Cluster Computing (2026) 29:151
https://doi.org/10.1007/s10586-026-05966-5

Artificial neural network based cluster head selection process in wireless sensor networks

Volkan Gangal[1] · Erhan Sesli[2] · Gokce Hacioglu[3]

Received: 26 March 2025 / Revised: 5 January 2026 / Accepted: 14 January 2026
© The Author(s) 2026

Abstract

Wireless Sensor Networks (WSNs) face a significant challenge in extending network lifetime due to their limited energy capacity. This study proposes an energy-efficient, distributed cluster head selection method supported by artificial neural networks (ANNs) to enhance the energy efficiency of cluster-based sensor networks. The methodology allows each node to autonomously determine its optimal selection threshold, addressing the computational constraints of WSNs through a two-stage process: a Cascade-ANN is trained offline to model the complex relationship between a node's energy status and the optimal threshold. Subsequently, its learned behavior is translated into lightweight mathematical equations via curve fitting, which eliminates the need for nodes to run the ANN directly. Under the evaluated simulation scenarios, the results suggest that the proposed method improves key performance metrics compared to the standard LEACH protocol, within the defined network parameters. Specifically, network lifetime was observed to extend by approximately 97% (from 1773 to 3497 rounds), and total data packets successfully transmitted to the sink showed an increase of approximately 57% (from 61,766 to 96,757 packets). Furthermore, a comparative analysis indicates that the proposed method performs favorably against similar recent studies, with a performance factor ranging from 3.63 to 6.26 in terms of network lifetime. This work aims to contribute a robust and scalable solution that may enhance network performance.

Keywords Wireless Sensor Networks · ANN-Based WSN Clustering · Cluster Head Selection · Distributed Clustering

1 Introduction

Wireless sensor networks (WSN) consist of multiple sensor nodes that gather information, such as outperforms, humidity, images, and more, from their surroundings for different purposes, including military surveillance, patient health monitoring, and detecting natural disasters [1, 2]. Despite their widespread utility, WSNs inherently encounter critical challenges, particularly concerning network lifetime and energy efficiency [3–5]. Energy depletion in individual nodes can lead to network disconnections and overall dysfunction, making energy management a paramount concern. A primary focus for researchers aiming to extend network lifetime is minimizing energy consumption during data transmission and reception. It is widely recognized that the most substantial energy expenditure in a WSN node occurs during the communication phase [6, 7]. Consequently, various energy-saving algorithms and routing protocols are being developed, with clustering serving as a particularly effective strategy for optimizing energy consumption by organizing nodes into groups [8–10].

E. Sesli and G. Hacioglu these authors contributed equally to this work.

✉ Erhan Sesli
erhansesli@ktu.edu.tr

Volkan Gangal
volkan.gangal@ktu.edu.tr

Gokce Hacioglu
gokcehacioglu@ktu.edu.tr

[1] Trabzon Vocational School of Higher Education, Karadeniz Technical University, Akcaabat, Trabzon 61300, Turkey

[2] Faculty of Technology, Department of Electronics and Telecommunication Engineering, Karadeniz Technical University, Trabzon 61830, Turkey

[3] Department of Electrical and Electronics Engineering, Karadeniz Technical University, Trabzon 61080, Turkey

Published online: 16 February 2026

The Low Energy Adaptive Clustering Hierarchy (LEACH) protocol is a well-known clustering method [9–11]. Numerous works aimed at improving the performance of the LEACH algorithm have been published [9, 12–16]. WSNs face significant challenges in terms of energy efficiency and network lifetime. To address these challenges, AI models, particularly Artificial Neural Networks (ANNs), have been explored for tasks such as clustering and classification, aiming to reduce communication overhead and extend the operational lifetime of the network [17–23].

Even though several lightweight AI models such as decision trees, support vector machines, or k-nearest neighbors could also be employed for decision-making in WSNs, their capacity to capture highly nonlinear relationships among node parameters is limited. In this study, we propose an ANN-based threshold determination method for cluster head (CH) selection, which distinguishes itself from existing work by allowing each node to determine its threshold using a pre-trained ANN independently. Cascade-ANN, by contrast, benefits from additional connections across layers, which enhances convergence and improves generalization when modeling residual energy. In the simulations, Cascade-ANN achieved higher predictive accuracy compared to alternative topologies. While the computational cost of ANN models is indeed higher, this drawback can be mitigated by transforming the ANN outputs into mathematical threshold energy equations via curve fitting. This post-processing step allows sensor nodes to operate with simple equations rather than executing the ANN itself, thereby avoiding additional processing overhead. Consequently, the Cascade-ANN architecture was selected for its high learning performance during the training phase and its potential for simpler deployment at the node level.

In light of the current approaches to ANN-based energy efficiency in WSNs within the literature, the main contributions of this study can be summarized as follows:

The study aimed to reduce the energy consumption of the network by determining the threshold value for CH selection by each node separately. This was done with the help of an ANN. In the considered scenario, node coordinates and sink are fixed, and nodes are categorised into four different categories according to their distance to the sink. Possible threshold values were sequentially applied to the input of the Cascade-ANN, allowing the ANN to estimate the remaining energy for each threshold. Consequently, the threshold value for each node was determined individually in a distributed manner using the ANN. The proposed method assumes a pre-trained ANN to be present at each node. The node decides which threshold value to use based on the output it receives by providing a set of possible threshold values as inputs to the ANN. The proposed method utilizes the distributed clustering mechanism of the LEACH algorithm. However, it determines the threshold values by consulting an AI model trained on past experiences. A review of the literature indicates that while ANNs are widely used to optimize clustering protocols, their specific application to determine the probabilistic threshold in LEACH has received less attention. In the proposed method, an ANN is utilized at each node to determine the threshold value, with each node deciding its threshold based on its own energy status and that of its neighbors. This work therefore contributes to this area by proposing a novel ANN-based method for threshold determination. The study's results indicate that the proposed ANN-based CH selection method can enhance energy efficiency of the network. Specifically, the method achieved 3497 operational rounds and transmitted 96757 packets, which is higher than the LEACH algorithm, which achieved 1773 rounds and transmitted 61766 packets under the same conditions. Moreover, compared to similar recent studies, our proposed approach demonstrated notable efficiency gains, with improvements ranging from 3.63 to 6.26 times.

The remainder of the paper is organized into five sections. In Sect. 2, we evaluate related works. In Sect. 3, a detailed explanation is provided of the proposed method for CH selection based on ANNs. Section 4 presents the performance results obtained from the proposed method. Finally, Sect. 5, includes a discussion and draws conclusions based on the study.

2 Related works

The most well-known method among clustered WSN techniques is the LEACH method. In LEACH, nodes probabilistically elect themselves as cluster heads (CHs) for each round, and this role is rotated among nodes to distribute the energy load evenly. The selected CHs aggregate data from their cluster members and transmit the combined information to the sink, reducing overall communication and conserving energy. This approach minimizes energy consumption, extends network lifetime, and supports scalability, making LEACH a widely adopted protocol in WSNs. The flow chart of the LEACH is depicted below Fig. 1.

Random selection of CHs in the LEACH protocol is a severe limitation because it ignores crucial factors such as the remaining energy of the node. In order to overcome this problem, many researchers have suggested more intelligent and energy-sensitive solutions. Machine learning and optimization techniques are among the methods that are widely used in the literature for more effective CH selection. Some of these sophisticated methods are examined in the next paragraphs, their potential advantages and limitations are explained, and the stage is set for the solution proposed in this paper.

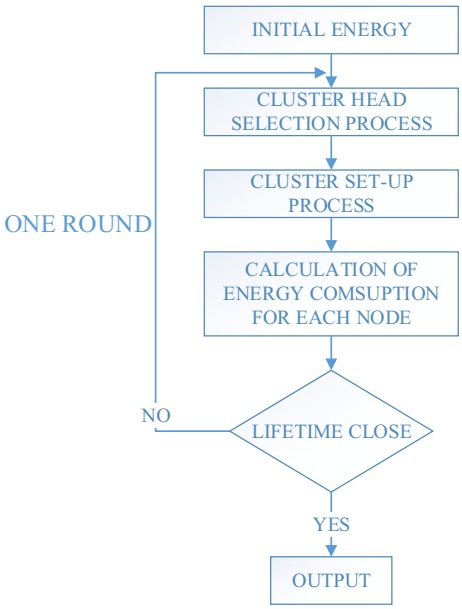


Fig. 1 Flow chart of LEACH protocol

In [24], Sharma and Kansal presents a hybrid methodology where an ANN acts as a security layer to filter malicious nodes. In contrast, this work redesigns the core decision-making logic of the LEACH protocol by using a Cascade-ANN to learn the optimal threshold behavior offline, then deploying a simple, lightweight mathematical equation onto the nodes. Similarly, in [25] Qamar et al. combined k-means and ANN to optimize the path of a mobile sink, primarily focusing on mobility and delay constraints. While this method also targets overall energy reduction, it does so by optimizing the sink's route. Proposed approach in this article differs in its focus and application: this work concentrates on static WSNs and enhance energy efficiency directly at the node level through an improved CH selection mechanism. As such, their solution is well-suited for mobile environments, whereas this work provides a methodology tailored for the operational context of static sensor networks. The OANN-EATSRA method in [21] Sharma and Kansal presents a multi-stage approach for secure routing, combining an advanced LEACH protocol with the Grasshopper Optimization Algorithm (GOA) and an ANN classifier. Their results indicate low computational complexity regarding memory footprint. Differently, proposed approach in this study places a primary emphasis on minimizing the runtime processing load on individual nodes. The approach in our methodology achieves this by leveraging an offline training phase for a complex ANN, which allows for the deployment of a single, ultra-lightweight mathematical equation for real-time, on-node decision making. This distinction highlights our focus on improving the practicality of deployment on resource-constrained nodes.

Beyond these studies the other notable publications in recent years addressing energy-efficient CH selection methods in wireless sensor networks, related to our study, are delineated in Table 1. A synopsis of the content within these publications is provided below:

In [26], Debasis et al. introduced the EECA algorithm, where an ANN scores nodes to select the most qualified CH in each region Table 1. Differing from this score-and-rank methodology, our work focuses on optimizing the probabilistic foundation of the LEACH protocol itself. By using an offline-trained ANN to derive a simple predictive equation for the selection threshold, our method is tailored to be exceptionally lightweight at the point of deployment, emphasizing practicality for energy-sensitive nodes. Other machine learning techniques have also been effectively applied to the CH selection problem. For example, Abdulzahra and Al-Qurabat used Fuzzy C-Means (FCM) to create well-defined clusters in a centralized structure, where the sink manages the clustering structure [27]. This represents a powerful and structured approach to network organization. In this work, we focus on a fully distributed solution, which addresses a distinct paradigm. Rather than relying on a central coordinator, our methodology is designed to remove this dependency by having each node make a fully autonomous decision, providing an alternative for scenarios where decentralized operation is the primary goal.

in [28], to select the optimal CHs, Wang et al. suggest a clustering approach by employing an enhanced artificial bee colony (ABC) algorithm [29]. According to Table 1, their study demonstrates better performance gains over other well-known meta-heuristic approaches like PSO [30] and GWO [31], demonstrating the efficacy of their improved optimization procedure. Their approach is based on a centralized paradigm, in which the sink finds a globally optimal solution by carrying out the optimization. This architectural choice is highly effective for global optimization but represents a different set of design trade-offs compared to decentralized systems. Our work explores an alternative path by focusing on a fully distributed fashion of decision-making. We designed our method specifically to operate without a central coordinator by using an offline-trained ANN to derive a lightweight predictive equation. This empowers each node to make autonomous decisions locally, offering a practical and scalable approach for scenarios where distributed autonomy is a primary design goal.

A significant portion of the literature, including the work by Wang et al. and others utilizing optimization algorithms like PSO, ABC, GWO etc., approaches CH selection as a global optimization problem [32–34]. While powerful in finding near-optimal solutions, these meta-heuristic algorithms often operate under a centralized paradigm that requires network-wide state information to be gathered at a

Table 1 Comparison with recent studies

Ref.	$Authors[NONSPACE]\&Year$	Used Energy Efficiency Method	Compared Methods	Number of nodes	Node Deployment	$Area[m^2]$	$E_{init}[J]$	$E_{elec}[nJ/bit]$	$\varepsilon_{amp}[p.J/bit/m^2]$	$packetsize[bits]$
[28]	Wang et al./2020	Global Artificial Bee Colony Algorithm Based on the Cross-over and Tabu Search (CGTABC)	LEACH-C, FIGWHO, PSO, and ABC-SD	variable (100, 200, 300, 500)	Random and Uniform	variable(100*100, 200*200, 300*300)	0.5	50	10	4000
[35]	Pour S.E. and Javidan R./2021	DRE-LEACH	LEACH, MR-LEACH, and EM-LEACH	100	Random and Uniform	70*70	0.5	50	10	4000
[36]	Nezha et al./2021	Energy-Aware Clustering and Efficient Cluster Head Selection (EAC-ECHS)	LEACH, Routing G5, and OSCH G6	100	Random	100*100	0.5	50	10	4000
[37]	Wu et al./2022	Dual Cluster-Head Energy-Efficient Algorithm (DCK-LEACH)	LEACH K-LEACH, IEECHS-WSN, and RCH-LEACH	variable (100, 200, 300)	Random	100*100	0.2	50	10	4000
[39]	Gulbas G. and Cetin G./2023	Simulated Annealing Algorithm (SA) and LEACH, (LEACH-SA)	LEACH, and PSCH-SA	100	Random	100*100	0.5	50	10	4000
[26]	Debasis et al./2023	ANN-Based Energy-Efcient Clustering Algorithm (EECA)	LEACH, ES-MAC, and EE-MAC	100	Random	100*100	5	Not defined	Not defined	800+160
Proposed	**Volkan et al.**	**Cascaded ANN**	**LEACH**	**100**	**Random**	**100*100**	**0.5**	**50**	**100**	**4000**

central entity. This can introduce communication overhead and scalability challenges in large or dynamic WSNs. In contrast, our work intentionally reframes the problem from one of global optimization to one of distributed, predictive decision-making. For this specific task, an ANN presents a more suitable and natural tool. Instead of searching for an optimal set of CHs across the network, our ANN is trained to act as a predictive model at the individual node level, allowing it to answer a critical local question: "Which selection threshold will maximize my own future residual energy?" This architectural choice is inherently distributed, predictive, and, through our two-stage methodology, exceptionally lightweight at runtime. Therefore, the inclusion of these optimization algorithms in our review serves to highlight the limitations of the centralized paradigm, which in turn motivates our choice of a distributed, ANN-based predictive approach.

In [35], Pour and Javidan introduced DRE-LEACH, a protocol that selects CHs using a score-based system that considers residual energy, node position, and centrality.In order to minimize the computational load, centrality and neighbor counts are calculated within a dynamic range that adapts to network density. While this represents an intelligent approach to localize calculations, the methodology still relies on a multi-step process of information exchange among neighbors at the beginning of each round to determine the highest-scoring node. This iterative local information exchange can introduce communication overhead and latency. This article, in contrast, aims to eliminate this round-by-round communication requirement. By using a pre-trained, equation-based model, each node makes a completely autonomous and non-communicative decision, which is a key distinction designed to minimize energy consumption associated with control packet exchanges.

Nezha et al.'s EAC-ECHS methodology in [36] divides the network into fixed grids to provide an organized strategy for clustering Table 1. This method effectively reduces energy consumption by using residual energy and distance metrics for CH selection within these pre-defined clusters. Although this grid-based structure offers stability, our adaptive approach has a different set of trade-offs. A rigid grid system may result in load imbalances or necessitate frequent re-clustering, which can impose control overhead, in situations including non-uniform node distributions or dynamic topology changes brought on by node failures. On the other hand, our work is built around a paradigm that is naturally flexible and unstructured. We let clustering develop naturally in each round based on the independent, predicted

choices made by the nodes, rather than pre-defining groupings based on geography. A dynamic and self-organizing clustering process results from each node determining its ideal participation probability using a lightweight equation. This strategy is designed to provide more adaptability and durability, especially in network environments that are unpredictable or irregular.

In [37], Wu et al. introduced the DCK-LEACH technique, which successfully extends network lifespan by combining Canopy and K-means optimization with an inventive dual CH approach. This method divides the CH duties between a main and a vice CH in order to handle load balancing. Compared to our approach in this study, this hierarchical, multi-layered approach has a distinct set of design trade-offs, especially in terms of implementation complexity and flexibility, notwithstanding its strength. The initial clustering may have an impact on the performance of a K-means-based method, and maintaining a dual-head hierarchy adds more operational complexity at the node or sink level. In our study, in contrast, is designed around the principles of simplicity and adaptability. Instead of introducing a complex, hierarchical clustering structure, we retain the fundamental simplicity of the LEACH protocol but enhance its core intelligence. Our approach relies on a single, lightweight predictive equation, enabling an emergent and self-organizing clustering process independent of complex multi-head management and sensitive initial conditions. With dynamic or unpredictable network environments in particular, this method is designed to provide a reliable and computationally efficient substitute.

Abdulzahra et al.'s work further advances the field by introducing EFUCSS, a comprehensive protocol that combines a sleep scheduling mechanism, a fuzzy logic controller for CH selection, and fuzzy C-Means for unequal clustering [38]. This multi-layered strategy shows how integrating multiple strategies can result in notable performance improvements. Our research focuses on the principle of simplicity and takes a different approach to the objective of increasing network lifetime. Our study aims to improve the fundamental intelligence of the original LEACH protocol itself, rather than adding several additional operational layers. This is an alternative design philosophy that aims to maximize efficiency by optimizing a single, basic mechanism instead of coordinating multiple mechanisms.

In [39], Gulbas and Cetin investigated how to optimize the CH selection procedure in LEACH using the Simulated Annealing (SA) algorithm. To improve network lifetime, their approach, LEACH-SA and PSCH-SA, efficiently use residual energy and distance metrics. This study demonstrates how meta-heuristic optimizers can be used to improve clustering choices. The computational effort needed for convergence in each round of such iterative optimization algorithms is a practical consideration.. Our work approaches this challenge from a different perspective, aiming to separate the complex learning process from the real-time decision-making. Instead of performing an online, iterative optimization on the nodes, proposed approach in this article utilizes a Cascade-ANN in an offline training phase to capture the optimal threshold-setting behavior. This learned intelligence is then deployed as a single, non-iterative mathematical equation. This design choice allows each node to make an autonomous and instantaneous decision without the computational and time overhead associated with iterative search algorithms, offering a methodology tailored for high-speed and energy-critical WSN deployments.

The reviewed literature highlights various powerful optimization strategies. Our study complements these works by focusing on a different aspect: enhancing the intelligence of the LEACH protocol at the individual node level. Instead of relying on centralized optimization or complex internode communication, we propose a method where each node uses a simple, predictive equation derived from an offline-trained ANN to autonomously determine its selection threshold. This methodology is tailored to address the scalability and overhead challenges noted in other approaches by promoting a fully distributed and computationally lightweight decision process, thus seeking to improve energy efficiency while maintaining the distributed nature of the original LEACH protocol.

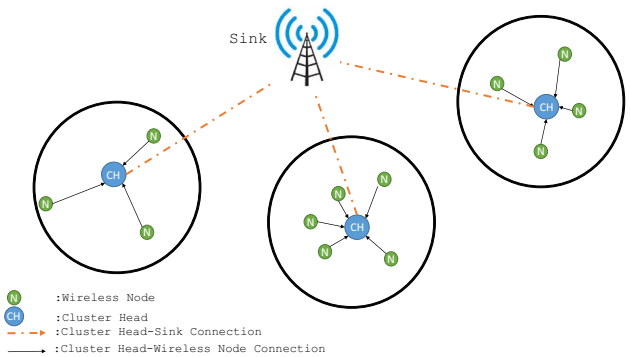


Fig. 2 A clustered wireless sensor network

3 The proposed threshold determination system

Figure 2 illustrates a simplified representation of a WSN. The sensor nodes (N) are organized into clusters, each led by a (CH). The CHs act as intermediaries, collecting data from their respective cluster members and transmitting it to the sink. This hierarchical structure helps distribute energy consumption as the responsibility of data transmission is shared between the nodes and the CHs. The sink serves as

the central point of control and data collection for the entire network. This arrangement allows for efficient data gathering and communication within the sensor network while also minimizing energy usage.

The ANNs are systems inspired by the brains of living beings, which are highly successful in solving complex problems that involve many interconnected nodes. In an ANN, weight coefficients are assigned to each link connecting the nodes, and these coefficients are adjusted using a dataset of known inputs and outputs, enabling the ANN to learn from this process. As a learning model, the accuracy of an ANN largely depends on the quality of the training data provided to it [40]. The objective of learning in ANN is to optimize the weights to make accurate predictions based on the given data. A specialized form, Cascade ANN, includes links from all of the previous layers, as illustrated in the block diagram in Fig. 3.

In the proposed method, nodes utilize a Cascade-ANN to determine potential threshold values. As depicted in Fig. 3 the Cascade-ANN had four inputs: the energy ratio (R), which represents the ratio of available energy to total energy of category value (CT) that the node belongs, the possible threshold value (T_i), the initial energy (E_{init}), and the category value (CT). Neuron numbers of hidden layers (i.e. Hidden-1, Hidden-2 and Hidden-3) are selected 5-5-5, 10-10-10 or 50-30-20 in simulation studies.

The primary rationale behind choosing the Cascade-ANN architecture as the neural network architecture in this study is to increase the model's learning capacity. By allowing each layer to directly access the outputs of previous layers and the original input variables, the cascade architecture is more effective than traditional feedforward networks in capturing the complex and nonlinear parameter interactions that determine node energy.

Determining the number of neurons per layer and the depth of layers is an empirical decision on the basis of model performance versus computational complexity. Various architectures (50-30-20, 10-10-10, and 5-5-5) were tested, and their performance is highlighted in Section 4. While the larger architectures (50-30-20) exhibit higher performance in terms of network lifetime, the memory and processing power requirements of these architectures are far from practical for a typical WSN node. For example, the 50-30-20 architecture requires approximately 1770 parameters (∼6.8 KB), while the 5-5-5 architecture requires only 170 parameters (∼0.68 KB). The 5-5-5 architecture offered a 10x improvement in resource consumption compared to 50-30-20, while offering a reasonable trade-off in performance (3497 rounds vs. 4457 rounds). Therefore, considering its low resource consumption and high applicability, the 5-5-5 Cascade-ANN structure was considered as an optimal balance point for distributed WSNs and was taken as the basis in this study.

To validate the practical feasibility of this approach on resource-constrained nodes, the following analysis details the memory and computational load for the 5-5-5 architecture, which serves as a representative baseline.

Memory Requirements: Our model, which has a Cascaded-ANN architecture with 4 inputs, 3 hidden layers (each with 5 neurons), and 1 output, consists of a total of 170 parameters (weights and biases). Assuming a standard 32-bit floating-point representation, the total memory required to hold the model is approximated at 680 bytes. This storage requirement is considered quite manageable for common WSN platforms such as the TelosB (MSP430), which typically have 48 KB of Flash memory. The model is expected to occupy only approximately 1.4% of this platform's program memory.

Computational load: A single feedforward pass for threshold estimation requires approximately 154 Multiply-Add (MAC) operations, which is approximately ∼500 Floating-Point Operations (FLOPs). Even on a low-power

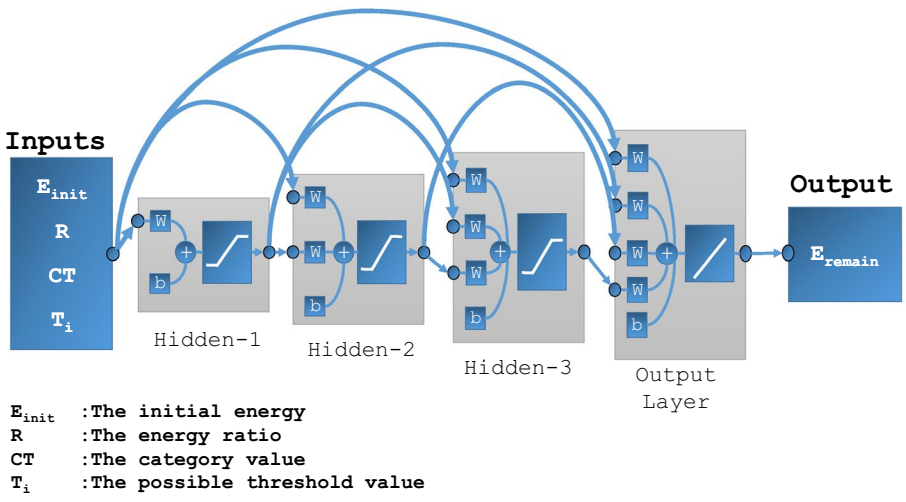


Fig. 3 Cascade-ANN block diagram used in the study

microcontroller such as the MSP430, which lacks a dedicated Floating-Point Unit (FPU), software-based libraries are expected to finish this calculation in milliseconds ($\sim$ 10–20 ms). Since our protocol only performs this operation once every 100 rounds, it should have very little effect on the node's overall energy budget. These calculations demonstrate that direct implementation of the 5-5-5 ANN model on a commodity platform like TelosB is practical and effective. Also, our approach offers a second, even lighter alternative for extremely resource-constrained nodes. As detailed in Section 4.1 of the article, the learned behavior of the ANN is converted into a simple mathematical equation via curve fitting. This equation-based approach completely eliminates the need to run the ANN on the node, reducing memory requirements to a few tens of bytes and computational overhead to a handful of basic arithmetic operations. This two-stage strategy (a feasible ANN and an ultra-lightweight equation-based alternative) demonstrates the potential of our methodology to be a practical solution applicable to a wide range of WSN hardware, from mainstream platforms like TelosB to more constrained devices like MICAz, rather than simply a theoretical assumption. Having established the practical feasibility of deploying the proposed ANN on a sensor node, the operational details of the method can now be described.

The energy ratio (R) for a node is calculated using Equation 2, where $E_{m,\text{current}}$ and $E_{n,\text{current}}$ represent the current energy levels of the mth node and the nth node in the same CT, respectively. The output of the Cascade-ANN was the remaining energy (E_{remain}) of the wireless sensor node.

Furthermore, in the proposed method, CH selection takes place after 100 rounds instead of just one round. Here, one round signifies that all nodes communicate with the CH only once.

To generate the dataset, a network of 100 nodes was used, with a sink located at the center of a 100x100 square meter area. The initial energy levels (E_{init}) and threshold values (T) of each node were randomly assigned. The maximum initial energy value was set to range between 0.05 and 0.5, with increments of 0.05. The $\boldsymbol{E}_{\text{initial}}$ shown in Eq. 1 includes possible maximum initial energy values.

We started by selecting E_0 as the possible initial energy of the nodes, and then used the formula $E_{\text{init}} = E_0 \times \text{rand}$ to assign the E_{init} value for each node. Here, rand is a uniformly distributed random variable with values between 0 and 1. After 100 rounds of CH selection, the remaining energy values were stored and the nodes' energy levels were reset to $E_{\text{init}} = E_0 \times \text{rand}$. This entire process was repeated 200 times. We then replaced E_0 with E_1, and repeated the initial energy assignment process using the formula $E_{\text{init}} = E_j \times \text{rand}$, where j ranges from 0 to $g-1$. Algorithm 1 depicts the dataset creation process.

$$\boldsymbol{E}_{\text{initial}} = [\ E_0 \quad E_1 \quad \ldots \quad E_{g-1}\ ] \tag{1}$$

function CREATEDATASET($\mathbf{x}, \mathbf{y}, x_s, y_s, \boldsymbol{E}_{\text{initial}}, p_{ch}, p_{trial}$)
  $\mathbf{x}, \mathbf{y}$ - Coordinate arrays for the nodes
  Assign category as CT to all nodes ▷ Based on distance to sink (See Table 2)
  $x_s = 50, y_s = 50$ ▷ Coordinates of the sink
  $p_{ch} = 100$ ▷ CH election period
  $p_{trial} = 200$ ▷ Number of trials for each energy level
  for $i = 0$ to $g-1$ **do** ▷ Number of elements of $\boldsymbol{E}_{\text{initial}}$ is g
    $E_i = \boldsymbol{E}_{\text{initial}}\ (i)$
    for $k = 0$ to p_{trial} **do**
      for $m = 0$ to $N-1$ **do** ▷ N is the number of nodes
        Assign energy to m^{th} node as $E_{\text{init}} = \text{rand} \times E_i$
        Assign threshold to m^{th} node as $T = \text{rand}$ ▷ A possible threshold value T_i
      end for
      for $q = 0$ to p_{ch} **do**
        for $m = 0$ to $N-1$ **do**
          if rand $\leq T$ **then**
            Elect m^{th} node as CH
          end if
        end for
        Assign nearest CH to non-cluster head nodes
        Transmit and receive packets
      end for
      Calculate R in Equation 2 ▷ Energy ratio
      Store E_{init}, T, CT, E_{remain}, and R of nodes into dataset
    end for
  end for
end function

Algorithm 1 Algorithm for Creating Dataset

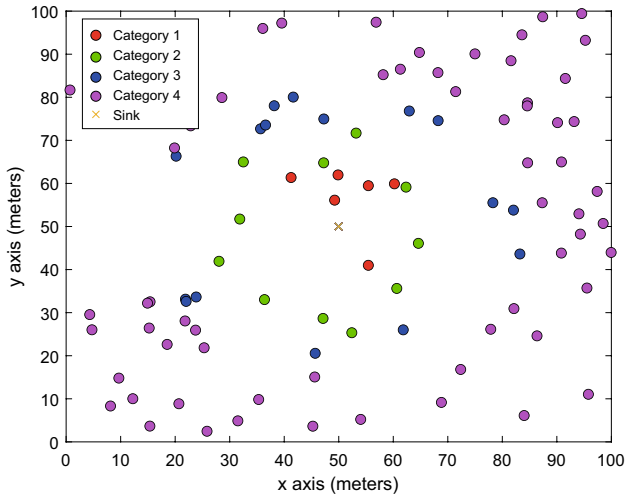


Fig. 4 Representation of the four distance categories in a 100x100 meter area

Table 2 Categories of the nodes based on distance to the Sink

Distance to the Sink (u in meters)	Category
$u < 15$	1
$15 \leq u < 25$	2
$25 \leq u < 35$	3
$u \geq 35$	4

The data generation methodology described in Algorithm 1 was designed with a specific aim of fostering diversity of data and improving generalizability of the model. Diversity of data is achieved through two main mechanisms: (1) by randomly assigning primary parameters such as the initial energy (E_{init}) and threshold (T) to the nodes for each trial, over a wide range and drawn from a uniform distribution, and (2) by systematically changing initial levels of energy. This prevents the model from overfitting to a single initial scenario. Furthermore, a new network is created each time by randomly distributing nodes throughout the environment. It is also important to note that the "category" (CT) variable is an input feature, not a class label. Because the output of our model (E_{remain}) is a continuous value, this is a regression problem. Combining this high-variance data generation process with a large sample size of 200,000 provides a solid foundation for the model's generalization. Robust validation metrics, like the high R value and low Mean Square Error (MSE) shown in Figs. 5 and 6, further support the model's strong generalization performance.

The nodes were categorized into four groups based on their distance from the sink: nodes within 15 meters were labeled as $\mathrm{CT} = 1$, nodes situated between 15 and 25 meters as $\mathrm{CT} = 2$, nodes positioned between 25 and 35 meters as $\mathrm{CT} = 3$, and nodes located beyond 35 meters as $\mathrm{CT} = 4$. Following each trial, data for variables such as E_{init}, E_{current}, T, CT, and R were collected and stored for the purpose of Cascade-ANN training. Additionally, the remaining energy of each node (E_{remain}) was recorded as the output for the Cascade-ANN training. It was assumed that all nodes, including the sink, remained stationary. In total, a dataset comprising 200,000 data points was created.

In this study, a Cascade-ANN was used to predict the remaining energy of a wireless sensor node in a wireless sensor network. In a Cascade-ANN, there are connections from the input layer to each subsequent layer. Using the MATLAB Neural Network Toolbox, the Cascade-ANN was trained on a dataset of 200,000 samples.

$$R_m = \frac{E_{m,\mathrm{current}}}{\sum_{n=0}^{M-1} E_{n,\mathrm{current}}} \tag{2}$$

In the proposed method, a first-order radio model is used for the energy consumption simulations of the nodes [11]. Accordingly, the amount of energy to transmit a k-bit packet to a distance d is as in Eq. 3 and the amount of energy to receive a k-bit packet is as in Eq. 4.

$$E_{Tx}(k,d) = E_{\mathrm{elec}} * k + \epsilon_{amp} * k * d^2 \tag{3}$$

$$E_{Rx}(k) = E_{\mathrm{elec}} * k \tag{4}$$

The energy required to process a single bit in the receiver and transmitter circuits is given as E_{elec}. Meanwhile, the energy coefficient for the amplifier is represented by ϵ_{amp}.

Figure 4 illustrates the nodes belonging to the four main categories and their locations in the network. In the figure, the colored dots indicate the nodes belong to specified categories, while 'x' marks the sink, positioned within a 100x100 square meter area. Additionally, Table 2 shows the categorization of the nodes based on their distance from the sink. The output of the Cascade-ANN is the remaining energy of the node if the node uses a possible threshold value $T_i \in \boldsymbol{Tp}$ selected from the threshold value array $\boldsymbol{Tp}$. Equation 5 shows the threshold value array, which is a row vector consisting of the threshold values T_0 through T_{m-1}.

$$\boldsymbol{T_p} = [\ T_0 \quad T_1 \quad \cdots \quad T_{m-1}\ ] \tag{5}$$

Since the neural network output is the E_{remain}, the index of the element in the threshold values array ($\boldsymbol{T_p}$) that corresponds to possible T_i values yielding the largest E_{remain} can be found by trying all possible threshold values one by one.

$$\mathrm{T} = \underset{T_i \in \boldsymbol{T_p}}{\arg max} E_{\mathrm{remain}}; i = 0, 1, \ldots, m-1 \tag{6}$$

Table 3 Simulation Parameters

Parameter	Value
Area	100x100 m^2
Number of nodes	100
Data packet size	2000 bits
Initial energy	0.5 J
E_{elec}	50 nJ/bit
E_{DA}	5 nJ/bit/packet
ϵ_{amp}	100 pJ/bit/m^2

It is presumed that each node has its own ANN and $\boldsymbol{T_p}$ saved in memory, allowing it to determine the optimal threshold value to use. After 100 rounds, each node repeats the process of utilizing its Cascade-ANN to identify the most appropriate threshold value to use.

4 Simulation results

In light of the resource-constrained nature of WSNs, the ANN architectures tested in this study (5-5-5, 10-10-10, and 50-30-20) were chosen. Our primary objective was to analyze the trade-off between model complexity and network performance. Therefore, three distinct structures were chosen as low, medium, and high complexity. The architecture's three hidden layers were motivated by deep learning principles, which theoretically offer the potential to learn complex relationships with fewer neurons than single-layer networks. This strategy seeks to improve the model's learning ability while controlling resource usage.

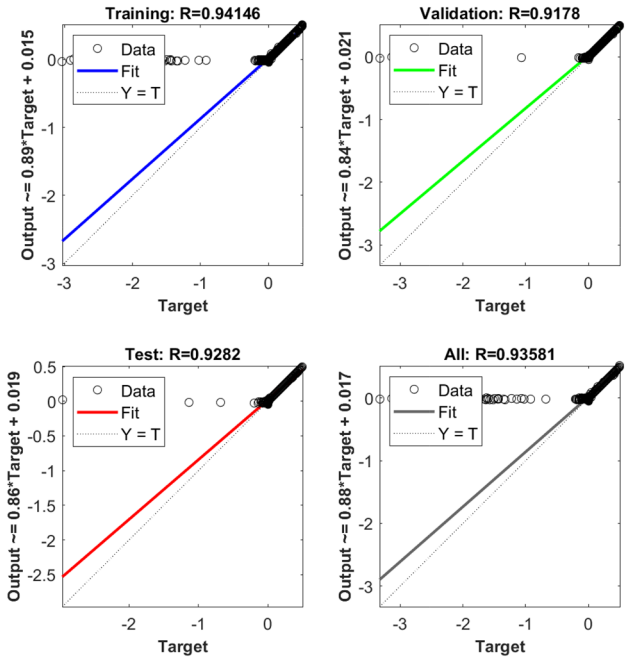
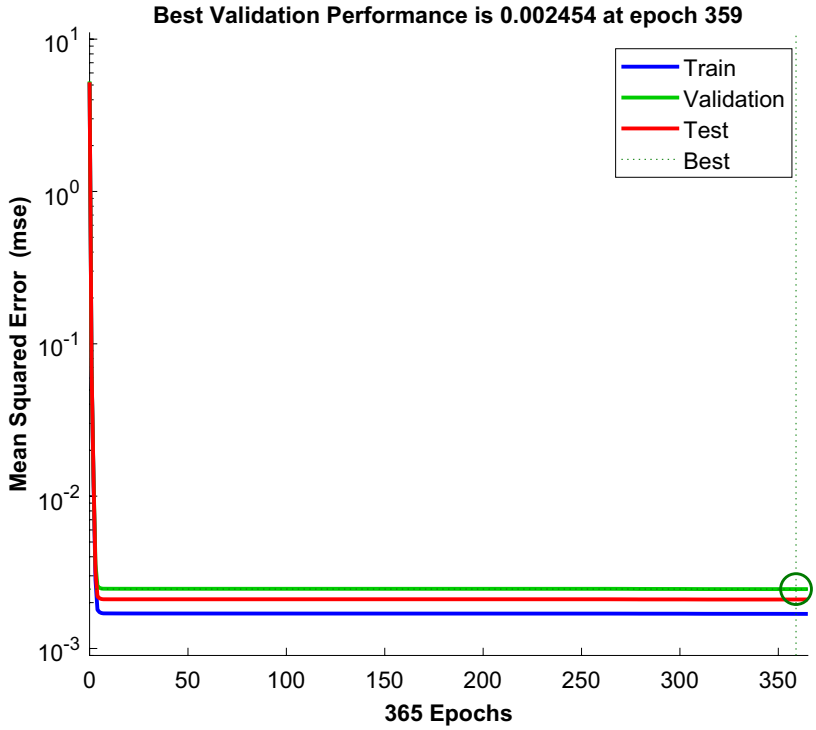


Fig. 6 Regression plot of the trained network

To evaluate the performance of these architectures, a series of simulations were conducted under a consistent set of parameters.

The simulations were performed using MATLAB. The packets had a size of 2000 bits. It was assumed that

Fig. 5 Validation Performance



the nodes have knowledge about potential thresholds, initial energy levels, and category information for CH selection. The receiver and transmitter electronic circuits were assumed to use 50 nJ of energy per bit for processing ($E_{\text{elec}} = 50nJ/bit$). The amplifier energy coefficient, ϵ_{amp}, was set at $100pJ/bit/m^2$ and energy for data aggregation E_{DA} was set at 5*nJ/bit/packet*. Other parameters are shown in Table 3.

The proposed method was simulated using the algorithm described in Algorithm 2.

Load the trained cascade ANN
$\mathbf{x}, \mathbf{y}$ ▷ Coordinate arrays for the nodes
Assign category as CT to all nodes ▷ Based on distance to sink (See Table 2)
$x_s = 50$, $y_s = 50$ ▷ Coordinates of the sink
$p_{ch} = 100$ ▷ CH election period
Assign each node $E_{\text{init}} = 0.5$ joules initial energy
Load threshold value array $\boldsymbol{T_p}$ ▷ Equation 5
Calculate total energy of categories
Calculate ratio values for the nodes ▷ Equation 2
for $q = 0$ to $N - 1$ **do** ▷ N is the number of nodes
  $R = R_q$ ▷ Energy ratio for node q
  for $k = 0$ to $m - 1$ **do** ▷ m is the number of possible threshold values
    $T = T_k$
    Use $E_{\text{init}}, R, \text{CT}, \text{T}$ as input to ANN and get E_{remain} as output from ANN
  end for
  Find the index $k_{\max} \in k$ that gives maximum E_{remain} for node q
  Set threshold value of node q as $T = T_{k_{\max}}$
end for
for $q = 0$ to $N - 1$ **do**
  if rand $< T$ **then**
    Node q becomes cluster head
  else
    Node q becomes non-cluster head
  end if
end for
Assign nearest cluster head to non-cluster head nodes
while Total energy of the nodes ≥ 0 **do**
  for $k = 0$ to $p_{ch} - 1$ **do**
    for $q = 0$ to $N - 1$ **do** ▷ N is the number of nodes
      Transmit and receive packets
      Reduce energy for transmitting and receiving from all nodes
    end for
  end for
  Calculate ratio values for the nodes ▷ Equation 2
  for $q = 0$ to $N - 1$ **do** ▷ N is the number of nodes
    $R = R_q$
    for $k = 0$ to $m - 1$ **do** ▷ m is the number of possible threshold values
      $T = T_k$
      Use $E_{\text{init}}, R, \text{CT}, \text{T}$ as input to ANN and get E_{remain} from ANN
    end for
    Find the index $k_{\max} \in k$ that gives maximum E_{remain} for node q
    Set threshold value of node q as $T = T_{k_{\max}}$
  end for
  for $q = 0$ to $N - 1$ **do**
    if rand $< T$ **then**
      Node q becomes CH
    else
      Node q becomes non-cluster head
    end if
  end for
  Assign nearest cluster head to non-cluster head nodes
end while

Algorithm 2 Algorithm of the Proposed Method

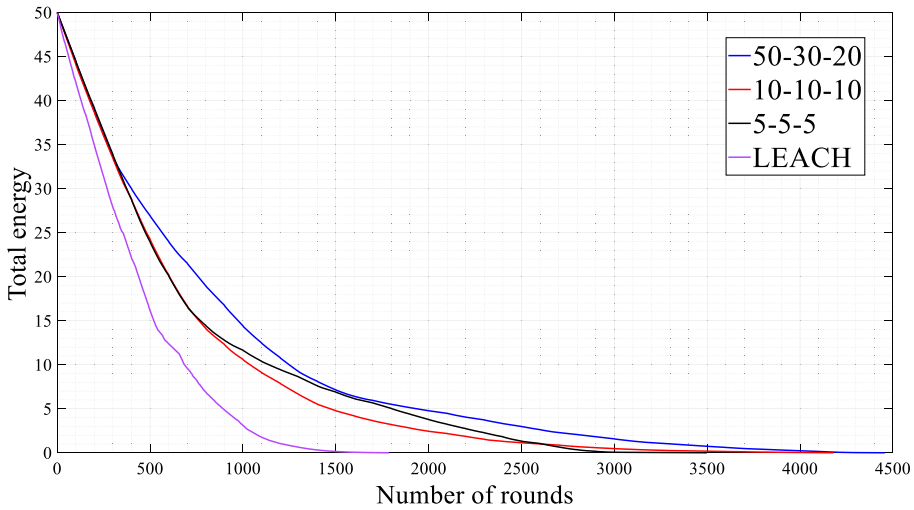


Fig. 7 Variation of the remaining energy according to the number of rounds

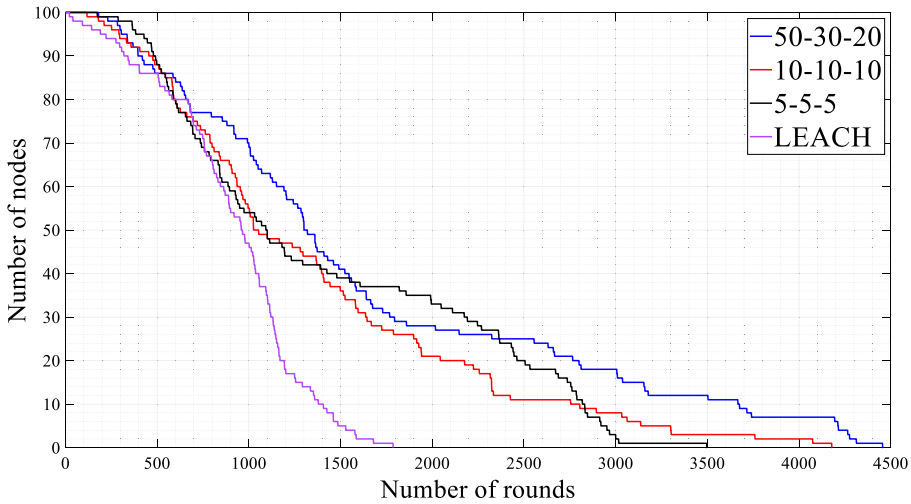


Fig. 8 Variation of the number of live nodes with respect to the number of rounds

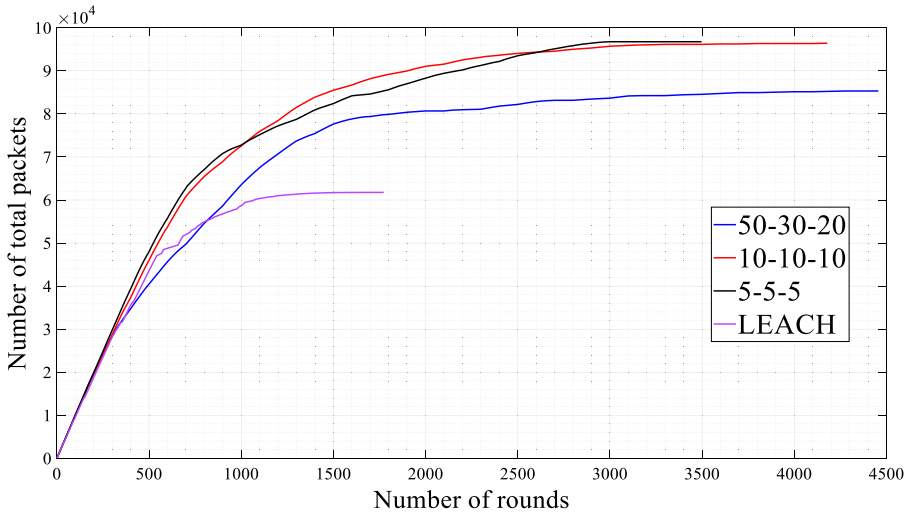


Fig. 9 Variation of the total number of packets with respect to the number of rounds

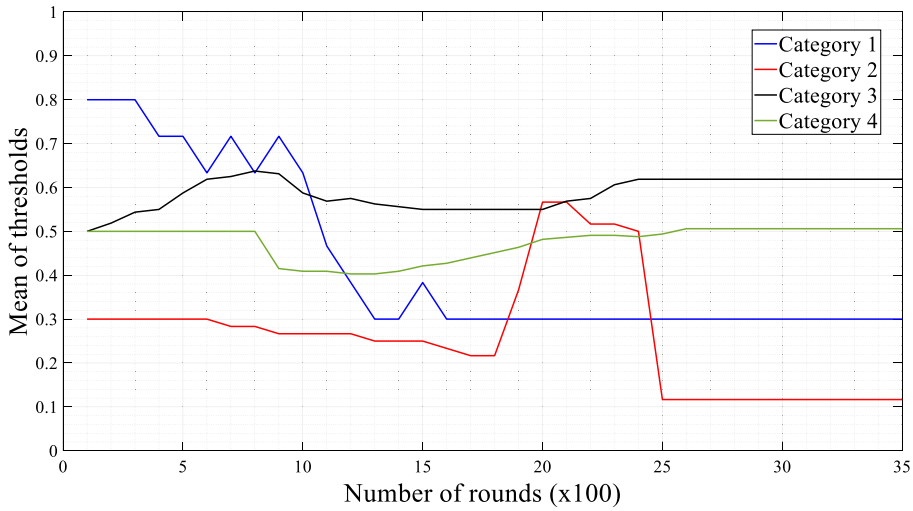


Fig. 10 Mean of thresholds

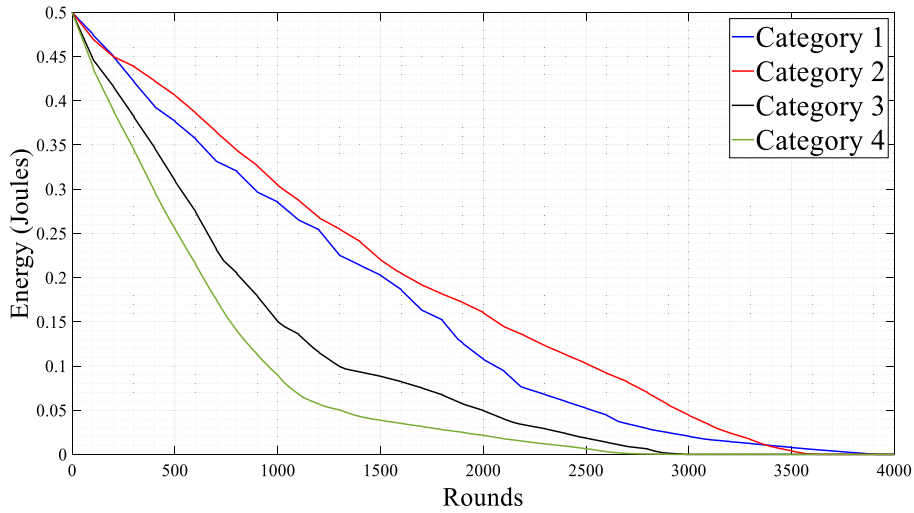


Fig. 11 Energy consumption of categories

Figures 5 and 6. depict the training performance of the network when each hidden layer contains 5 neurons. According to Fig. 5, the network was trained for 359 epochs, resulting in the best validation performance with a mean squared error (MSE) value of 0.002454. Additionally, upon analyzing the regression plot in Fig. 6, it is evident that there exists a correlation coefficient (R) of at least 0.9178 between the data and the network's output.

We applied the proposed method to a Wireless Sensor Network (WSN), using node locations shown in Figure 4. and leveraging the trained cascade neural network. The performance of the proposed method is illustrated in Figs. 7, 8, and 9.

In Fig. 7, the figure compares the total energy consumption of various Cascade-ANN approaches with different node configurations in the hidden layers to the LEACH protocol across multiple rounds. It is observed that while LEACH depletes its total energy by round 1500, the Cascade-ANN approach with 50-30-20 neurons in the hidden layers sustains its total energy until approximately round 4500. This demonstrates that the total energy of the proposed method consistently surpasses that of LEACH, from the initial rounds through to the final round. Different hidden layer configurations for Cascade-ANN are considered to observe their performance effects on the proposed method.

Fig. 8 compares the number of operational nodes over time (in terms of rounds) for different configurations of Cascade-ANN (with 50-30-20, 10-10-10, and 5-5-5 neurons in the hidden layers) and the LEACH protocol. The figure demonstrates that networks with a larger number of neurons in the hidden layers (50-30-20) can sustain node operation for nearly three times longer than LEACH. Consequently, the objective of maximizing residual energy (E_{remain}) is more effectively achieved by using hidden layers with more neurons. Interestingly, around 40% of the nodes survive up to approximately the 1500th round, regardless of the number of neurons in the hidden layers.

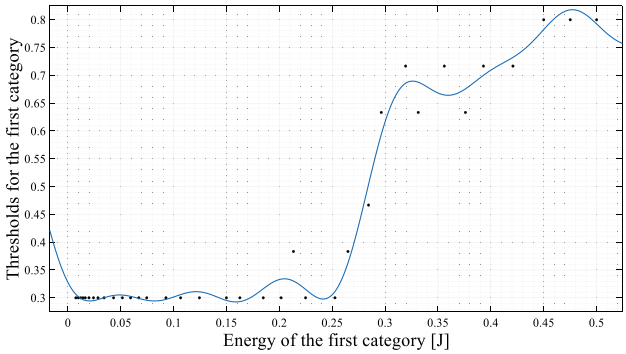


Fig. 12 Threshold-energy curve of the 1st category

The ANN with 50-30-20 neurons in hidden layers, survived 4457 rounds and transmitted a total of 85318 packets to the sink in Fig. 9. The ANN trained with 10-10-10 neurons survived for 4180 rounds and transmitted a total of 96420 packets to the sink. The ANN trained with 5-5-5 neurons survived 3497 rounds and transmitted a total of 96757 packets to the sink. The LEACH algorithm survived for 1773 rounds under the same conditions and transmitted a total of 61766 packets to the sink. Accordingly, the variations of the proposed method exceeded the LEACH algorithm both in terms of total packets and network lifetime. However, given that the nodes have limited processing power, it is deemed more crucial to select the scenario with three hidden layers having five neurons, as it demands less processing power.

In addition, when analyzing the average thresholds, it was observed that the average threshold of the nodes in the category closest to the sink, which initially started relatively high (0.8), decreased over time (as shown in Fig. 10). This behavior can be interpreted as these nodes conserving their energy for a certain period because of their proximity to the sink. However, as their energy levels decrease, they adjust their threshold averages downward to prolong the network's lifetime.

On the other hand, nodes in Category 2 initially have the lowest average threshold values and maintain this characteristic for most of the network's operational lifespan. These lower threshold values reduce the likelihood of Category 2 nodes becoming CHs. It's important to note that there is a provision for increasing the threshold values of Category 2 nodes when the energy levels of Category 1 nodes fall below a certain threshold.

The average threshold for Categories 3 and 4 has not changed significantly. However, these categories are located the farthest from the sink, which means that the CHs in these categories consume more energy compared to the other categories. Traditionally, the solution would be to reduce the thresholds for energy efficiency. But if nodes in Categories 3 and 4 cannot find a closer CH, then their energy consumption while communicating with the CH could become even higher.

Table 4 Performance metrics of curve fitting functions

	T_1	T_2	T_3	T_4
SSE	0.01797	0.05964	0.0009086	0.005413
R-Square	0.9857	0.9084	0.9771	0.8784
Corrected R-Square	0.9762	0.8218	0.9631	0.853
RMSE	0.02926	0.05756	0.007105	0.01502

Figure 11 illustrates the energy consumption of nodes in different categories. It can be observed from Fig. 11 that nodes in all categories have a lifespan exceeding 2500 rounds.

In another analysis, the performance of ANN trained with fixed node positions was compared to random placement of nodes in the same area during training. The nodes were randomly positioned within a 100x100 meter square area each time, and a neural network with three hidden layers of 5 neurons each (5-5-5) was used, assuming the threshold value at each node was determined as in the proposed method. After 10 trials, the average network lifetime was 3670 rounds, and the average number of packets reaching the sink was 71,006. These results indicate that while the performance decreases slightly when the node positions differ from those used during ANN training, Based on the simulation-based comparative analysis, the proposed approach achieves approximately twice the network lifetime of LEACH and about a 15% increase in the number of delivered packets under the considered simulation conditions, suggesting its potential to enhance network longevity in similar scenarios.

4.1 Findings on curve fitting using ANN data

In another study, the network lifetime and threshold value change using ANN data were obtained by using the curve fitting function in the MATLAB environment. The change between the energy and threshold value of each category was plotted and the equation that best fits this curve was taken into account. In more clear terms, four equations related to the remaining energies of four categories were assigned. For example, when the current energy of the node belonging to the first category is entered into the equation, the calculation result shows which threshold value that node should produce. This situation is noteworthy as it will eliminate the necessity of storing the trained neural network in the memory of each node. The results obtained with the curve fitting were evaluated by comparing with the original ANN.

Accordingly, the threshold-energy relationship of the nodes belonging to the first category is given in Fig. 12 in the light of the data obtained with ANN. These data were tested with MATLAB Curve Fitting Toolbox and the additive sine, which gave the best result from the regression

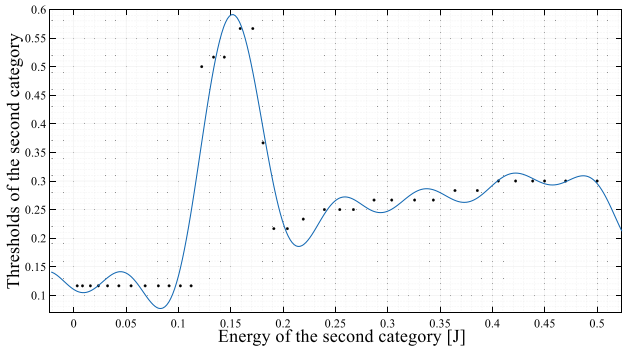


Fig. 13 Threshold-energy curve of the 2nd category

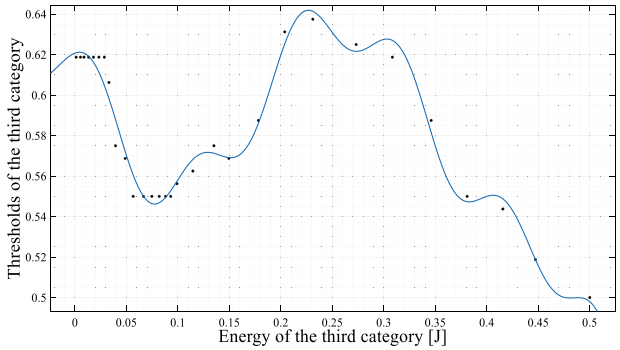


Fig. 14 Threshold-energy curve of the 3rd category

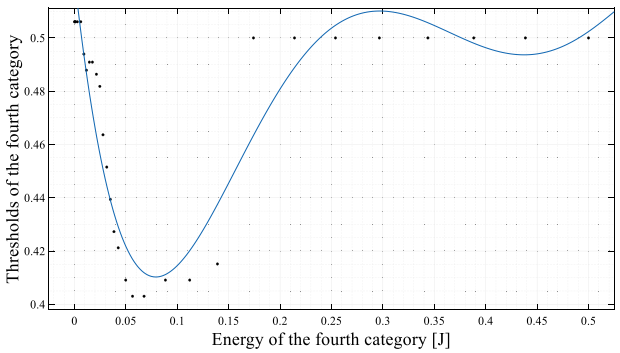


Fig. 15 Threshold-energy curve of the 4th category

models, was used. The corresponding expression is given in Eq. 7.

$$\begin{aligned} T_1 = {} & 6.923 \cdot \sin(5.441 \cdot E + 0.777) + 6.654 \\ & \cdot \sin(5.822 \cdot E + 3.873) + 0.039 \\ & \cdot \sin(50.89 \cdot E - 1.981) + 0.022 \\ & \cdot \sin(70.58 \cdot E - 1.008) + 0.061 \\ & \cdot \sin(32.26 \cdot E + 3.599) \end{aligned} \tag{7}$$

The apparent complexity of this additive sine model warrants further explanation. The objective here is not to fit a model to noisy experimental data in the traditional sense, but rather to represent, as accurately as possible, the deterministic and non-linear relationship learned by the trained ANN. It was observed that simpler models were unable to provide this representation adequately.

In our model selection process, various models were evaluated using the MATLAB Curve Fitting Toolbox, and the selection was made from among the models that offered the best goodness-of-fit metrics. The results presented in Table 4 indicate that the chosen models exhibit a reasonable fit with the ANN output data.

It is important to note that the ultimate evaluation of whether this approach leads to overfitting is based on its simulation performance. The results presented in Figs. 16 and 17 show that the network performance obtained with the equations is close to that of the original ANN model. This finding suggests that the derived equations are promising for capturing the general behavior of the ANN. Indeed, the logical and explainable behavior captured by these equations can be seen in the analysis of the first category. The first category includes the nodes in the category closest to the BS. Therefore, at low energy values (i.e. 0.25 J and below), the threshold values are around 0.3–0.35.3.35. If the energy of the node is above 0.3 J, the threshold value increases to 0.65–0.8.65.8. When the energy is high, more CHs are elected from the nodes, and when the energy is low, fewer CHs are elected from the nodes, which is a logical and explainable situation. Similarly, using the data of the ANN-supported CH election method, the threshold-energy curve for the second category nodes is given in Fig. 13. The threshold-energy equation of the second category nodes is obtained with the additive sine as in Eq. 8.

$$\begin{aligned} T_2 = {} & 1.546 \cdot \sin(5.346 \cdot E - 0.809) + 0.09 \\ & \cdot \sin(28.84 \cdot E - 3.281) + 1.275 \\ & \cdot \sin(6.488 \cdot E + 1.904) + 0.092 \\ & \cdot \sin(40.13 \cdot E + 2.064) \\ & + 0.059 \cdot \sin(56.36 \cdot E - 0.771) \\ & + 0.0426 \cdot \sin(69.14 \cdot E - 2.428) \end{aligned} \tag{8}$$

In the nodes in the second category, when the energy level was between 0.5 and 0.2 J, the threshold values gradually decreased from 0.3 to 0.2. When the energy level was between 0.2 and 0.1 J, there was an increase in the threshold values up to 0.55. Since the gradual decrease in the threshold values made the CH selection difficult, this sudden increase can be considered to facilitate the CH selection. The threshold-energy graph obtained when the same operations were applied to the nodes in the third category is given in Fig. 14 below. The regression model that best fits this data is again the additive sine and is as shown in Equation 9.

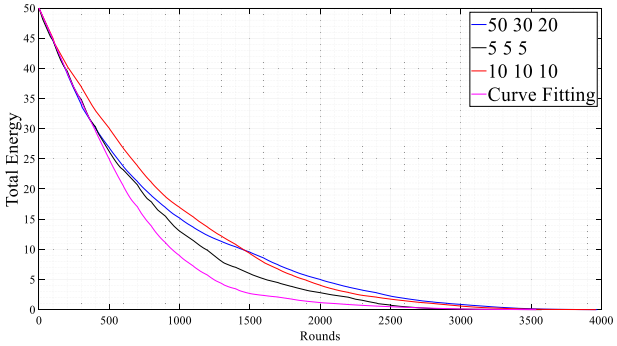


Fig. 16 Comparison of ANN-assisted routing method and curve fitting method in terms of total energy-round count

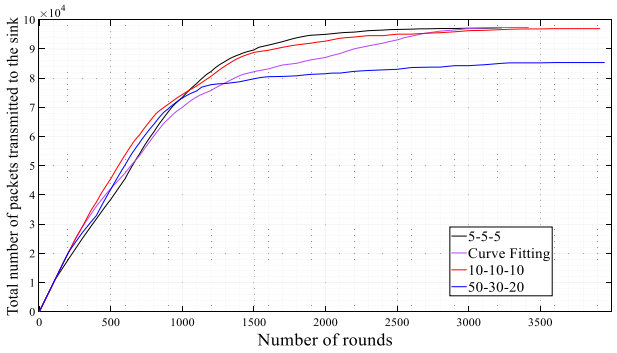


Fig. 17 Comparison of ANN-assisted routing method and curve fitting method in terms of total packet count

$$T_3 = 1.016 \cdot \sin(5.645 \cdot E + 0.274) + 1.057 \cdot \sin(11.83 \cdot E + 1.897) + 0.677 \cdot \sin(13.91 \cdot E + 4.486) + 0.013 \cdot \sin(63.29 \cdot E + 0.326) \quad (9)$$

The threshold-energy graph for the fourth category nodes is given in Fig. 10 below. When the relevant figure is examined, it is seen that the threshold values are at the level of 0.5 when the energy of the drops is approximately 0.2 Joule and above. However, when the energy level drops to the level of 0.1 J, it is seen that the threshold also drops to the level of 0.41. As the energy drops even further, the threshold gradually increases. It should not be forgotten that the fourth category consists of the nodes farthest from the BS. Therefore, the fact that the nodes are especially far away significantly increases the energy consumption of the CH, so the threshold decreases to create fewer CHs. The most suitable regression model for this data is the polynomial model, unlike the other three models. However, since the threshold level reaches 1 when the energy level is very small, the maximum that the equation can give is limited to 0.9.

The energy threshold relationship corresponding to the fourth category is given below in Eq. 10. The equations defined in the expressions T_1, T_2, T_3 and T_4 represent the threshold values of the 4 categories and the E value in the equations represents the current energy level of the relevant node.

$$T_4 = -113.8 \cdot E^5 + 199.3 \cdot E^4 - 126.4 \cdot E^3 + 34.37 \cdot E^2 - 3.441 \cdot E + 0.5225 \quad (10)$$

Performance metrics of the curve fitting functions are given in Table 4.

A wide range of curve fitting models, including linear, polynomial, and sinusoidal, was evaluated separately for each category. The model selection criteria were based on the Adjusted R-Square, which also accounts for model complexity, and the Root Mean Square Error (RMSE) metrics. Simpler models were also evaluate in this process but were inadequate to capture the nonlinear, complex nature of the ANN output with high accuracy. The final selection was made based on the model that was determined to provide the best fit on these metrics for each category. As a result of this process, additive sine models were found to be more suitable for the first three categories (T1–T3), while a polynomial model was found to be more suitable for the fourth category (T4). Therefore, the energy threshold equations for the four categories were determined by curve fitting using ANN data. Thanks to these equations, any node calculates the threshold value according to the energy and category value through the equation.

The simulation was repeated under the same conditions using the four equations given above and the results were compared with the ANN model. Accordingly, the energy-round number obtained by curve fitting is given in Fig. 16.

When the total energy-round number change obtained with curve fitting is examined, it is seen that the nodes lose energy faster compared to the ANN approach using 5 neurons in each hidden layer. For example, the decrease in total energy becomes apparent after 500 rounds. However, the network using the relevant method still managed to survive for 3000 rounds. Therefore, both the low processing load for CH selection and the fact that it does not require a central process can be seen as advantages of this method. The comparison of the packet number findings of the ANN CH selection method with the packet number findings of the curve fitting method is given in Fig. 17. Accordingly, the total number of packets transmitted to the BS via the curve fitting method is 97372, while the 5-5-5 neuron ANN is 97246. In addition, although the network lifetime of the 10-10-10 and 50-30-20 neuron ANN methods is longer, they produced worse results in terms of the total number of packets transmitted to the BS. This is because the optimization was done according to the network lifetime. In the light of these results, the curve fitting method was able to produce very positive results in terms of both the

number of packets and the network lifetime. The practical implementation of our proposed CH selection mechanism is designed to be adaptable for resource-constrained Wireless Sensor Network (WSN) nodes. While the 5-5-5 Cascade-ANN model is significantly more efficient than larger architectures, its cumulative computational load from multiple runs and memory footprint can be challenging for some devices. To overcome this, our methodology includes a lightweight, equation-based alternative derived from the ANN. This approach replaces the iterative runs with a single calculation, drastically reducing memory and computational load. This adaptability ensures our proposed mechanism is a practical and scalable solution suitable for a wide range of WSN hardware.

To better understand the practical implications of this equation-based alternative, a brief analysis of the memory and computational load of the derived equations was conducted. This analysis helps to evaluate whether even the most complex of these equations could be considered suitable for resource-constrained nodes. The provided additive sine models were selected due to their high fidelity in simulating the ANN output. Although the used trigonometric functions are computationally more costly than basic arithmetic, this computation is performed extremely infrequently in our protocol, once every 100 rounds, and so the impact of this additional overhead on the overall energy budget can be termed as negligible. Furthermore, this study demonstrates that simpler alternatives can be employed to approximate the ANN's behavior with sufficient accuracy. These include straightforward threshold equations that rely solely on basic arithmetic (such as the polynomial model derived for Category 4) or the use of lookup tables. Such simplified approaches can be executed on even the most resource-constrained nodes at a computationally trivial cost. The proposed equation-based methods are substantially more efficient in terms of memory than the full ANN model.

As an example, the 5-term sine equation (Equation 7), which is the most complex, only requires 60 bytes of storage, and the 5th-order polynomial equation (Equation 10) only needs 24 bytes. This is more than a 10x reduction in memory compared to the ANN model, which required 680 bytes of memory. When comparing computational loads, the polynomial equation has a very low cost of ∼10 FLOPs, and even the cost of the sine-based equation (∼1300 FLOPs) can be executed in milliseconds on a low-power microcontroller. Even though this occurs once every 100 rounds, these events will incur a negligible impact on the node's overall energy budget.

4.2 Comparative analyses

The performance of the proposed method has been compared with the LEACH protocol and other important approaches in the literature. To compare the performance of the studies on a consistent and fair basis, all algorithms were compared in a common simulation environment with the same underlying parameters. The algorithmic logic of the studies that are being compared was implemented in our own codebase according to the specifications outlined within the respective publications. To further enhance the fairness of the comparative analysis, a normalized methodology was followed instead of absolute performance values. This approach evaluates the performance of each algorithm relative to the standard LEACH protocol running under the specific simulation conditions of that algorithm. The process operates as follows:

For each study to be compared, the original simulation parameters listed in Table 5 were applied to our own environment. Under these parameters, both the standard LEACH protocol and our own proposed method (PM) were run. Finally, the performance of our own method was ratioed against the base performance of LEACH running under the same conditions to calculate an Improvement Ratio (IRoL for PM). The performance comparison was made based on these relative improvement ratios.

The purpose of our adoption of this methodology is to try to show that the observed performance differences are largely due to algorithmic improvements, rather than external factors such as different simulation environments or radio models.

As summarized in Table 5, the results indicate that the proposed method shows a consistent performance improvement

Table 5 Performance Analysis of the Proposed Study Between the Recent Studies

Ref.	*Authors*	NoN	ND	*Area* $[m^2]$	E_{init} $[J]$	E_{elec} $[nJ/bit]$	E_{amp} $[pJ/bit/m^2]$	*packetsize* [*bits*]	*IRoL* [%]	*IRoLforPM* [%]
[28]	Wang et al.	100	R	100*100	0.5	50	10	4000	85.9	**313.2**
[35]	Pour S.E. and Javidan R.	100	R	70*70	0.5	50	10	4000	32.9	**288**
[36]	Nezha et al.	100	R	100*100	0.5	50	10	4000	50	**313.2**
[37]	Wu et al.	100	R	100*100	0.2	50	10	4000	69.3	**322.1**
[39]	Gulbas G. and Cetin G.	100	R	100*100	0.5	50	10	4000	81.5	**313.2**
[26]	Debasis et al.	100	R	100*100	5	Not defined	Not defined	800+160	31.25	**313.2**

Abbreviations: NoN ⇐ Number of Nodes, ND ⇐ Node Deployment, R ⇐ Random, IRoL ⇐ Improvement Ratio over LEACH

IRoL for PM ⇐ Improvement Ratio over LEACH for Proposed Method

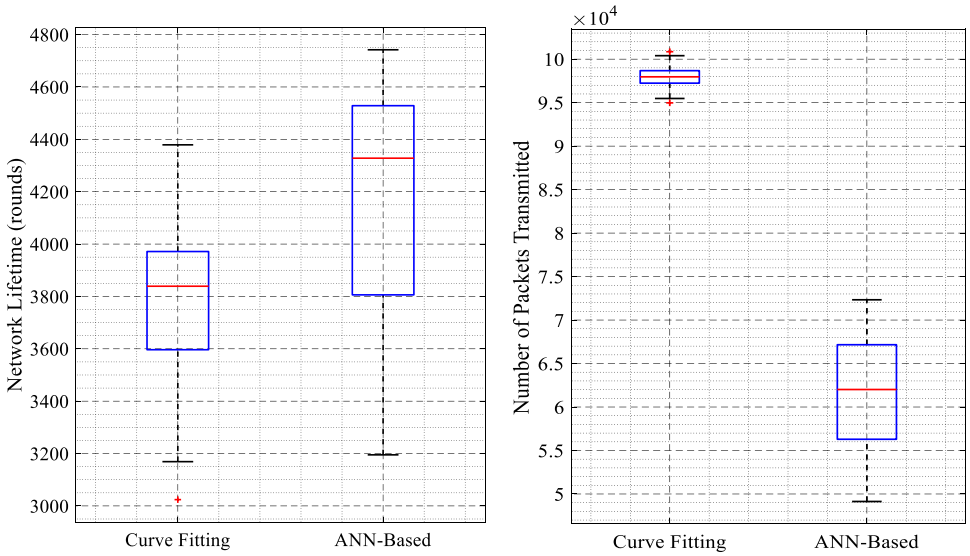


Fig. 18 Box plot comparison of Network Lifetime (left) and Total Packets Transmitted (right) for the Curve Fitting and ANN-Based methods over 100 simulation runs

Table 6 Comparison of Network Lifetime (Min: Minimum, Q1: First Quartile, Median, Q3: Third Quartile, Max: Maximum) in rounds

Method	Min	Q1	Median	Q3	Max
Curve Fitting	3024	3596.5	3879	3971.5	4379
ANN Based	3195	3806	4328	4528.5	4742

Table 7 Comparison of Packet Transmission (Min, Q1, Median, Q3, Max) to the Sink

Method	Min	Q1	Median	Q3	Max
Curve Fitting	94974	97240	97968	98677.5	100856
ANN Based	49136	56291.5	62013.5	67162.5	72330

over the similar studies across all tested parameter sets. When comparing the 'IRoL for PM' to the original 'IRoL', the calculated relative performance improvement factor ranges from 3.63 to 6.26.

Beyond this comparison with external studies, a further statistical analysis was conducted to investigate the internal trade-offs within our own proposed methodology. The performance of the two primary approaches the direct ANN-based method and the computationally lighter equation-based method were compared by replicating the simulations 100 times.

The network lifetime results, presented in Fig. 18 (left side) and Table 6, show that the ANN-Based Method offers a statistically longer network lifetime. This is attributed to the ANN model's ability to make more precise decisions than the equations, which serve as an approximation.

Conversely, the packet transmission results in Fig. 18 (right side) and Table 7 suggest a notable trade-off between the two methods. It was observed that the Equation-Based Method transmitted a higher number of packets during its operational lifespan. This finding implies that the two proposed approaches may be suited for different operational objectives, offering flexibility for various application requirements such as maximizing absolute lifetime or maximizing data throughput.

5 Conclusion

We present a distributed approach for CH selection in wireless sensor networks using a Cascade-ANN-based method that determines threshold values. The proposed methodology demonstrates a substantial improvement in performance compared to the conventional LEACH protocol, as well as some of the recent studies in this field. It is worth noting that the proposed approach requires a database to train the Cascade-ANN. This database can be obtained either through simulations or from a physical network. Additionally, the nodes involved in this method must possess adequate processing resources to perform the necessary calculations for the ANN.

To determine the optimal complexity threshold, nine different threshold values ranging from 0.1 to 0.9 are applied to the neural network, and the threshold that results in the highest remaining energy in the network's output is selected. Consequently, the neural network runs nine times, with its operations executed nine times every 100 rounds. Although this approach can be computationally demanding, particularly as the number of nodes per layer increases, our findings show that even with just five nodes per layer, the neural network delivers sufficiently good results. This demonstrates that the proposed method can be implemented with a reasonable computational load. Furthermore, the threshold equations derived from ANN-generated threshold values help reduce computational complexity without compromising energy efficiency or communication capacity.

Although the simulation results for the proposed approach are encouraging, it is important to discuss certain

aspects and limitations regarding its practical implementation. While this study does not include physical hardware tests, our methodology was designed with practical applicability in mind from the outset.

Considering the practical applicability, the computation and memory analysis presented in Section 3 shows that the load of our proposed final solution (the lightweight mathematical equation) on WSN hardware is extremely low (∼ 680 bytes of memory and ∼500 FLOP operations per 100 rounds). These requirements are within the capacity of commonly used low-power sensor nodes such as the TelosB and MSP430. This reveals that, unlike some computationally intensive optimization algorithms, our methodology exhibits reduced computational overhead. While the simulation results suggest potential applicability to physical performance, they are obtained under an idealized environment.

However, it is important to critically evaluate the practical implementation of the proposed mechanism. While the 5-5-5 Cascade-ANN model is significantly more efficient than larger architectures, its cumulative computational load from multiple runs (approximately ∼4,500 FLOPs per round) and its memory footprint (∼680 bytes) can still be challenging for some highly resource-constrained devices. To overcome this potential drawback, an important contribution of the current study is the creation of an equation-based alternative derived from the ANN. As detailed in Section 4.1, this approach replaces the several iterative runs of the ANN with a single, non-iterative calculation. This reduces the computational load to a negligible level (down to ∼10 FLOPs for the polynomial model) and decreases memory usage. This flexibility ensures that our suggested mechanism is an effective and scalable solution that can be applied for a wide range of WSN hardware. Therefore, our recommendation is that nodes should use these simple equations, an actually lightweight and effective solution that properly addresses the computing overhead of the direct ANN approach.

While the proposed method demonstrates improved performance under the considered simulation settings, several limitations should be noted. The current implementation assumes a static network topology; the effectiveness of the ANN-based threshold in highly mobile or rapidly changing environments has not yet been evaluated. On the other hand, the proposed method requires a pre-collected dataset for offline training, implying that the initial configuration depends on deployment-specific historical data.

On the other hand the current study assumes a static network topology, the underlying mechanism for this generalization is based on practical techniques suitable for dynamic environments.

In this study, as in many fundamental WSN research, a network topology with static nodes was assumed. However, the underlying mechanisms of the proposed method are designed to minimize assumptions. The proposed architecture minimizes the need for global information, largely basing the decision-making process on each node's own local energy level and its estimated distance to the sink.

Instead, a significant practical advantage of the proposed method is that it does not require precise positioning (e.g., GPS) for this category information. A node can estimate its approximate distance to the sink by measuring the Received Signal Strength Indicator (RSSI) of periodic beacon packets emitted from the sink. This estimation would be sufficient to place the node into one of the four broad distance categories defined in this study (e.g.,<15m, 15-25m, etc.). Therefore, the potential for generalization 'without precise location information' is based on this commonly used, low-cost, and energy-efficient RSSI-based mechanism. This feature makes our methodology extremely practical and applicable for statically deployed, scalable networks.

These features provide a flexible foundation for the method's future studies. The low computational load of our method offers significant potential for its adaptation to dynamic network scenarios, particularly those with mobile nodes. A mobile node can periodically update its category based on changes in its position and operate the decision-making mechanism instantaneously. Our future research will focus on evaluating the performance of this proposed method in mobile WSNs.

In future research, we intend to implement and evaluate the proposed method in a physical network. Although the first-order radio model used in this study is a widely accepted abstraction, the next step is to fully measure the robustness of our methodology under the dynamic conditions of real-world wireless channels, such as fading and interference. This is particularly relevant for networks where node locations are fixed and known, such as in remote meter reading applications.

Furthermore, by examining known distance spans to the sink, future research may investigate the generalization of the proposed threshold equations. This strategy might make it possible to use the method without requiring knowledge about the exact locations of nodes. Consequently, a key part of this future work will be to validate the performance of the practical RSSI-based categorization mechanism, especially in dynamic and mobile WSN environments, thus exploring the full potential of our computationally lightweight approach.

Author Contributions Gökçe Hacıoğlu: Conceptualization, Methodology, Formal Analysis, Writing - Original Draft, Supervision. Gökçe Hacıoğlu led the conceptual development of the study, designed the methodology, and conducted the formal analysis. He also took the lead in drafting the manuscript and supervising the research project. Volkan Gangal: Data Curation, Software, Writing - Review and Editing. Volkan Gangal was responsible for data curation and the implementation

of the software used in the simulations. He also contributed to the manuscript through critical review and editing. Erhan Sesli: Validation, Visualization, Writing - Review and Editing. Erhan Sesli provided validation for the research results and contributed to data visualization. He also played a role in reviewing and editing the manuscript to enhance its clarity and scientific rigor.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). No funding was received for conducting this study.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Ethics declaration Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Ayaz, M., Ammad-Uddin, M., Baig, I., Aggoune, E.H.M.: Wireless sensor's civil applications, prototypes, and future integration possibilities: a review. IEEE Sens. J. **18**, 4–30 (2018)
2. Kandris, D., Nakas, C., Vomvas, D., Koulouras, G.: An up-to-date survey, Applications of wireless sensor networks (2020)
3. Ali, S., Kumar, R.: Hybrid energy efficient network using firefly algorithm, pr-pegasis and adc-ann in wsn. Sensors International **3**, 100154 (2022)
4. Zivkovic, M. et al.: Wireless sensor networks life time optimization based on the improved firefly algorithm. 2020 Intl. Wireless Commn. Mob. Comput. (IWCMC), pp. 1176–1181. IEEE, Limassol, Cyprus (2020)
5. Umbreen, S., Shehzad, D., Shafi, N., Khan, B., Habib, U.: An energy-efficient mobility-based cluster head selection for lifetime enhancement of wireless sensor networks. IEEE Access **8**, 207779–207793 (2020)
6. Akyildiz, I.F., Vuran, M.C.: Wireless Sensor Networks. John Wiley & Sons (2010)
7. El Khediri, S. et al.: Improved node localization using K-means clustering for wireless sensor networks (2020)
8. Sethi, D., Bhattacharya, P.: A study on energy efficient and reliable data transfer (eerdt) protocol for wban . 2016 Second Intl. Conf. Computational Intell. Commn. Tech. (CICT), IEEE, Ghaziabad, India, 254–258pp (2016)
9. Daanoune, I., Abdennaceur, B., Ballouk, A.: A comprehensive survey on leach-based clustering routing protocols in wireless sensor networks. Ad Hoc Netw. **114**, 102409 (2021)
10. Chang, L., Li, F., Niu, X., Zhu, J.: On an improved clustering algorithm based on node density for WSN routing protocol. Clust. Comput. **25**, 3005–3017 (2022)
11. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. Proc. 33rd Annual Hawaii Intl. Conf. Syst. Sci., IEEE, Maui, HI, USA, 10 pp. vol.2 (2000)
12. Gamal, M., Mekky, N.E., Soliman, H., Hikal, N.A.: Enhancing the lifetime of wireless sensor networks using fuzzy logic leach technique-based particle swarm optimization. IEEE Access **10**, 36935–36948 (2022)
13. Raziah, I., Yunida, Y., Away, Y., Muharar, R., Nasaruddin, N.: A new adaptive power control based on leach clustering protocol for interference management in cooperative d2d systems. IEEE Access **10**, 113513–113522 (2022)
14. Sajedi, S.N., Maadani, M., Nesari Moghadam, M.: F-leach: a fuzzy-based data aggregation scheme for healthcare iot systems. J. Supercomput. **78**, 1030–1047 (2022)
15. Aljumaie, G.S., Alhakami, W.: A secure leach-pro protocol based on blockchain. Sensors **22**, 8431 (2022)
16. Nagarajan, M., Janakiraman, N., Balasubramanian, C.: A new routing protocol for wsn using limit-based jaya sail fish optimization-based multi-objective leach protocol: an energy-efficient clustering strategy. Wireless Networks **28**, 2131–2153 (2022)
17. Nehra, N., Kumar, M., Patel, R.: Neural network based energy efficient clustering and routing in wireless sensor networks. 2009 First Intl. Conf. Netw. Commn. IEEE, Chennai, India, 34–39pp (2009)
18. Subha, C., Malarkan, S., Vaithinathan, K.: A survey on energy efficient neural network based clustering models in wireless sensor networks. 2013 Intl. Conf. Emerging Trends in VLSI, Embedded Syst. Nano Elect. Telecommn. Syst. (ICEVENT), IEEE, Moradabad, India, 1–6pp (2013)
19. Osamy, W., Khedr, A.M., Salim, A., AlAli, A.I., El-Sawy, A.A.: Recent studies utilizing artificial intelligence techniques for solving data collection, aggregation and dissemination challenges in wireless sensor networks: a review. Electronics **11**, 313 (2022)
20. Mishra, K., Sharma, P.: Improved cluster head selection using particle swarm optimization and neural network in wsn . 2021 Intl. Conf. Comput. Sci. (ICCS), IEEE, Phagwara, India, 13–18pp (2021)
21. Sharma, A., Kansal, A.: Advanced ANN based secured energy efficient routing protocol in WSN. Wireless Pers. Commun. **132**, 2645–2666 (2023)
22. Kong, P.Y.: Distributed sensor clustering using artificial neural network with local information. IEEE Internet Things J. **9**, 21851–21861 (2022)
23. Revanesh, M., et al.: Artificial neural networks-based improved Levenberg–Marquardt neural network for energy efficiency and anomaly detection in WSN. Wireless Netw. (6),(2023). https://doi.org/10.1007/s11276-023-03297-6
24. Sharma, A., Kansal, A.: Enhanced ch selection and energy efficient routing algorithm for wsn. Microsyst. Tech. 1–13 (2024)
25. Qamar, M.S., et al.: A novel approach to energy optimization: Efficient path selection in wireless sensor networks with hybrid ann. Computers, Materials & Continua (2024). https://doi.org/10.32604/cmc.2024.050168
26. 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. Mobile Netw. Appl. **28**, 853–867 (2023)
27. Abdulzahra, A.M.K., Al-Qurabat, A.K.M.: A clustering approach based on fuzzy c-means in wireless sensor networks for iot

applications. Karbala International Journal of Modern Science **8**, 579–595 (2022)
28. Wang, Z., DIng, H., Li, B., Bao, L., Yang, Z.: An energy efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks. IEEE Access **8**, 133577–133596 (2020)
29. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. Rep. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
30. Kennedy, J., Eberhart, R.: Particle swarm optimization. Proc. ICNN'95 - Intl. Conf. Neural Netw. IEEE, vol. 4, pp. 1942-1948 (1995)
31. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
32. Somula, R., Cho, Y., Mohanta, B.K.: Swaram: osprey optimization algorithm-based energy-efficient cluster head selection for wireless sensor network-based internet of things. Sensors (2024). https://doi.org/10.3390/s24020521
33. Lonkar, B., Kuthe, A., Charde, P., Dehankar, A., Kolte, R.: Optimal hybrid energy-saving cluster head selection for wireless sensor networks: an empirical study. Peer-to-Peer Netw. Appl. **18**, 182 (2025)
34. Ahmad, I., et al.: Accelerated particle swarm optimization algorithm for efficient cluster head selection in wsn. Computers, Materials & Continua **79**, 3585–3629 (2024)
35. Pour, S.E., Javidan, R.: A new energy aware cluster head selection for leach in wireless sensor networks. IET Wireless Sens. Syst. **11**, 45–53 (2021)
36. Nezha, E.I., Abdellah, N., Hassan, E.A.: Energy-aware clustering and efficient cluster head selection. International Journal on Smart Sensing and Intelligent Systems **14**, 1–15 (2021)
37. Wu, M., Li, Z., Chen, J., Min, Q., Lu, T.: A dual cluster-head energy-efficient routing algorithm based on canopy optimization and K-Means for WSN. Sensors **22**(24), 9731 (2022)
38. Abdulzahra, A.M.K., Al-Qurabat, A.K.M., Abdulzahra, S.A.: Optimizing energy consumption in wsn-based iot using unequal clustering and sleep scheduling methods. Internet of Things **22**, 100765 (2023)
39. Gülbaş, G., Çetin, G.: Lifetime optimization of the LEACH protocol in WSNs with simulated annealing algorithm. Wireless Pers. Commun. **132**, 2857–2883 (2023)
40. Jo, H.-S., Park, C., Lee, E., Choi, H.K., Park, J.: Path loss prediction based on machine learning techniques: principal component analysis, artificial neural network, and gaussian process. Sensors **20**, 1927 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.