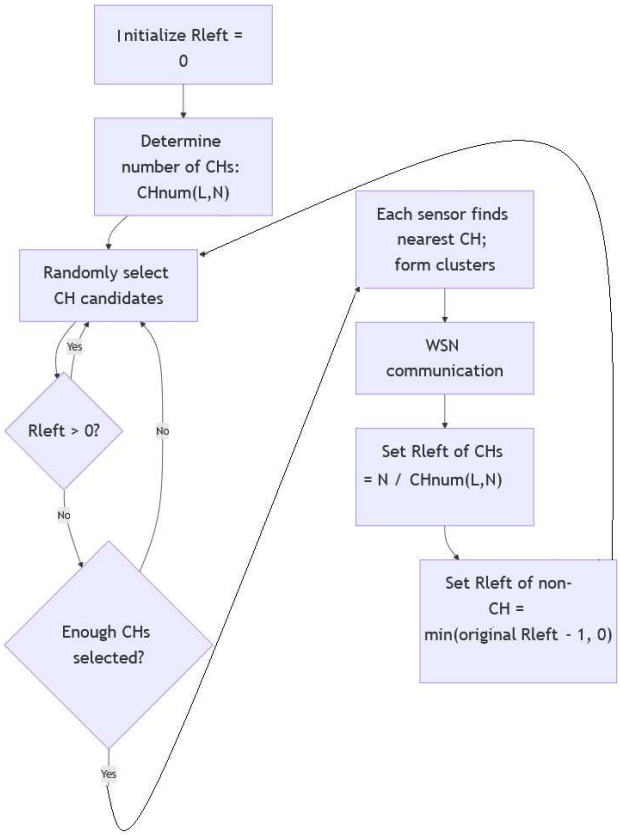


Figure 3. Operational model of our proposed clustering mechanism.

Genetic Algorithm in Cluster Head Selection

Principles of Genetic Algorithms

GAs are powerful optimization techniques inspired by the principles of natural selection and evolution.[41] Originally developed by John Holland[42] in the 1960s, GAs have found widespread application in solving complex problems across various domains. In the context of RWSNs, GA can be an effective tool to improve the selection of CHs in clustering protocols. Fundamentally, a GA operates by simulating the process of natural evolution to find optimal solutions within a search space. The fundamental components of a GA include the following:

- Population: A group of possible answers, frequently shown as chromosomes or individuals.
- Fitness function: Measure how well each solution solves the problem. The fitness function directs the search for more effective answers.
- Selection: Higher fitness scores mean a higher probability of being chosen for reproduction, so they reflect the survival of the fittest.
- Crossover: To discover new answers, a few individuals trade genetic information. This technique adds variation and stimulates research of the solution space. The chromosome crossover process of the genetic algorithm is shown in Figure 4.
- Mutation: The genetic information of individuals is randomly altered to provide variation and stop early convergence. The chromosome crossover process of the genetic algorithm is shown in Figure 5.

By repeatedly applying these procedures, GAs change the population over time, becoming closer and closer to the optimal or almost optimal solutions. The selection of suitable CHs has a major impact on aspects such as the network's lifespan, data routing efficiency, and distribution of energy usage. The fitness function is an important part of GAs that uses relevant parameters to judge the quality of potential CH candidates. These parameters can be defined by the researcher, such as

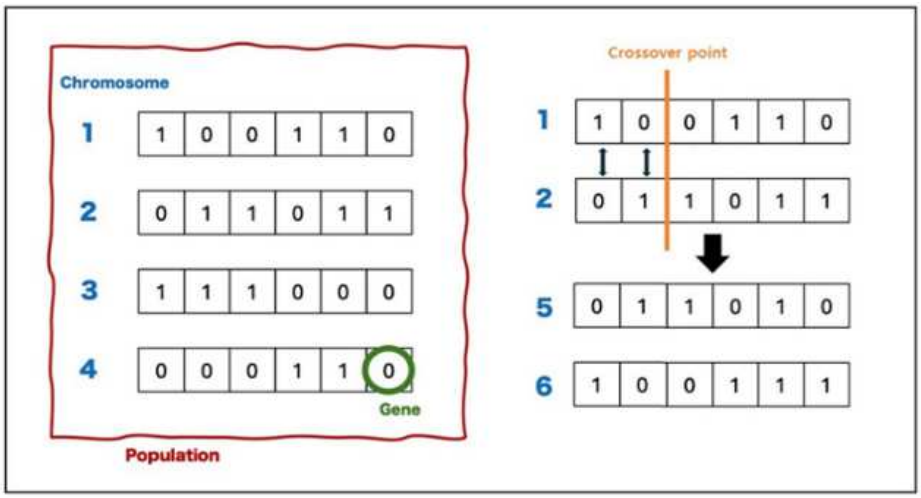


Figure 4. The crossover process in genetic algorithms.[43]

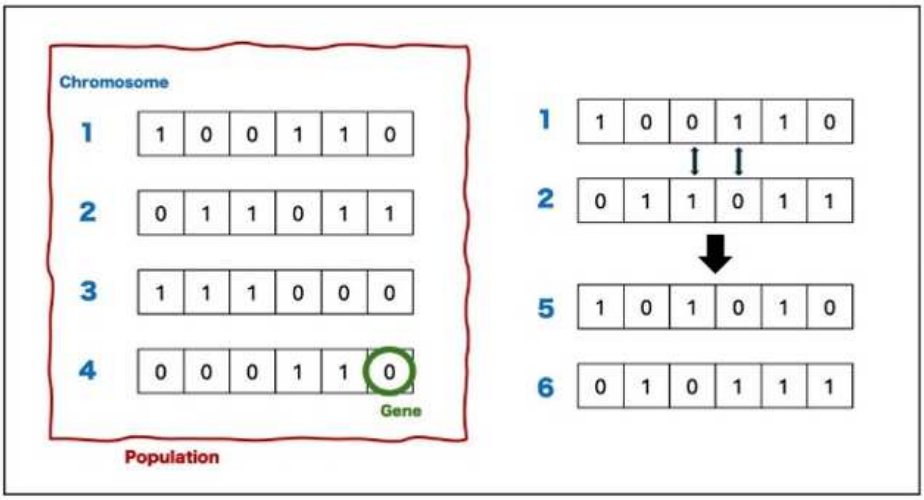


Figure 5. The mutation process in genetic algorithms.[43]

connectivity to nearby nodes, how close it is to the sink, and residual energy levels. Selecting CHs that maximize network performance and energy efficiency is guided by the fitness function. Several benefits result from the clustering protocol's use of GAs:

- Adaptability: For RWSNs operating in a variety of environments, GAs are ideal because they can adjust to changing network circumstances and environmental changes.
- Exploration of the solution space: Through the crossover and mutation procedures, GAs can investigate a large number of possible solutions and prevent early convergence to a local optimum.
- Optimization: GAs are capable of handling multi-objective optimization problems, allowing the simultaneous consideration of various network objectives such as energy efficiency and load balancing. The pseudocode of the genetic algorithm is shown in Algorithm 1.

Design of the Fitness Function

Next, we introduce a new fitness function designed specifically for this study. This function acts as a guide, helping the GA select CHs with the most advantageous features. The selection of these CHs directly enhances network performance. Below, we explain each component of the proposed GA fitness function below, along with its inclusion and contributions. Ultimately, our primary objective is to extend the operational lifespan of the RWSN.

Input: initial population, population size, chromosome length, and mutation rate
Evaluate fitness: Using the fitness function to evaluate the fitness value of each individual in the population.
while *termination condition is not met* **do**
    Select individuals from the population to be parent chromosomes based on their fitness.;
    Crossover;
    Mutation;
    Replacement: Replace some individuals in the current population with the new offspring.;
    Termination: Check termination conditions ;
end
Output: best individual.

Algorithm 1: Genetic algorithm.

- *CHNUM* (the number of CHs): The number of CHs significantly influences the effectiveness of clustering techniques and overall performance of RWSNs. In our configured RWSN environment, a network with fewer CHs typically consumes less energy, as CHs require greater energy resources compared to regular nodes due to their responsibilities in data aggregation and communication with the sink node. Therefore, a smaller number of CHs reduces the total energy expended on these tasks. Conversely, increasing the number of CHs can shorten the transmission distance between nodes and their CHs, thereby reducing the energy consumption for intra-cluster data transmission. Thus, optimizing the number of CHs to balance the two aforementioned types of energy consumption is critical to enhancing RWSN longevity. In this study, we calculate the optimal number of CHs based on the radio transmission energy formula and integrate this number into our proposed fitness function. To the best of our knowledge, previous research has not explicitly emphasized the importance of controlling the number of CHs within the design of a fitness function.
- *RSCD* (the relative sum of cluster distances): Energy consumption is affected by the transmission distance between the sensor and the CHs, as well as the distance between the CHs and the sink. Our fitness function considers this and preferentially selects CHs that can shorten the information transmission distance in order to reduce the transmission consumption of energy.
- *RLEFT* (the number of rounds the sensor cannot be used as a CH): The clustering protocol introduced here involves randomly selecting CHs in each operational round. This randomness makes sure that nodes get an equal number of CH roles throughout time. Additionally, once a node serves as a CH, it is excluded from subsequent CH selection for several rounds. This measure prevents excessive energy depletion in any individual node, thereby contributing significantly to prolonging the operational lifespan of the RWSN.

Based on the preceding, we propose the following fitness function:

$$F = a \times \ RSCD + b \times \ CHNUM + c \times \ RLEFT, \tag{14}$$

where a, b, and c represent the weights of the different variables. According to our best knowledge, the approach we presented to simultaneously regulate the number of CHs and the signal transmission distance to balance the additional energy consumption of every node as a CH is novel. In determining the weights a, b, and c, we consider the impact of each variable on energy consumption. A variable with a greater influence on energy consumption is assigned a higher weight to achieve the objective of reducing overall energy consumption. For RSCD, since the transmission distance between nodes and clusters is relatively short, we primarily focus on the energy consumption of data transmission between CHs and sinks, particularly in large-scale environments, such as square areas with a side length of 500 meters. Without loss of generality, we narrow our analysis to the first quadrant. Assuming the first quadrant forms a square of side length L, the expected distance between a node (located at (u,v)) and the sink (located at (0, 0)) is

$$\int_0^L \int_0^L \sqrt{u^2+v^2} f(x,y)\, dxdy, \tag{15}$$

where the probability density function $f(x,y)$ is $\frac{1}{L^2}$. Converting the coordinates into polar form, let $x = rcos\theta,\ y = rsin\theta$. Next, we take the partial derivatives of x and y with respect to r and θ to find the Jacobian matrix A. That is,

$$A = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \tag{16}$$

Since

$$\det A = (cos\theta \cdot rcos\theta) - (sin\theta)(-rsin\theta) = r(cos^2\theta + sin^2\theta) = r, \tag{17}$$

we obtain $dxdy = rdrd\theta$. Under this transformation, Equation (1) can be rewritten as

$$\frac{1}{L^2}\int_0^{\frac{\pi}{2}}\int_0^{L} r \cdot r\ d\ r\ d\theta. \tag{18}$$

Here, r ranges from (0, 0) to the maximum possible distance $\sqrt{2}L$ but is constrained by the square boundary, while θ varies within $\left[0, \frac{\pi}{2}\right]$. Therefore, the expected value is computed as

$$\frac{1}{L^2}\int_0^{\frac{\pi}{2}}\int_0^{L} r^2 dr d\theta = \frac{1}{L^2}\int_0^{\frac{\pi}{2}} \frac{L^3}{3} d\theta = \frac{1}{L^2} \cdot \frac{L^3}{3} \cdot \frac{\pi}{2} = \frac{\pi L}{6} \tag{19}$$

which the energy consumption corresponding to RSCD can be obtained from Equations (2) and (16) as

$$O\left(E_{elec} \times\ l + E_{efl} \times\ l \times \left(\frac{\pi L}{6}\right)^4\right) \tag{20}$$

The impact of CHNUM and RLEFT on energy consumption primarily relates to the frequency at which a node is selected as a CH. In our experiments, we set the probability of each node being selected as a CH to $\frac{CHNUM}{N}$, based on the side length L of the RWSN deployment environment and the total number of sensors N. In our model, the additional energy consumption incurred by assuming the CH role comes only from receiving messages. Thus, the energy consumption associated with CHNUM and RLEFT can be obtained from Equation (3) as

$$O\left(\left|\frac{N}{CHNUM}\right| \cdot (l \times E_{elec} + l \times E_{BF})\right). \tag{21}$$

Since Equations (20) and (21) reflect the magnitude of the impact of RSCD, CHNUM, and RLEFT on energy consumption, we use them to calculate the weight parameters a, b, and c in our subsequent experiments. We substitute the relevant wireless radio transmission parameters into these equations and normalize the results. This ensures that variables with different units can be fairly weighted and compared within the fitness function.

Proposed Lightweight ResNet model

Deep learning for RWSN lifetime extension

This study presents a novel deep learning architecture that is optimized for RWSNs. Our approach translates the entire RWSN environment, along with the precise status of each sensor, into an image representation. This image is then used to train a deep learning model, which identifies the most suitable sensors to act as CHs.

We used our custom-made GA, which we explained in the previous section, to select the optimal candidates for CH. Considering a RWSN deployed within an $L \times L$ square area, with sensor coordinates defined by integer values along the X and Y axes, we represent the network as an $L \times L$ pixel grid (G). Each sensor at coordinates (m, n) corresponds directly to the pixel located at row m and column n within grid G.

Each pixel in G has three color channels: Red, Green, and Blue (RGB). Each channel may contain data that is specific to a sensor. These channels record, respectively, the sensor's remaining energy, RLEFT, and the energy consumption required for the sensor to transmit data to the sink node. Pixels without corresponding sensors have all RGB channel values set to zero. Figure 6 shows how the results of each round of RWSN operation are converted into an image. By applying the advanced features of deep learning for image classification and the information we get from our GA computations, our model quickly determines whether each sensor is suitable to serve as a CH based on its location and encoded color information.

Overview of the ResNet Model

Based on the above, we propose an integrated architecture that combines our novel RWSN clustering protocol with a GA and a deep learning model for the selection of CHs. This integrated approach aims to significantly prolong the operational lifespan of the RWSN. However, considering the relative simplicity of the images generated by our architecture for training purposes, we

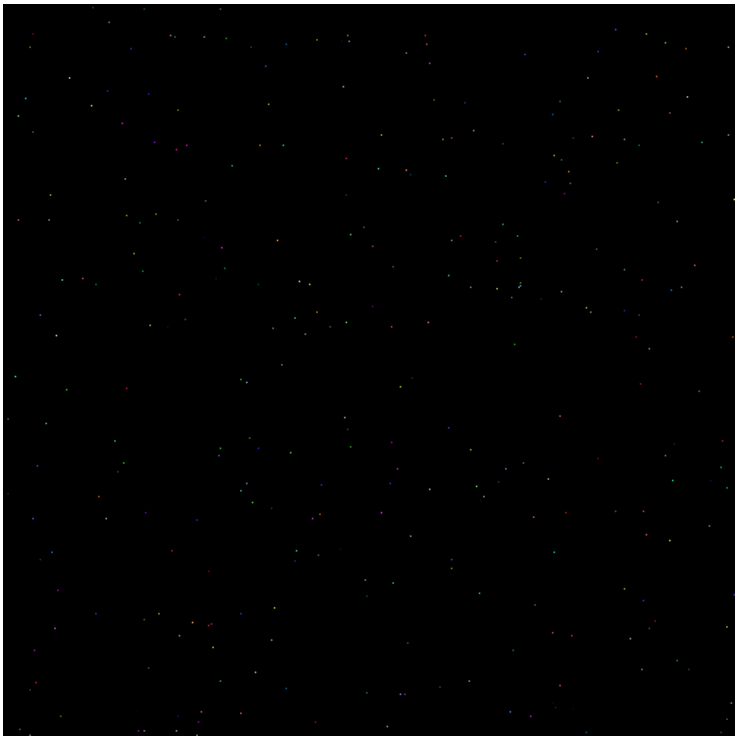


Figure 6. Training map. All non-black pixels correspond to a node and its current state.

modified the well-known ResNet model. Specifically, we simplified its architecture, creating a more lightweight structure that optimizes training efficiency without compromising resource utilization or performance.

ResNets modified the way deep learning works by solving the vanishing gradient problem, which has always made training deep neural networks more challenging.[44] A key innovation in ResNet is the introduction of residual learning blocks, which incorporate skip connections that directly add the input to the output of convolutional operations.[45] These identity mappings enable gradient signals to move more smoothly during backpropagation, thus assisting with problems like gradient vanishing or explosion. As a result, ResNets not only support the training of substantially deeper networks but also ensure that the addition of new layers does not degrade overall performance, even when such layers contribute minimally.[46]

At the core of ResNet is the residual mapping function, defined as $F(x) = H(x) - x$, where the network is trained to learn the residual function $F(x)$ rather than the direct mapping $H(x)$. In conventional deep learning architectures, each layer—or stack of convolutional layers—is trained to approximate a desired underlying mapping $H(x)$, transforming the input x directly into the target output. In contrast, the ResNet architecture reformulates this learning objective by focusing on the residual function, denoted as $F(x)$, which represents the difference between the target mapping and the input, such that $H(x) = F(x) + x$.

Rather than learning $H(x)$ directly, ResNet trains its stacked non-linear layers to model the residual $F(x) = H(x) - x$. There is a shortcut connection, known as a skip connection, that adds the output of this residual function to the original input element by element. This architectural innovation enables the model to bypass intermediate layers by directly propagating input information to deeper layers.

One of the key benefits of this design is its ability to facilitate gradient flow during backpropagation. ResNet solves the vanishing gradient problem by allowing gradients to go straight through the shortcut path. This allows networks to be trained much deeper than before.[47]

Each residual block typically consists of two 3×3 convolutional layers, interleaved with Batch Normalization and Rectified Linear Unit (ReLU) activations.[47] Batch Normalization accelerates convergence and stabilizes training, while ReLU introduces non-linearity and makes the model more expressive. The skip connection ensures that gradients can bypass multiple layers, thereby facilitating efficient optimization.

A standard ResNet pipeline includes input preprocessing, an initial convolutional layer for capturing low-level features, a stack of residual blocks, and final fully connected layers for classification.[48] Shortcut connections within residual blocks may either preserve dimensions via identity mapping or adjust them through 1×1 convolutions. The network is optimized using a suitable loss function and trained through backpropagation with iterative weight updates. The operational flow of a traditional ResNet is as follows:

- Input Preprocessing : Prior to being fed into the network, input data undergoes preprocessing steps such as normalization, which may include scaling pixel values and mean centering to ensure stable and efficient training.

- Initial Convolutional Layer : The network begins with an initial convolutional layer that extracts low-level visual features, such as edges and basic textures, from the input images.

- Residual Blocks : Following the initial convolution, the core structure of ResNet comprises multiple residual blocks. Each block typically consists of several convolutional layers, along with a skip connection that bypasses one or more layers.

- Internal Convolutional Layers : Within each residual block, successive convolutional layers are responsible for learning increasingly complex feature representations. These hierarchical features enable the model to capture high-level semantic information as the depth increases.

- Skip Connection Mechanism : A defining characteristic of ResNet is the use of skip connections, which allow the gradient to flow through shortcuts during backpropagation. When the dimensions of the input and output match, the skip connection performs an identity mapping. When the dimensions change, a projection shortcut with 1×1 convolutions is used to align the dimensions correctly.

- Activation Function : A non-linear activation function, typically the ReLU, is applied after each convolutional operation to make the model more expressive and add non-linearity.

- Feature Aggregation : At the end of each residual block, the outputs from the main path and the skip connection are added together, one element at a time. This aggregated result then passes through another ReLU activation function before being forwarded to subsequent layers.

- Fully Connected Layers and Output : The input to the fully connected layer is the Global Average Pooling (GAP) feature vector, which maps the high-dimensional feature vector to the number of categories required for the classification task. The Softmax function then converts it into a probability distribution, used for loss calculation in multi-class classification.

- Loss Function and Optimization Strategy : The network uses a method of optimization like stochastic gradient descent (SGD) with momentum or the Adam optimizer to minimize a task-specific loss function during training. For example, it uses cross-entropy loss for classification tasks or mean squared error for regression problems.

- Backpropagation and Weight Updates : Model parameters are updated iteratively through backpropagation, wherein gradients of the loss function with respect to the weights are computed and used to adjust the network's internal parameters in order to minimize the overall loss.

Figure 7 is a flowchart of the ResNet model.

Lightweight ResNet Model

In this study, we modify the ResNet design to make it work better for RWSNs, where speed and efficiency are particularly crucial. Convolutional structures can use spatial relationships in sensor deployment data, which are shown as simple images. To reduce computation time, we built a lightweight version of ResNet for the RWSN environment, enabling the GA to select optimal CHs based on image representations.

Key architectural modifications include:

- Reduced Depth: There are only two residual blocks left for feature extraction, each composed of two 3×3 convolutional layers with identity skip connections. To address the multi-scale nature of features in a simple image, we incorporate multi-stage feature fusion. Outputs from different residual blocks are concatenated along the channel dimension before GAP, enabling richer scale-aware feature representation. Therefore, since we don't need to adjust the dimensions, we can leave out all the 1×1 convolutional layers that are usually in the residual blocks of typical ResNet models, maintaining input size throughout. This decreases the number of parameters significantly while keeping spatial detail, which is very crucial for RWSN inputs.

- Cross-layer Skip Connections, which preserve direct information flow between residual stages and support adaptive feature fusion across depths.

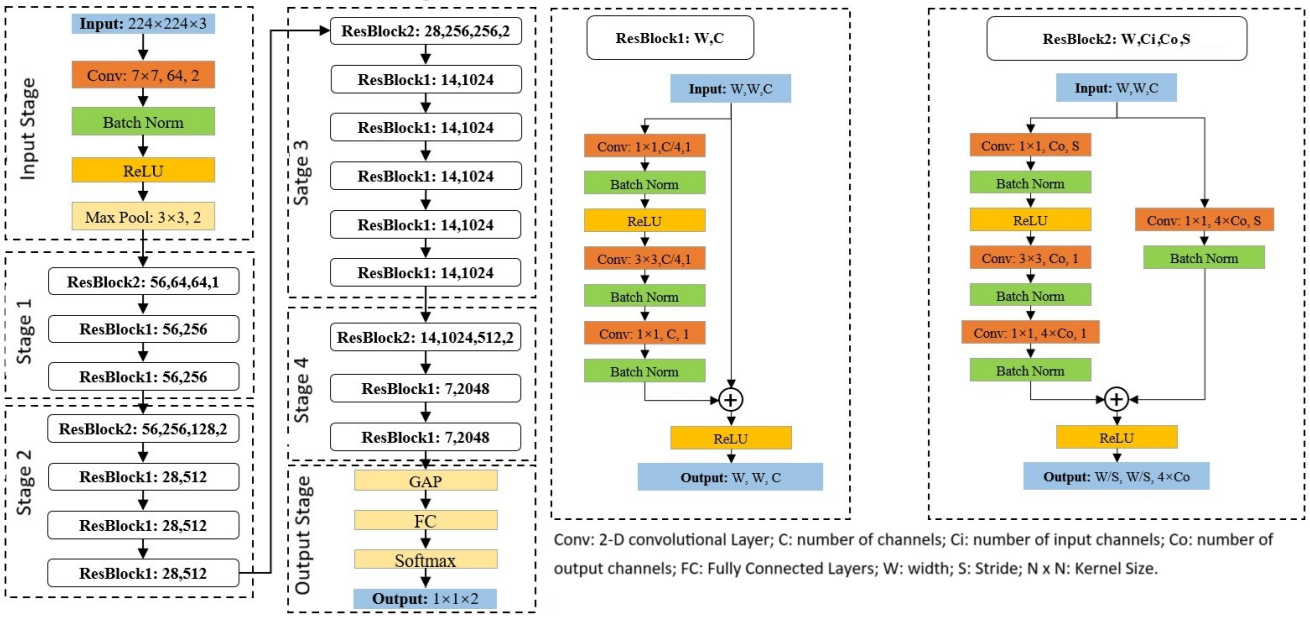


Figure 7. ResNet 50 flowchart.

- Minimized Fully Connected Layers: GAP is used to take the output of residual layers and replace the primary function of subsequent fully connected layers. This means that even though the fully connected layer is still in the proposed model, the input size is now only $1 \times 1 \times 2$. This modification in strategy cuts down on parameters by more than 90% and makes the model much better at generalizing.

- No Downsampling within residual blocks, maintaining full spatial resolution throughout and preserving fine-grained input details (stride=1).

- CPU-Oriented Optimization: The architecture, training iterations, and parameter number of the proposed lightweight ResNet model have been specifically optimized to enable efficient training on standard CPU platforms.

Figure 8 is a flowchart of our proposed lightweight ResNet model.

The proposed deep learning model was trained using 10,000 images of nodes identified as suitable for CH roles and 10,000 images of nodes deemed unsuitable, all generated through our proposed GA. These images were sampled by running RWSN on 10 network topologies until it reached the time to FND. To support real-time operation, the input module of the proposed model is designed to accept fixed-size figures that correspond to the dimensions of the RWSN deployment area. This design concept is both intuitive and practical for operators monitoring a fixed region. In our experiments, the largest RWASN was deployed within a square with sides of 501 meters. Therefore, the input for our model was fixed to a 501x501 pixel image.

In addition, the initial learning rate was adjusted to the range of 0.001–0.005, and the maximum number of training epochs was limited to 30. These modifications were introduced to reduce computational demands and enhance compatibility with CPU-based implementations. The proposed model was trained on a workstation running Ubuntu 22.04.4 LTS, equipped with an Intel Xeon Gold 6240R processor (2.4 GHz), an NVIDIA GeForce RTX 4090 GPU, and 512 GB of RAM. The dataset was divided into a training-to-validation ratio of 4:1. Training was conducted on a GPU with an initial learning rate of 0.001 and using a mini-batch size of 32. Model performance was evaluated on the validation set every 100 iterations. Although the training was conducted on a workstation equipped with a GPU, the model has been simplified to ensure compatibility with CPU-only computers. Once trained, the model can be run on a regular computer without a GPU, making it suitable for broader, resource-constrained application scenarios.

To enhance training robustness, the data were shuffled at the beginning of each epoch. A piecewise learning rate decay strategy was implemented, and training progress was visualized in real time. Figure 9 presents the accuracy and loss trajectories during training. The upper panel illustrates the evolution of training and validation accuracy as a function of iteration count, while the lower panel depicts changes in cross-entropy loss over the same period. To facilitate trend interpretation, both curves have been smoothed.

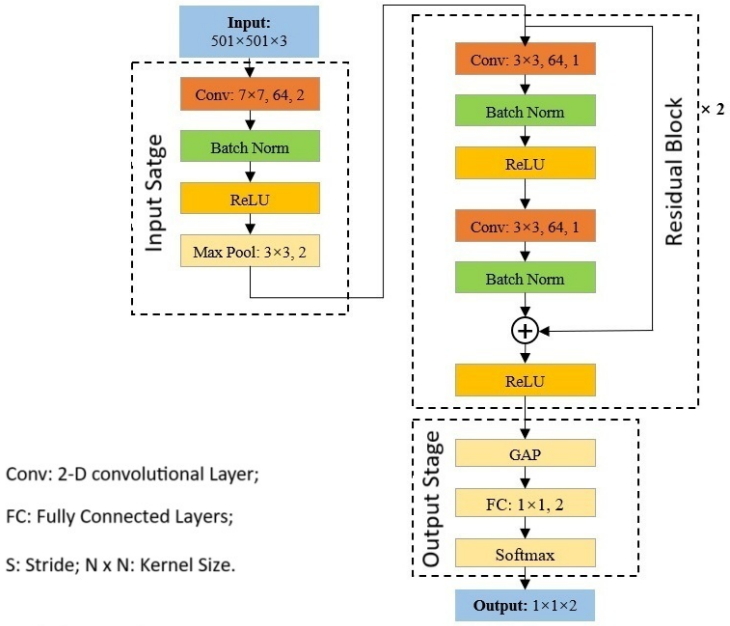


Figure 8. Lightweight ResNet Model. The input size is determined based on the dimensions of the RWSN deployment environment. In our experiments, the largest deployment area is a square with a side length of 501 meters.

Hybrid optimization framework

Here, we introduce an operating framework that extends the lifespan of RWSNs and postpones the time to FND. Ten independent RWSNs were created by randomly deploying sensor nodes within square regions, and each network was partitioned using the clustering protocol described in Section 3. Signal transmission was then simulated, and a GA identified the optimal set of CHs.

The GA output was converted into image-based training data to fine-tune a lightweight ResNet model, yielding a deep-learning model capable of predicting CH assignments (Figure 10).

During deployment, users simply provide the current network state; the trained model selects CHs in each round, after which the network re-clusters according to our protocol. Although heuristic methods can also prolong the lifespan of the network, they impose heavy computational consumption. Our hybrid framework combines GA-guided optimization with instantaneous inference to avoid the repeated calculations that are common in heuristic methods. This makes it an ideal option for RWSN scenarios as a replacement for time-consuming heuristic-based schemes. To reduce computation time, we built a lightweight version of ResNet for the RWSN environment to select optimal CHs based on image representations. (Fig. 11).

Experimental results

The results of the simulation experiments conducted in this study will be presented in this section. The hybrid scheme was simulated on a standard consumer-grade laptop running the Windows 10 (64-bit) operating system, equipped with an Intel Core i5-10210U processor (1.6 GHz), and 8 GB of RAM. The wireless-sensor-network dynamics were modelled in MATLAB. As we mentioned before, our research objective is to delay the appearance of sensor nodes that are insufficiently powered in RWSN. More precisely, the algorithm will perform better the more rounds the RWSN executes. To validate the suitability of our approach as a replacement for heuristic-based schemes in latency-sensitive RWSN application scenarios, we also compared it with recent approaches that adopt heuristic algorithms.

To ensure the robustness of our findings, we established a controlled experimental setup. The parameters we used for wireless transmission are enumerated in Table 1.[49] Before conducting the experiment, we substitute the data from Table 1 into Equations (20) and (21), setting the map's side length to 500 meters. The results showed that RSCD, CHNUM, and RLEFT had a proportional effect on energy consumption approximately 4:3:3. We set the weights for the three metrics, RSCD, CHNUM, and RLEFT, at 0.4:0.3:0.3 in the experiments based on these results. We first determined the contribution of each variable in our fitness function and then conducted a series of experiments to evaluate the performance of our proposed scheme, OPS, against four contemporary methods that combine clustering strategies with heuristic algorithms for CH selection in RWSNs. The comparative algorithms included GAOC, DCRN-GA, OGACHE, and MWSN.[11–13,50] Based on previous studies highlighting the importance of limiting the number of clusters, our experiment also followed this approach by restricting the number of CHs

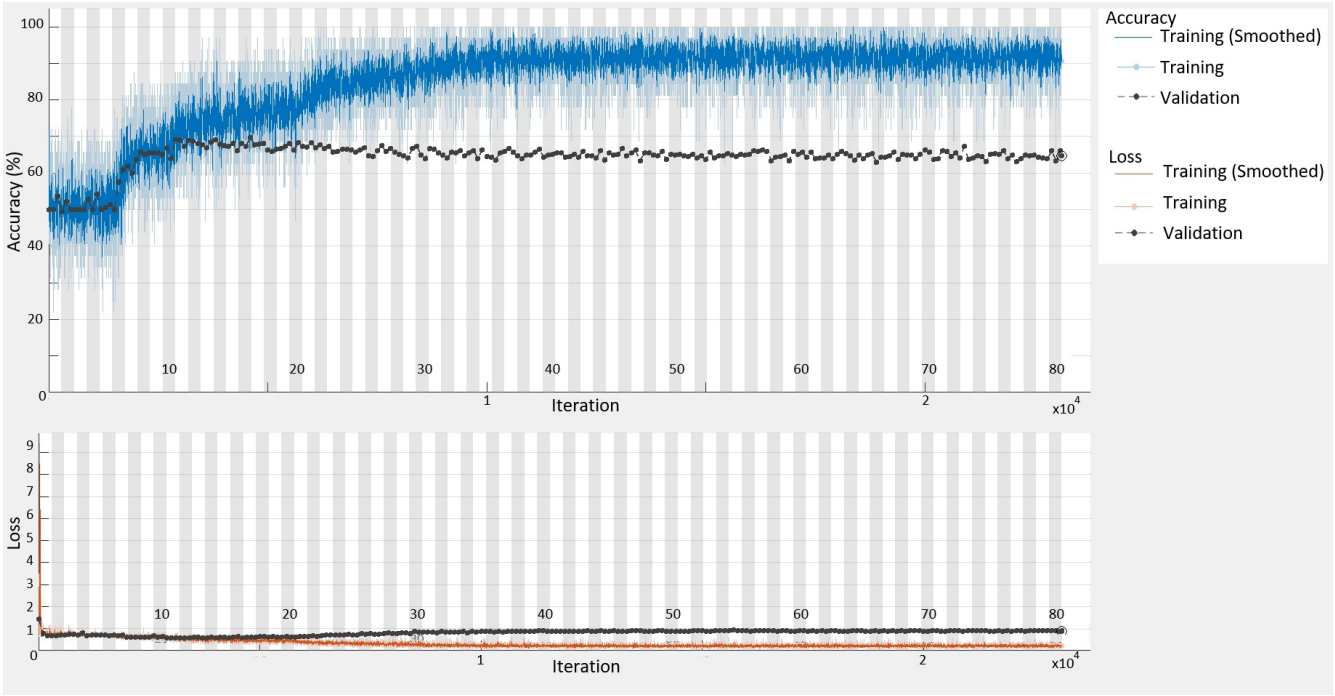


Figure 9. The ResNet

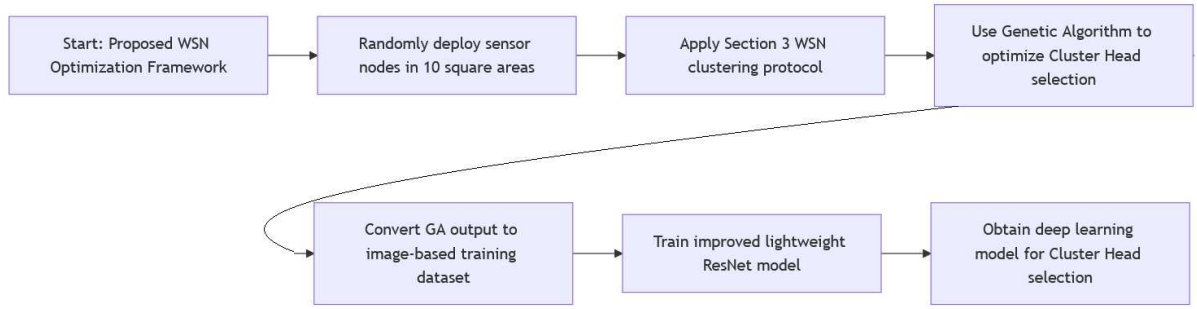


Figure 10. Framework flowchart.

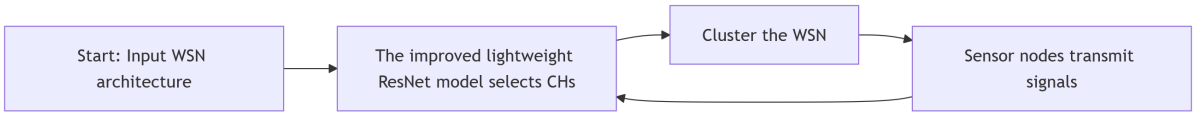


Figure 11. Proposed hybrid framework's operation in RWSNs.

to 10% of the total nodes when implementing other comparison schemes.[51] The results are presented as a reference.

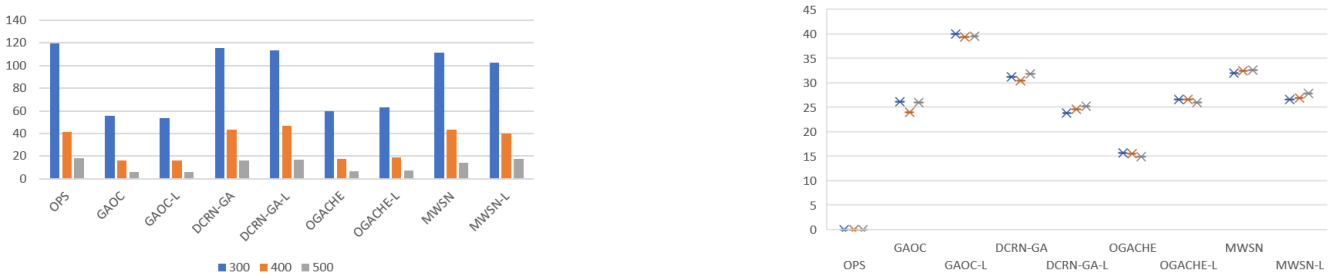
To ensure robustness, we randomly deployed 50 nodes within square regions measuring 300 m, 400 m, and 500 m per side, with the sink node positioned at the center of each region. Each node was initialized with 0.2 joules of energy. For every map size, ten randomized deployment scenarios were generated to account for environmental variability and to support comprehensive evaluation.

Figure 12 demonstrates that our hybrid strategy substantially extends the operational lifetime of the network, as measured by the number of communication rounds prior to the time to FND. Specifically, in environments with side lengths of 300 and 500 meters, our approach consistently outperforms all benchmark methods. Even in the 400-meter scenario, where it trails DCRN-GA and MWSN slightly, it still achieves 96% and 95% of their respective lifespans—indicating only marginal deviation from the top-performing strategies. The key advantage of our method lies in its reliance on a pre-trained deep learning model

Network parameters	Values
Network size	$300 \times 300,\ 400 \times 400,\ 500 \times 500\ m^2$
Number of nodes	50, 113, 200, 313
Packet size	2000 bits
Initial energy	0.2, 0.5, 0.48̃0.52 J/node
Transmitter energy	50 nJ/bit
Receiver energy	50 nJ/bit
Amplification energy for short distance, E_efs	10 pJ/bit/m2̂
Amplification energy for long distance, E_efl	0.0013 pJ/bit/m4̂
Consumption of energy during beam formation	5 nJ/bit
Data aggregation energy	5 nJ/bit
Maximum number of Iteration	100

Table 1. Parameters Used In Simulation Experiments

for CH selection, which eliminates the need for repetitive computations during runtime. Compared to other heuristic-based schemes, this leads to a notable reduction in overhead, which makes our scheme especially suitable to replace existing heuristic algorithms for latency-sensitive RWSN applications. Taken together, these experimental findings underscore the practical efficacy and competitiveness of our proposed scheme.



(a) Rounds

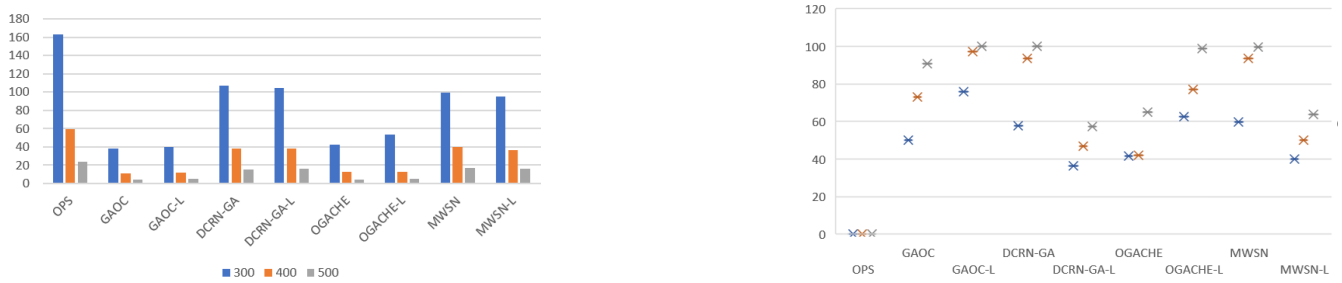
(b) Iterations

Figure 12. Performance comparison of OPS with other clustering mechanisms incorporating genetic algorithms. Adding the letter L after the algorithm name indicates that the experiment is conducted with a limited number of CHs.

Experimental results also showed that restricting the number of clusters during the GA population generation phase did not consistently improve the performance of other GA-based schemes. However, some versions of GA such as DCRN-GA and OGACHE can indeed significantly improve performance by controlling the number of clusters. Moreover, controlling the number of clusters can improve the performance of almost all other versions of GA when the map is small, that is, when the sensor density is high. These results can serve as a guidance for future related research.

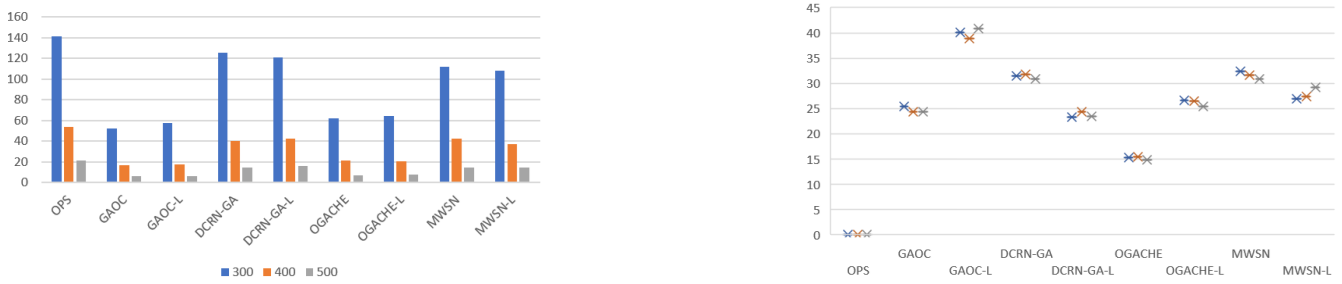
The density of nodes in the map will also affect the distance over which the sensors transmit information as well as the size of the map. A fixed number of 50 nodes was employed in the previous experiment, resulting in a reduced sensor density across larger maps. To evaluate the generalizability of our proposed fitness function across diverse deployment scenarios, we conducted experiments on maps of varying sizes while maintaining consistent sensor density. Specifically, we increased sensor density randomly deployed 113, 200, and 313 nodes on square maps with side lengths of 300 m, 400 m, and 500 m, respectively. All other experimental parameters were held constant to ensure a controlled comparison. As shown in Figure 13, our proposed hybrid method outperforms other recent schemes in sustaining RWSN longevity under fixed node density. This result highlights the method's advantages in both performance and computational efficiency, particularly in environments where the number of deployed nodes is not excessively low. One possible explanation is that higher sensor density tends to yield more uniform spatial distribution of nodes—conditions that closely align with the assumptions underlying the clustering strategy embedded in our design.

The previous two experiments assumed that each node had the same initial energy, but this is not always feasible in real-world circumstances. So, in the following two experiments, we randomly set the initial energy of each node between 0.18 J and 0.22 J. The side lengths of the maps used in the experiment are 300 m, 400 m, and 500 m, with 50 nodes randomly placed in each map. The experimental results are shown in Figure 14. In contrast to previous experiments, when the number of nodes was fixed at 50 and their initial energy levels were randomly assigned, our proposed hybrid scheme consistently achieved the highest number of rounds to the time to FND across all environmental configurations. This result demonstrates the robustness and adaptability of the hybrid scheme when employed in RWSNs with different initial energy levels.



(a) Rounds **(b)** Iterations

Figure 13. Performance comparison of OPS with other clustering mechanisms incorporating genetic algorithms on various map sizes while maintaining sensor density fixed



(a) Rounds **(b)** Iterations

Figure 14. Performance comparison of OPS with other clustering mechanisms incorporating genetic algorithms under different initial energies

In the previous experiments, square maps were utilized to position nodes randomly; however, such an environment may result in a reasonably regular RWSN architecture, which differs from some real RWSN application environments. To further assess the robustness of our proposed method, we conducted additional experiments using non-square deployment areas. Since our approach assumes a uniform and random distribution of nodes within a square region, it is essential to evaluate its effectiveness under less regular topologies.

For this set of experiments, nodes were randomly placed within modified deployment regions, as illustrated in Figure 15. These regions were derived by removing square segments from each of the four sides of an original square map, resulting in what we refer to as "X-shaped maps." We tested map sizes with side lengths of 300 m, 400 m, and 500 m, and removed square blocks measuring 10 m, 20 m, and 30 m on each side, respectively. For each configuration, 10 X-shaped maps were generated, each populated with 50 randomly distributed nodes.

The results, shown in Figure 16, demonstrate that our hybrid approach consistently outperforms most state-of-the-art GA-based clustering methods, even under irregular and uneven node distributions. In terms of RWSN lifetime, our proposed scheme performs slightly below that of the DCRN-GA and MWSN approaches, consistent with the results observed on the square map configuration. Specifically, across the three maps with varying deformation parameters, our scheme, on average, shows the time to FND at 89% and 91% of the rounds achieved by DCRN-GA and MWSN, respectively. While this performance gap has widened slightly compared to earlier square map results, the number of rounds to the time to FND achieved by our method remains very close to the best results. This makes it a highly suitable replacement for heuristic-based schemes in latency-sensitive RWSN applications.

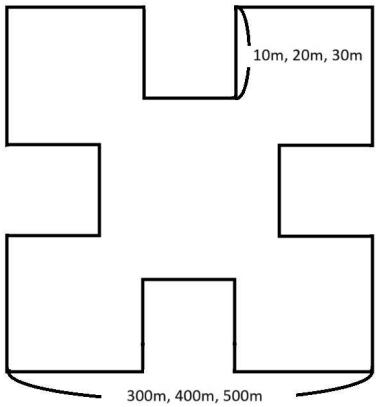
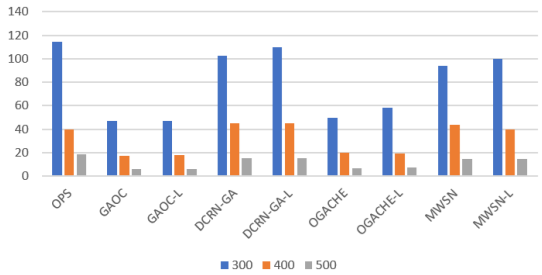
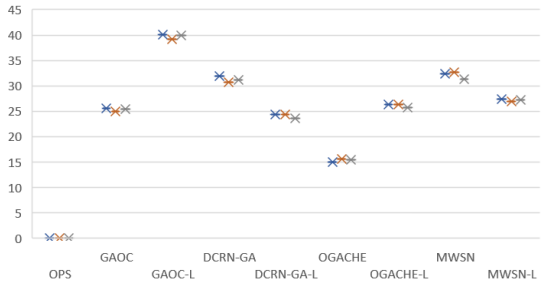


Figure 15. Schematic diagram of an X-shaped map.

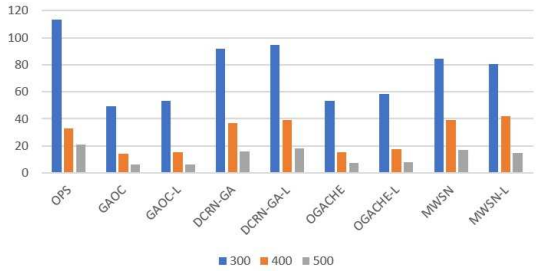


(a) Rounds

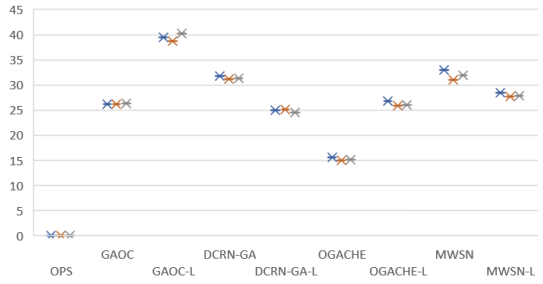


(b) Iterations

Remove a square block with a side length of 10m on each of the four sides.

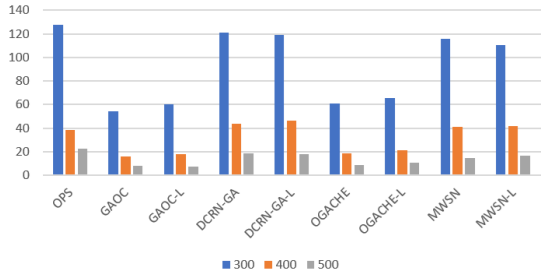


(c) Rounds

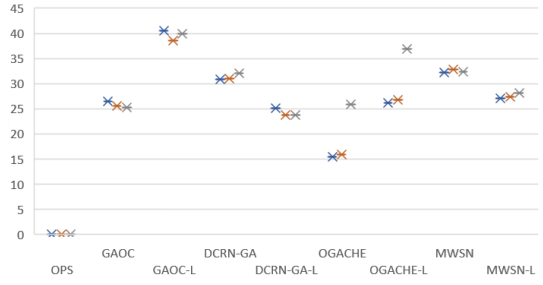


(d) Iterations

Remove a square block with a side length of 20m on each of the four sides.



(e) Rounds



(f) Iterations

Remove a square block with a side length of 30m on each of the four sides.

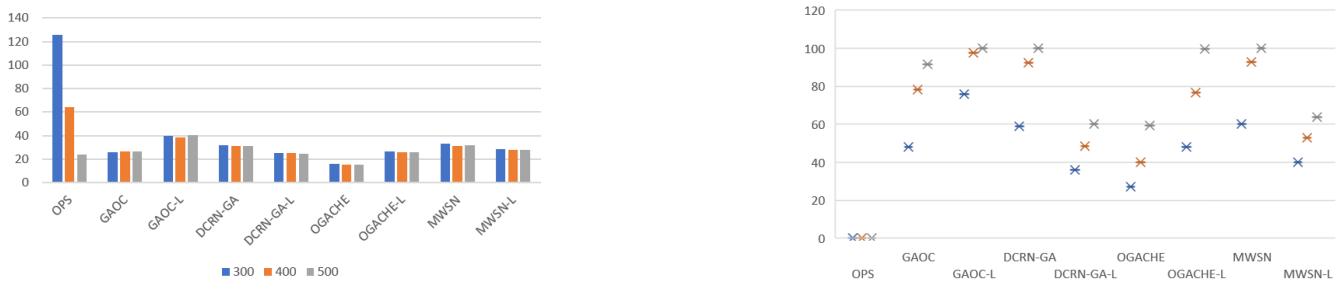
Figure 16. Performance comparison of OPS with other clustering mechanisms incorporating genetic algorithms on X-shaped maps

Next, we conducted experiments with increased sensor density; that is, 113, 200, and 313 nodes were randomly placed in the X-shaped map with the original square's side lengths of 300 m, 400 m, and 500 m, respectively. The experimental results are shown in Figure 17. Our experimental results demonstrate that the proposed hybrid scheme consistently outperforms the existing scheme across various map configurations. The scheme's superior performance is especially clear under high sensor density conditions, which greatly reduces the possibility of spatial abnormalities such as large coverage gaps caused by uneven node deployment. These results reinforce the validity of our analytical formulation, which was developed under the assumption of a uniformly random sensor distribution.



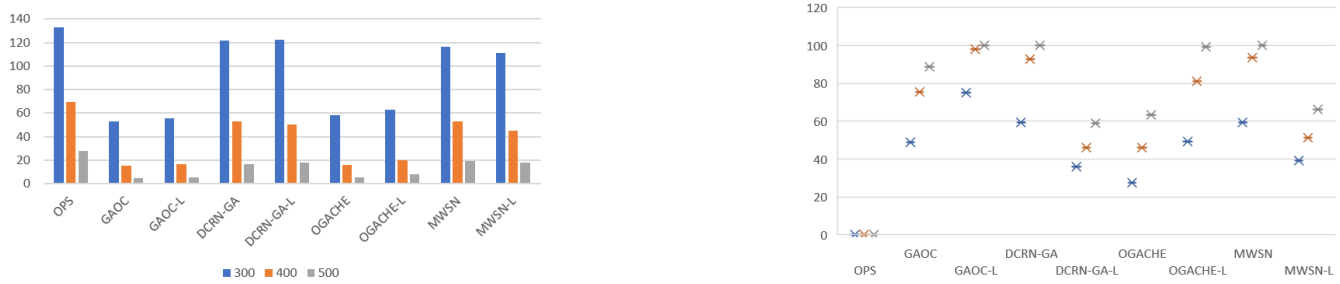
(a) Rounds **(b)** Iterations

Remove a square block with a side length of 10m on each of the four sides.



(c) Rounds **(d)** Iterations

Remove a square block with a side length of 20m on each of the four sides.



(e) Rounds **(f)** Iterations

Remove a square block with a side length of 30m on each of the four sides.

Figure 17. Performance comparison of OPS with other clustering mechanisms incorporating genetic algorithms on X-shaped maps with densities that are close around

In our earlier experiments using X-shaped maps, the advantages of the proposed hybrid approach were consistent with those observed in square-shaped maps. To further evaluate the robustness of our proposed scheme, we expand this non-square map experiment. Specifically, we modified square maps by removing small square blocks from each of the four corners. These removed blocks had side lengths of 10, 20, and 30 meters, respectively, as illustrated in Figure 18. This design eliminated corner-located nodes that were otherwise far from the map's center, resulting in what we define as a cross-shaped map.

Figures 19 and 20 present the experimental results conducted on cross-shaped map configurations. Consistent with earlier experimental findings, the proposed hybrid strategy maintains strong performance across all scenarios. When the number of

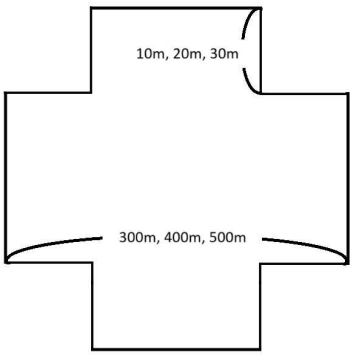
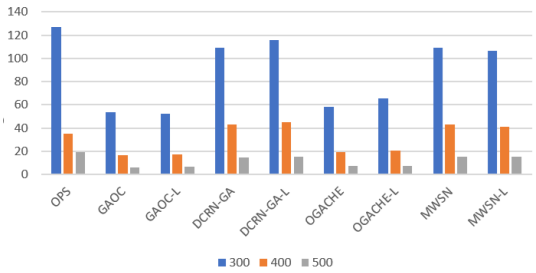


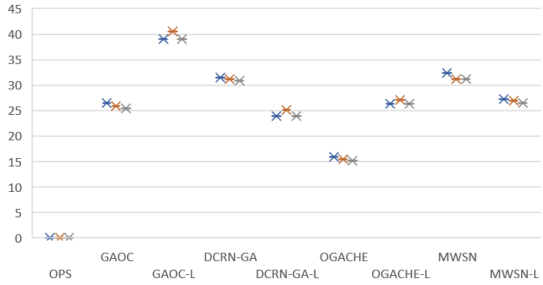
Figure 18. Schematic diagram of a Cross-shaped map.

nodes is held constant, our approach consistently outperforms all comparable methods.

However, in sparse deployment situations, especially on the 400-meter map, our scheme gets 81% and 83% of the RWSN lifespan that the heuristic-based DCRN-GA and MWSN schemes perform, respectively. One contributing factor may be the absence of nodes in the distant corners of the cross-shaped maps, which are far from the central sink. Our model incorporates the nodes located at the four corners when formulating the energy consumption equation for the entire RWSN, even though these nodes are not present on the cross-shaped map. Nonetheless, experimental results show that the performance gap between our approach and the best-performing results is minimal, and it outperforms most schemes based on heuristic algorithms.

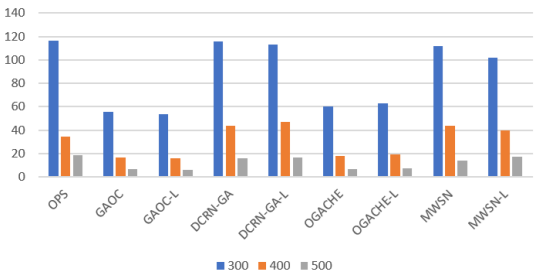


(a) Rounds

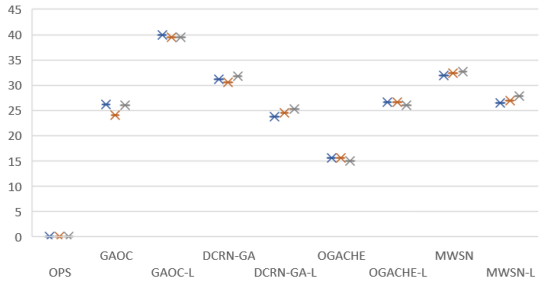


(b) Iterations

Remove a square block with a side length of 10m on each of the four corners.



(c) Rounds



(d) Iterations

Remove a square block with a side length of 20m on each of the four corners.

Figures 12 through 20 collectively indicate that while map shape exerts only a minor influence on RWSN lifetime, variations in map size significantly impact the convergence speed of heuristic-based optimization algorithms. Across most experimental configurations, our hybrid method consistently extends network longevity more effectively than leading heuristic schemes. Even in the few scenarios where its performance slightly lags behind that of DCRN-GA and MWSN, the margin is minimal. These findings underscore the suitability of our approach for real-time RWSN applications, particularly where the computational demands of heuristic clustering render such methods impractical.

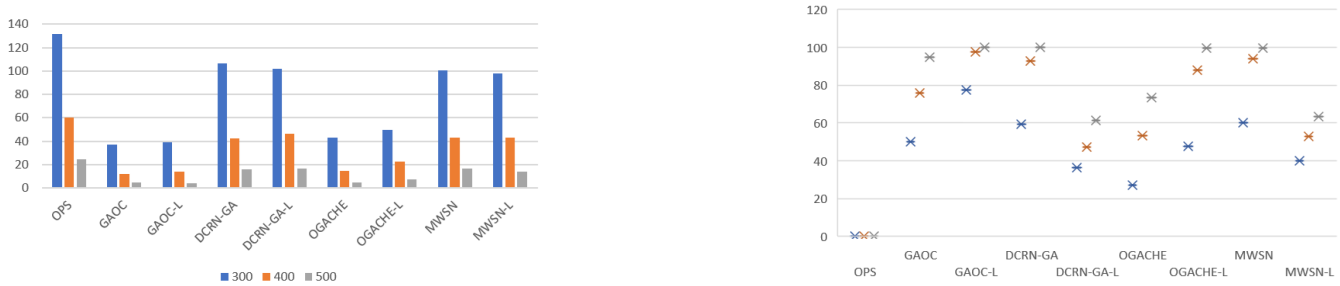
In the previous experiments, we varied the map environment and sensor settings several times. We then investigate the adaptation of several schemes to various settings by means of statistical analysis on all experimental data of X-shaped and



(e) Rounds **(f)** Iterations

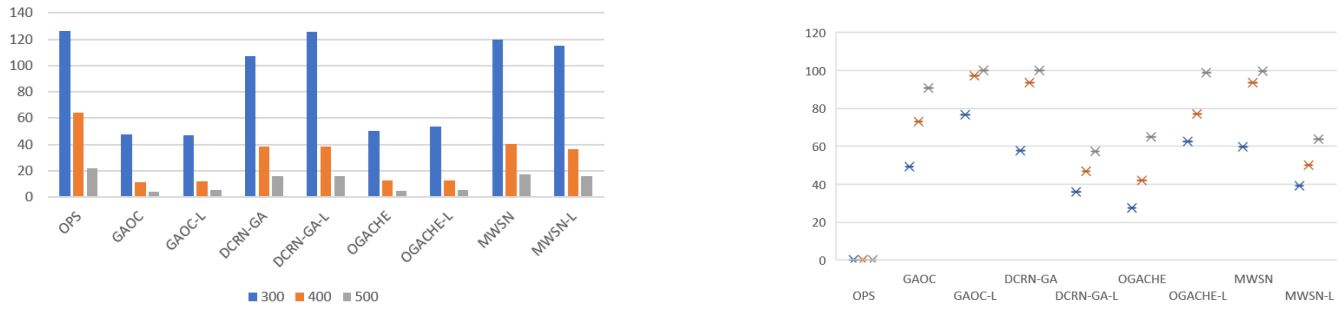
Remove a square block with a side length of 30m on each of the four corners.

Figure 19. Performance comparison of OPS with other clustering mechanisms incorporating genetic algorithms on Cross-shaped maps



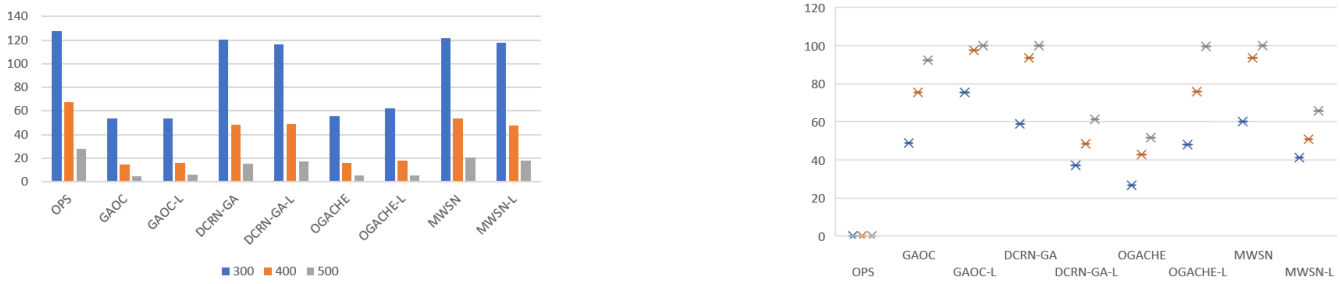
(a) Rounds **(b)** Iterations

Remove a square block with a side length of 10m on each of the four corners.



(c) Rounds **(d)** Iterations

Remove a square block with a side length of 20m on each of the four corners.



(e) Rounds **(f)** Iterations

Remove a square block with a side length of 30m on each of the four corners.

Figure 20. Performance comparison of OPS with other clustering mechanisms incorporating genetic algorithms on Cross-shaped maps with densities that are close around.

cross-shaped maps and various initial energies. It is expected that the algorithm can not only extend the lifespan of RWSN but also take into account the stability of performance on different maps. This will be more suited for practical uses since, upon deployment, the expected lifetime of a new RWSN can be more precisely predicted. We analyzed the standard deviation (SD), coefficient of variation (CV), range and median of the RWSN lifetime obtained in the experiment and presented the results in Table 2. As shown in Table 2, the proposed OPS method consistently achieves the highest median values across all three map scales, with its performance advantage becoming more pronounced as the map size increases. This indicates that OPS reliably delivers top-tier performance under typical deployment scenarios, and its comparative edge strengthens with increasing environmental complexity.

Furthermore, OPS exhibits the lowest CV among all evaluated approaches at both the 300 m and 500 m scales, suggesting minimal deviation from the mean and thus strong performance stability. Although the CV at the 400 m scale is moderately higher, it remains within an acceptable range, and does not undermine the method's overall robustness. These findings confirm that OPS not only excels in median performance but also demonstrates favorable consistency, as reflected in its low CV and narrow range in most settings.

The variations observed in the 400-meter environment may be attributed to occasional instances of exceptionally high efficiency during the experiment. While these cases increased the overall variability, they also underscored the potential upper bound of OPS performance. When evaluating long-term or average-case performance, OPS remains highly reliable.

Taken together, the four statistical measures—SD, CV, Range, and Median—support the conclusion that OPS offers a well-balanced trade-off between exceptional performance and moderate variability. As the experimental map got larger, its relative advantages and stability improved too. A larger map scale also makes the need to save energy obvious, which demonstrates OPS's applicability in more challenging environments.

Side length	Metric	OPS performance	Best in group	OPS vs. second place
300 m	SD	12.45	7.42	–
	CV	0.1	0.1	–
	Range	50.2	22.1	–
	Median	126.5	113.4	+11.6 %(+13.1)
400 m	SD	13.52	2.9	–
	CV	0.27	0.12	–
	Range	36.7	8.5	–
	Median	54	45.2	+19.5 %(+8.8)
500 m	SD	3	2.7	–
	CV	0.13	0.13	–
	Range	9.4	9.4	–
	Median	22.2	16.8	+32.1 %(+5.4)

Table 2. Statistics of experimental results

Conclusions

In this work, we introduce several advances for the clustering mechanism for RWSNs. First, the number of CHs has been related to the amount of energy consumed by radio transmission. An analytical solution was found for the optimal number of CHs and used in the clustering protocol. We then improve the GA optimizer by adding two more factors to its fitness function: a desired number of CHs and a cooling-off interval that stops recently selected CHs from being chosen in the next round.

Next, we suggest a deep learning architecture to solve the clustering problem in RWSNs by developing a lightweight version of ResNet that is made just for RWSN structures. By combining this network with the improved GA, we create a hybrid strategy that runs much faster than heuristic techniques and extends the lifespan of the network.

Comprehensive experiments demonstrate that the hybrid scheme dramatically increases energy efficiency and extends RWSN operational longevity. We also show that directly controlling the number of CHs, which is a key feature of our fitness function, improves the performance of several GA variations, particularly in high-density sensor node deployments. These results show that our strategy is useful in practical implementation and provides a clear way to improve current RWSN systems.

References

1. Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. Internet of things (iot): A vision, architectural elements, and future directions. *Futur. generation computer systems* **29**, 1645–1660 (2013).

2. Patil, K. & Banerjee, S. Brief overview on wireless sensor network technologies in the internet of things (iot). *Int. J. Eng. Manag. Res.* **13**, 1–8 (2023).

3. Sucasas, V. *et al.* A survey on clustering techniques for cooperative wireless networks. *Ad Hoc Networks* **47**, 53–81 (2016).

4. Betalo, M. L. *et al.* Dynamic charging and path planning for uav-powered rechargeable wsns using multi-agent deep reinforcement learning. *IEEE Transactions on Autom. Sci. Eng.* (2025).

5. Ahmad, W. *et al.* Optimizing energy efficiency in wireless sensor networks using enhanced k-means cluster head selection. *Int. J. Commun. Networks Inf. Secur.* **16**, 565–573 (2024).

6. Kaur, P. & Mehta, S. Resource provisioning and work flow scheduling in clouds using augmented shuffled frog leaping algorithm. *J. Parallel Distributed Comput.* **101**, 41–50 (2017).

7. Priyadarshi, R. Exploring machine learning solutions for overcoming challenges in iot-based wireless sensor network routing: a comprehensive review. *Wirel. Networks* **30**, 2647–2673 (2024).

8. Priyadarshi, R. Energy-efficient routing in wireless sensor networks: a meta-heuristic and artificial intelligence-based approach: a comprehensive review. *Arch. Comput. Methods Eng.* **31**, 2109–2137 (2024).

9. Al-Shalabi, M., Anbar, M., Wan, T.-C. & Alqattan, Z. Energy efficient multi-hop path in wireless sensor networks using an enhanced genetic algorithm. *Inf. Sci.* **500**, 259–273 (2019).

10. Tan, L., Wang, M., Wang, H. & Liu, L. Improved leach-m protocol for aerial sensor networks. In *2019 International Conference on computer, Information and Telecommunication systems (CITS)*, 1–4 (IEEE, 2019).

11. Nandan, A. S., Singh, S., Kumar, R. & Kumar, N. An optimized genetic algorithm for cluster head election based on movable sinks and adjustable sensing ranges in iot-based hwsns. *IEEE Internet Things J.* **9**, 5027–5039 (2021).

12. Rani, S., Ahmed, S. H. & Rastogi, R. Dynamic clustering approach based on wireless sensor networks genetic algorithm for iot applications. *Wirel. Networks* **26**, 2307–2316 (2020).

13. Verma, S., Sood, N. & Sharma, A. K. Genetic algorithm-based optimized cluster head selection for single and multiple data sinks in heterogeneous wireless sensor network. *Appl. Soft Comput.* **85**, 105788 (2019).

14. Shokouhifar, M. & Hassanzadeh, A. An energy efficient routing protocol in wireless sensor networks using genetic algorithm. *Adv. Environ. Biol.* **8**, 86–93 (2014).

15. Baranidharan, B. & Santhi, B. Gaech: genetic algorithm based energy efficient clustering hierarchy in wireless sensor networks. *J. Sensors* **2015**, 715740 (2015).

16. Rahmanian, A., Omranpour, H., Akbari, M. & Raahemifar, K. A novel genetic algorithm in leach-c routing protocol for sensor networks. In *2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 001096–001100 (IEEE, 2011).

17. Mollanejad, A., Khanli, L. M. & Zeynali, M. Dbsr: Dynamic base station repositioning using genetic algorithm in wireless sensor network. In *2010 Second International Conference on Computer Engineering and Applications*, vol. 2, 521–525 (IEEE, 2010).

18. Darji, H. & Shah, H. B. Genetic algorithm for energy harvesting-wireless sensor networks. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 1398–1402 (IEEE, 2016).

19. Sanapala, R. K. & Duggirala, S. R. An optimized energy efficient routing for wireless sensor network using improved spider monkey optimization algorithm. *Int. J. Intell. Eng. & Syst.* **15** (2022).

20. Panhwar, M. A. *et al.* Energy-efficient routing optimization algorithm in wbans for patient monitoring. *J. Ambient Intell. Humaniz. Comput.* **12**, 8069–8081 (2021).

21. Viet, T. N., Quynh, T. P. T. & Thi, H. D. Enhancement of leach based on k-means algorithm and stochastic optimization. *J. Commun.* **16**, 406–410 (2021).

22. Edla, D. R., Kongara, M. C. & Cheruku, R. A pso based routing with novel fitness function for improving lifetime of wsns. *Wirel. Pers. Commun.* **104**, 73–89 (2019).

23. Cheng, X., Xu, C., Liu, X., Li, J. & Zhang, J. Leach protocol optimization based on weighting strategy and the improved ant colony algorithm. *Front. Neurorobotics* **16**, 840332 (2022).

24. Hassan, A. A.-H., Shah, W. M., Habeb, A.-H. H., Othman, M. F. I. & Al-Mhiqani, M. N. An improved energy-efficient clustering protocol to prolong the lifetime of the wsn-based iot. *Ieee Access* **8**, 200500–200517 (2020).

25. Sinde, R., Begum, F., Njau, K. & Kaijage, S. Lifetime improved wsn using enhanced-leach and angle sector-based energy-aware tdma scheduling. *Cogent Eng.* **7**, 1795049 (2020).

26. Sun, Y., Dong, W. & Chen, Y. An improved routing algorithm based on ant colony optimization in wireless sensor networks. *IEEE communications Lett.* **21**, 1317–1320 (2017).

27. Pandiyaraju, V., Ganapathy, S., Mohith, N. & Kannan, A. An optimal energy utilization model for precision agriculture in wsns using multi-objective clustering and deep learning. *J. King Saud Univ. Inf. Sci.* **35**, 101803 (2023).

28. Surenther, I., Sridhar, K. & Roberts, M. K. Maximizing energy efficiency in wireless sensor networks for data transmission: A deep learning-based grouping model approach. *Alex. Eng. J.* **83**, 53–65 (2023).

29. Debasis, K., Sharma, L. D., Bohat, V. & Bhadoria, R. S. An energy-efficient clustering algorithm for maximizing lifetime of wireless sensor networks using machine learning. *Mob. networks applications* **28**, 853–867 (2023).

30. Ravi, G., Das, M. S. & Karmakonda, K. Reliable cluster based data aggregation scheme for iot network using hybrid deep learning techniques. *Meas. Sensors* **27**, 100744 (2023).

31. Akram, M. *et al.* Eemlcr: Energy-efficient machine learning-based clustering and routing for wireless sensor networks. *IEEE Access* (2025).

32. Maraiya, K., Kant, K. & Gupta, N. Application based study on wireless sensor network. *Int. J. Comput. Appl.* **21**, 9–15 (2011).

33. Joshi, D. & Jha, V. Energy-efficient multi-static sink model for wireless sensor network. *The J. Supercomput.* **81**, 1–32 (2025).

34. Patel, H. & Shah, V. A review on energy consumption and conservation techniques for sensor node in wsn. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 594–599 (IEEE, 2016).

35. Sahoo, B. M., Amgoth, T. & Pandey, H. M. Particle swarm optimization based energy efficient clustering and sink mobility in heterogeneous wireless sensor network. *Ad Hoc Networks* **106**, 102237 (2020).

36. Lung, C.-H. & Zhou, C. Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach. *Ad Hoc Networks* **8**, 328–344 (2010).

37. Heppes, A. & Melissen, H. Covering a rectangle with equal circles. *Period. Math. Hungarica* **34**, 65–81 (1997).

38. Ahn, J. & Krishnamachari, B. Derivations of the expected energy costs of search and replication in wireless sensor networks. *USC Comput. Eng. Tech. Rep. CENG-2006* **3** (2006).

39. Shih, T.-F. Particle swarm optimization algorithm for energy-efficient cluster-based sensor networks. *IEICE transactions on fundamentals electronics, communications computer sciences* **89**, 1950–1958 (2006).

40. Edelsbrunner, H., Iglesias-Ham, M. & Kurlin, V. Relaxed disk packing. *arXiv preprint arXiv:1505.03402* (2015).

41. Sivanandam, S., Deepa, S., Sivanandam, S. & Deepa, S. *Genetic algorithms* (Springer, 2008).

42. Holland, J. H. Genetic algorithms and the optimal allocation of trials. *SIAM journal on computing* **2**, 88–105 (1973).

43. Mallawaarachchi, V. Introduction to genetic algorithms—including example code. *Towards Data Sci.* **8**, 70–90 (2017).

44. Borawar, L. & Kaur, R. Resnet: Solving vanishing gradient in deep networks. In *Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022*, 235–247 (Springer, 2023).

45. Moniz, J. & Pal, C. Convolutional residual memory networks. *arXiv preprint arXiv:1606.05262* (2016).

46. Zhu, Z., Wang, S. & Zhang, Y. Roenet: A resnet-based output ensemble for malaria parasite classification. *Electronics* **11**, 2040 (2022).

47. Ceni, A. Random orthogonal additive filters: a solution to the vanishing/exploding gradient of deep neural networks. *IEEE Transactions on Neural Networks Learn. Syst.* (2025).

48. Aach, M., Inanc, E., Sarma, R., Riedel, M. & Lintermann, A. Large scale performance analysis of distributed deep learning frameworks for convolutional neural networks. *J. Big Data* **10**, DOI: 10.1186/s40537-023-00765-w (2023).

49. Mukti, F. S., Junikhah, A., Putra, P. M. A., Soetedjo, A. & Krismanto, A. U. A clustering optimization for energy consumption problems in wireless sensor networks using modified k-means++ algorithm. *Int. J. Intell. Eng. & Syst.* **15** (2022).

50. Badamchi, D. Clustering mobile wireless sensor networks (mwsns) with heuristic algorithms. *Master Diss.* (2016).

51. ALharthi, L., Ahmed, A. E. & Ibrahim, M. E. An energy efficient routing algorithm using chaotic grey wolf with mobile sink-based path optimization for wireless sensor networks. *Int. J. Adv. Comput. Sci. & Appl.* **14** (2023).

Acknowledgements

The authors would like to acknowledge the use of OpenAI's ChatGPT as a language editing tool during the preparation of this manuscript. All content, ideas, and analyses presented in this paper were solely conceived and developed by the authors; ChatGPT was used only for refining the English language and improving the clarity of expression.

Author contributions statement

Y.W. Ti conducted the formal analysis and developed the methodology. R.H. Cheng was responsible for the conceptualization of the study. W. Huang conducted the experiments. S. Wei contributed to the writing of the manuscript. C.M. Yu supervised the overall research process. All authors reviewed the manuscript.

Funding Declaration

1. Fujian Provincial Key Laboratory of Marine Physical and Geological Processes, No. KLMPG-25-05.
2. Natural Science Foundation of Xiamen, China, No. 3502Z202374070.
3. Scientific Research Foundation for Advanced Talents of Xiamen Ocean Vocational College, No. KYG202206.
4. National Science and Technology Council, Taiwan, No. NSTC 114-2221-E-033-026.

Declarations

Competing interest

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S. W.