Wireless Personal Communications
https://doi.org/10.1007/s11277-023-10812-7

RESEARCH

Enhancing Network Efficiency and Extending Lifetime Through Delay Optimization and Energy Balancing Techniques

Amrita Jyoti[1] · Rashmi Sharma[2] · Pooja Singh[3] · Harsh Khatter[4] · Rashmi Mishra[5]

Accepted: 6 December 2023

Abstract

Contrary to homogeneous WSNs, heterogeneous WSN protocols make use of sensor fork with a variety of capabilities to extend the network's life, improve cluster stability, and assure accurate information transfer. Even though numerous authors have put forth various protocols, none of them have been able to successfully balance power consumption among basic fork, advanced fork, and cluster heads according to application needs and localization. Improving system longevity and performance requires reducing power consumption by sensor fork. While the location of the base station is known, each protocol arranges the sensor fork at random. Cluster head selection, set-up, and steady state phases are the standard three processes in the protocols. Depending on the network design, each protocol's decision-making process considers the node's remaining power as well as the system's total power. This study examines partitioning or cluster strategies based on power remaining and contrasts them in terms of a numerals of facets, including power effectiveness and stability duration.

Keywords Residual power · Heterogeneous · Hierarchical routing · WSN · Power conservation

✉ Rashmi Mishra
dtuphd.rashmi@gmail.com

Amrita Jyoti
amritajyoti2013@gmail.com

Rashmi Sharma
drrashmisharma20@gmail.com

Pooja Singh
pooja.singh@snu.edu.in

Harsh Khatter
harsh.khatter@kiet.edu

1 ABES Engineering College, Ghaziabad, UP, India

2 Ajay Kumar Garg Engineering College, Ghaziabad, UP, India

3 Shiv Nadar University, Greater Noida, UP 201314, India

4 KIET Group of Institutions, Delhi-NCR, Ghaziabad, UP 201206, India

5 Delhi Technology University, Rohini, New Delhi, Delhi 110042, India

Published online: 20 December 2023

List of Symbols

Eelec	Power for transmitting 1 bit
Efs	Free space power
Eamp	Amplification power
EDA	Power for data aggregation
D0	Threshold interval
E0	Original power of the normal fork

1 Introduction

Wireless Sensor Networks (WSNs) have emerged as a cutting-edge technology with enormous potential for revolutionizing many facets of our life in the age of connectivity and ubiquitous computing. These networks are made up of tiny, self-contained sensor fork having wireless communication, processing, and sensing capabilities. They work together to collect data from the environment, process it locally, and send it to a base station or central node. Real-time monitoring, analysis, and control of physical and environmental facets are made possible by this comprehensive data collection in a variety of applications [1]. The developments in microelectronics, wireless communication technologies, and the rising demand for effective monitoring and control systems have all contributed to the development and dissemination of WSNs. Numerous industries, including environmental monitoring, industrial automation, healthcare, agriculture, smart cities, and more use these networks. WSNs offer previously unattainable opportunities for data collection from remote and hostile areas by placing a large numeral of small sensors in the target area [2]. WSNs' wireless nature, which eliminates the need for a major cabling infrastructure and enables simple deployment and reconfiguration, is one of their most important advantages. The processing, storage, and communication capabilities of the sensor fork are often restricted and they run on batteries. The development of effective protocols and algorithms for data communication, power management, and network optimization presents special problems because of these resource constraints [3].

Numerous industries, including environmental monitoring, industrial automation, healthcare, agriculture, smart cities, and more use these networks. WSNs offer previously unattainable opportunities for data collection from remote and hostile areas by placing a large numeral of small sensors in the target area. WSNs' wireless nature, which eliminates the need for a major cabling infrastructure and enables simple deployment and reconfiguration, is one of their most important advantages [4]. The processing, storage, and communication capabilities of the sensor fork are often restricted and they run on batteries. The development of effective protocols and algorithms for data communication, power management, and network optimization presents special problems because of these resource constraints. We will go into the difficulties WSNs encounter and investigate the cutting-edge fixes suggested in the literature. Understanding the complexities of WSNs will help us realize their full potential and open the door for their widespread adoption across a variety of sectors, bringing about a world that is smarter and more connected [5].

Homogeneous and heterogeneous networks are combined in WSNs. Power usage and resource restrictions are both heavily influenced by routing algorithms. Due to their limited power resources, sensor fork needs computational power and sensing capabilities. Strong routing algorithms are required to increase scalability, reliability, and time efficiency and extend the system lifetime of WSNs [6].

Wireless sensor networks (WSNs) are composed of small sensors distributed throughout the network. These sensors could sense data, process it, and transmit it from one node to another. They are utilized in various applications such as military operations, weather forecasting, industrial areas, medical services, and agriculture, among others, for data transmission. These sensors are compact in size and powered by limited batteries. Once deployed in harsh and inaccessible environments, they must be utilized efficiently since they cannot be easily replaced or recharged [7]. Fork in the network consume power for sensing and transmitting data to the base station. Efforts have been made to maximize the overall lifespan of the network by conserving the power of the sensors. Liank et al. proposed a mechanism known as the Huang mechanism, which aims to balance power utilization across the network by selecting optimal clusters. However, this mechanism is complex and can potentially block the channel when dealing with large data packet sizes [8]. On the other hand, Cardei et al. introduced the TianD mechanism, where sensor fork is organized in disjoint sets to cover the maximum range. Compared to the Huang mechanism, the TianD mechanism is less complex but lacks the ability to detect duplicate fork and data. The main challenges in wireless sensor networks can be categorized into three sections: power efficiency, quality of service, and security. Optimization mechanisms are employed in WSNs to address these challenges, including optimizing sensor coverage, data aggregation, power-efficient partitioning or cluster and routing, and sensor localization [9]. These issues are interconnected, meaning that addressing one problem may require trade-offs with the network's lifespan. Similarly, achieving security may also impact the network's longevity. Focusing on individual problems separately leads to unresolved loopholes. Therefore, to create a balanced wireless sensor network, an optimized mechanism that simultaneously addresses all these problems is needed. This can be achieved by developing a multipurpose function and employing an appropriate optimizer mechanism [10]. When selecting the appropriate mechanisms, various facets such as problem type, time constraints, resource availability, and desired accuracy are considered. Researchers have utilized different approaches, including classical approaches and swarm intelligence-based approaches, inspired by nature, to enhance the network's performance. In the literature, various mechanisms have been proposed to tackle specific issues in wireless sensor networks, such as Optimal Scope, Data Aggregation, Power Efficient Partitioning or cluster, Power Efficient Routing, and Sensor Localization. Routing protocols define how fork communicate with each other and how information is transported between locations [8, 9]. These methods can be classified into three categories: 1. Centralized Algorithms: These algorithms are uncommon due to the impracticality and cost associated with transmitting the system's status to a single node. 2. Scattered Algorithms: Communication is achieved through memo passing. 3. Local-based Algorithms: These algorithms operate within specific areas, constrained or contiguous. They store local information on a single node and run the algorithm exclusively on that node, utilizing the locally cached data. Wireless Sensor System routing protocols need to meet specific standards due to limitations within the system. These standards include: 1. Autonomy: Wireless sensor networks (WSNs) operate in a decentralized manner without a centralized organization making routing decisions. This lack of well-defined routing procedures makes WSNs vulnerable to potential attacks. 2. Power Efficiency: Routing protocols should be designed to maximize the system's lifespan and maintain efficient communication between fork. As sensor fork are often placed in inaccessible locations, it becomes challenging to replace their batteries. 3. Scalability: With WSNs consisting of hundreds of fork, routing protocols must effectively handle the large numerals of fork within the system. 4. Resilience: Routing protocols need to establish alternative paths for data transmission in case certain fork become non-functional due to facets like external influences or battery depletion. 5. Device Heterogeneity: The diversity of sensor forks in terms of processing power, transceivers, power units, and bandwidth allows for different routing strategies suitable for WSNs. 6. Mobility Adaptability: Wireless sensor systems face the challenge of node mobility, as some applications require fork to accommodate movement. Routing protocols should incorporate provisions to handle such mobility requirements. 7. Complexity: Due to hardware limitations and power constraints, routing strategies should strike a balance between functionality and system performance to avoid excessive complexity that could hinder the wireless system's performance. The use of partitioning or cluster algorithms has been widely acknowledged in the literature as an effective means of reducing power consumption in fork, while also enhancing system performance and scalability. With advancements in micro electro mechanical sensors and input from researchers, there has been an increase in the deployment of dense and cost-effective sensor fork. Wireless Sensor Networks (WSNs) find applications in various fields such as military, traffic monitoring, agriculture, healthcare, surveillance, disaster relief, and more. Partitioning or cluster plays a crucial role in balancing the power consumption within the environment. In this approach, the cluster head collects and relays aggregated information to the base station [7, 8, 10]. Only long-range sensors are required to transmit data directly to the base station, thereby prolonging the system's operational duration. Partitioning or cluster can be implemented in two types of environments: homogeneous, where sensor fork is in proximity and possess similar power capabilities, and heterogeneous, where fork exhibit varying power levels [9, 10]. Researchers have proposed several protocols for the homogeneous environment, such as Threshold Sensitive Power Efficient Sensor Environment (TEEN), Adaptive Threshold-sensitive Power Efficient Environment (APTEEN), Hybrid Power-Efficient Distributed partitioning or cluster (HEED), Low-Power Adaptive Partitioning or cluster Hierarchy (LEACH), among others. WSNs typically comprise a combination of both homogeneous and heterogeneous networks [3]. Routing algorithms play a critical role in resource management, particularly power consumption. Sensor fork face limitations in terms of power availability, computational capacity, and network awareness. Hence, robust routing algorithms are necessary to prolong system life, enhance scalability, and ensure reliable network operations [4]. In terms of heterogeneous resources in WSNs, three main categories can be identified: power, link, and computation. Power heterogeneity is characterized by different power levels among fork, including two-level, three-level, or multi-level configurations. Routing methods optimize network performance by assigning power-intensive tasks to high-power sensors. Link heterogeneity allows for diverse forms of interaction between sensor fork, such as bidirectional or unidirectional communication. Several techniques leverage link heterogeneity to improve network lifespan and reduce delays. Cognitive heterogeneity considers the varying hardware capabilities of sensor fork to handle more complex tasks, while accounting for traffic diversity [5]. The following paragraphs discuss the impact of heterogeneity on wireless sensor networks (WSNs), addressing the differences between diverse and homogeneous networks. Diverse wireless sensor networks tend to have a longer network lifetime, as the variability in link quality, computing capabilities, and power consumption directly influence their performance Network health is evaluated based on the numerals of active fork, the selection of cluster heads, packets transmitted through the central station, and the stability of network timing [6, 7]. The equilibrium period represents the cycle during which the first fork fail, while system time indicates the integrity of the network and correlates with the lifespan of sensing fork. The numerals of active fork in each round, determined by the remaining power, reflects the numerals of fork still functioning. Capacity refers to the numerals of packets delivered either from a regular network to a cluster leader or from a cluster head to the sink node [8]. The selection of cluster heads is influenced by the transmissions received by the base station, favoring sensor fork with higher remaining power and proximity to the sink station [9]. Node heterogeneity plays a role in determining the throughput and latency of data transmission from source to destination. Cluster-based heterogeneous algorithms can be classified using various parameters. In a heterogeneous environment, there are fewer hops between the source and destination, resulting in a higher end-to-end delivery rate compared to a homogeneous environment [6, 7].

1.1 Types of Heterogeneous Resources

The idea of heterogeneous resources is significant when it comes to resource allocation in many systems, including computer networks and distributed computing environments. Resources that differ in their capabilities, competencies, or traits are referred to as heterogeneous resources [11]. These variations can affect a wide range of facets, including processing speed, memory size, network bandwidth, storage space, and power supply. In many computing systems, effective resource management and optimization depend on an understanding of the different types of heterogeneous resources.

a. Processing Resources: A system's processing resources include its available computing power. They consist of various processors or computer units with diverse speeds, architectures, and capacities. A system might include general-purpose processors, high-performance multi-core processors, specialized accelerators (such GPUs or FPGAs), or even low-power embedded processors. Each processing resource has strengths and weaknesses, therefore using these resources effectively calls for load balancing and sophisticated scheduling strategies [12].
b. Memory Resources: A system's memory resources are its available memory devices' performance and storage capabilities. A variety of memory structures, including as registers, caches, main memory (RAM), and secondary storage (hard drives or solid-state drives), are included in these resources. Memory resource heterogeneity results from differences in capacity, access latency, bandwidth, and durability. The efficiency of memory utilization and data placement techniques can have a big impact on system performance [13].
c. Network Resources: A system's or a network's communication architecture is referred to as a network resource. They include distinct network topologies, multiple communication links, such as wired or wireless connections, and various bandwidth capacity [14, 15]. Different transmission speeds, latency, reliability, and QoS (Quality of Service) facets lead to heterogeneous network resources. Routing algorithms, congestion control systems, and bandwidth distribution plans are all necessary for effective network resource usage.
d. Storage Resources: A system's storage resources are its storage devices' performance and capacity. Local hard drives, network-attached storage (NAS), and cloud-based storage systems are some examples of these gadgets. Disparities in capacity, access latency, throughput, durability, and fault tolerance cause heterogeneity in storage resources. Data management strategies, replication tactics, and data placement algorithms all play a role in the efficient use of storage resources [16, 17].
e. Power Resources: A system's devices or fork' ability to get electricity and how much of it they use depends on its power resources. Heterogeneity in power resources occurs in battery-powered systems as a result of differences in battery capacity, component

power consumption rates, and power harvesting abilities. In order to increase the system's operating lifetime, efficient power management approaches are required, such as dynamic power scaling, duty cycling, and power-aware job scheduling [18].

For the creation of resource management rules, scheduling algorithms, and optimization strategies in varied computing environments, it is essential to comprehend the types and characteristics of heterogeneous resources. System designers and resource managers can boost performance, usage, and overall efficiency in a variety of computer systems by utilizing the advantages of various resources and adjusting to their constraints [19].

1.2 Impact of Heterogeneity on WSN

Wireless Sensor Networks (WSNs)' functionality and performance are significantly shaped by heterogeneity. Multiple effects on WSNs may result from the existence of various sensor fork with various capabilities, features, and resource limitations include:

a. Better resource usage is made possible by WSN heterogeneity. Based on their strengths, individual fork with distinct skills might be given particular jobs. For instance, fork with more processing capacity can tackle complex computations, whilst fork with less power can concentrate on easier jobs. Task distribution in this way optimizes resource use and boosts overall network performance [20].
b. Network Lifetime: The network lifetime of WSNs is directly impacted by the heterogeneity in power resources. Power resources will be used up by fork at varied rates depending on their power capacity and consumption rates. Power-aware approaches can be used to increase the network lifetime by utilizing the heterogeneity. While fork with lesser power can be put into a sleep mode to save power, fork with higher power can take on more demanding duties [21].
c. Data Processing and Fusion: Data processing and fusion in WSNs are impacted by heterogeneity in processing speeds and memory sizes. Data fusion and analysis tasks can be completed by fork with differing levels of computing capability [22].
d. Heterogeneity in transmission power and communication range has an impact on the connectivity and coverage of WSNs. Wider network coverage can be provided by fork with various communication ranges, enabling improved connectivity and data collecting from a variety of locations. Due to its heterogeneity, the network is able to successfully handle communication difficulties including obstructions and signal strength changes and adapt to changing environmental conditions [23].
e. Heterogeneity improves WSNs' fault tolerance and dependability in terms of both. The heterogeneous architecture of the network enables unaffected fork to make up for the loss by taking over the duties of failing fork in the event of node failures or malfunctions. By placing fork with comparable functionality, redundancy can be added, guaranteeing that the network can still function in the event of a breakdown [24, 25].
f. Application Specificity: Due to their heterogeneity, WSNs may accommodate a variety of applications and circumstances. Application-specific specifications for sensing capabilities, computing power, and communication range may exist. Because heterogeneous resources are present, WSNs are more adaptable and suitable to a variety of domains because they can be customized and adjusted to meet the needs of certain applications [26].

The effects of heterogeneity in WSNs must be carefully considered during the design and optimization phases. In WSNs, effective resource management, increased power efficiency, and higher overall performance can be attained by utilizing the strengths and capabilities of various fork.

1.3 Types of Node Heterogeneity in HWSN

Heterogeneous Wireless Sensor Networks (HWSNs) exhibit various forms of heterogeneity, primarily stemming from the diversity of sensor fork. This fork possesses distinct capabilities, characteristics, or functionalities, resulting in different types of heterogeneity within HWSNs. The following are common forms of node heterogeneity found in HWSNs:

a. Sensing Heterogeneity: HWSN fork display variations in sensing capabilities, including the use of different sensor types such as temperature, humidity, light, or pressure sensors. Moreover, fork may differ in sensing range, resolution, or sampling rates. This heterogeneity allows the network to capture diverse environmental parameters and enables specialized monitoring of specific aspects [16, 17].
b. Processing Heterogeneity: HWSN fork vary in processing capabilities due to differences in computational power, processing speed, or memory capacity. Some fork possesses greater computational resources for complex calculations or data processing tasks, while others have limited capabilities suited for simpler tasks [18].
c. Communication Heterogeneity: Heterogeneity in communication capabilities is prominent among HWSN fork. It encompasses variations in communication range, transmission power, or supported communication protocols. This heterogeneity enables fork to have different communication capacities, ensuring efficient data exchange and network connectivity [19].
d. Power Heterogeneity: Power heterogeneity refers to variations in power resources among sensor fork in HWSNs. Fork may have different battery capacities or power consumption rates, leading to disparities in their power levels. Managing power-heterogeneous fork requires techniques that balance power consumption, extend network lifetime, and prevent power-depleted fork [20].
e. Mobility Heterogeneity: Some HWSNs feature fork with differing mobility patterns and capabilities. Mobile fork can move within the network, collecting data from various locations or performing specific tasks. Mobility heterogeneity allows for dynamic network topology, adaptable data collection, and distributed monitoring in environments with freely moving fork [21].
f. Data Processing Heterogeneity: HWSN fork possess varying data processing capabilities, resulting from differences in their ability to process and analyze collected data. Advanced processing algorithms or data fusion techniques may be present in some fork, enabling sophisticated data processing tasks, while others have limited processing capabilities [22].

The presence of these forms of node heterogeneity in HWSNs brings both challenges and opportunities. Effective management of heterogeneous fork involves designing customized algorithms, protocols, and resource allocation strategies to leverage the diverse capabilities of the fork, optimize resource utilization, enhance network performance, and enable efficient data collection and analysis in HWSNs.

Key Contribution: By introducing dynamic cluster formation based on energy levels and distance from the base station, the proposed method ensures even energy distribution across the network and increases network longevity. In order to reduce energy-intensive transmissions and facilitate effective data aggregation, cluster head selection takes energy levels and proximity into account. Concise announcements are sent through reliable cluster maintenance procedures, which also reduce overhead and latency while improving end-to-end delay and total network throughput. The algorithm's flexibility allows for smooth transition between real-time reporting and energy-efficient aggregation for prompt data transmission. Our method regularly beats LEACH, SEP, and HEED through thorough simulations and evaluations, excelling in active nodes, energy consumption, end-to-end latency, and throughput, demonstrating its potential to dramatically improve WSN performance.

2 Literature Survey

The authors recognize that there are earlier survey articles on Wireless Sensor Networks (WSNs) that give thorough overviews of the industry. However, they draw attention to the paper's particular emphasis on addressing problems with WSN power usage. Their goal is to categorize routing protocols according to how they communicate with base stations (BS) and the variables utilized to make routing decisions, with a focus on power conservation and balance. However, other studies, such those in [27, 28], provide a broader view on WSNs by going through a variety of applications and the influences on their design. These studies examine communication architecture, examine routing methods across various communication layers, and offer potential WSN research axes as a conclusion. A taxonomy addressing power-efficient and power-balanced routing protocols is presented in this research, which focuses on power consumption optimization challenges. A different survey [11] categorizes routing methods used in WSNs into three categories: flat, hierarchical, and location-based routing. Additionally, it takes into account metrics for negotiation-based, QoS-based, and multipath routing. The survey compares different routing algorithms in-depth, highlighting their advantages and disadvantages in terms of reducing power consumption and communication overhead. This work, in contrast, focuses on a survey of contemporary power-efficient and power-balanced routing methods. The taxonomy presented in this research is based on the communication method and the choice facets used to create these routing algorithms.

Overall, by focusing on power-efficient and power-balanced routing methods in WSNs, the authors want to add to the body of knowledge. Their taxonomy and analysis provide insights into various communication and decision-making modalities employed in these protocols, ultimately aiming to optimize power usage. The surveys described in [29–31] are acknowledged by the authors since they offer insightful information on a variety of WSN-related topics, such as routing protocols, WSNs as a whole, and power conservation. However, they draw attention to the paper's special focus on WSN power usage optimization during network data transmission. According to different quality of service (QoS) needs, the examined routing methods in [29] are divided into data-centric, hierarchical, and location-based groups. In contrast, this research focuses exclusively on power-efficient and power-balanced routing protocols with the goal of reducing sensor node (SN) power consumption and extending network lifetime. The decision facets employed in the routing algorithms and the different types of solutions or algorithms are the basis for the categorization in this study. A basic overview of WSNs, including uses, difficulties, and research advancements, is given in [30]. The internal platform and operating system, communication protocol stack, and network services and deployment are how the authors categorize the difficulties. In contrast, this research focuses primarily on how SNs in WSNs might optimize their power use. It examines the research on power-efficient and power-balanced routing protocols and offers a categorization based on the channel used to communicate with the base station (BS), the kind of solution or algorithm, and the design parameters for each routing method. The survey in [31] focuses on power saving in WSNs, especially looking at how much power SN hardware components utilize. The authors break down the power use for the power supply unit, radio, computation, and sensor subsystems.

They present a taxonomy that divides power-saving plans into mobility-based, duty-cycling, and data-driven categories. when power conservation is important, this study places a focus on reducing power use when network data is being sent. It offers classification and research guidelines for this particular field and focuses on power-efficient and power-balanced routing techniques. The goal of this research is to optimize power usage throughout the network data transmission phase in WSNs. It does this by providing a focused analysis of power-efficient and power-balanced routing algorithms. It provides a taxonomy of these protocols and outlines future research options for increasing network functionality and longevity.

The surveys referenced in literature, which offer insightful information on many aspects of power-efficient and clustered routing protocols for WSNs, are acknowledged by the authors. They draw attention to their paper's unique focus on power-efficient and power-balanced routing protocols, as well as its categorization according to the communication method utilized to reach the base station (BS) and the decision facets employed in the routing algorithms. According to performance concerns and metrics like QoS requirements and data delivery models, the survey in [32] focuses on power-efficient routing methods for wireless multimedia sensor networks (WMSNs).

This study emphasizes the need of dependable power balancing by considering power-efficient and power-balanced routing methods in addition to WMSNs. An extensive overview of swarm intelligence-based routing protocols in WSNs is presented in [33], which also introduces a new taxonomy for categorization. Swarm intelligence-based routing is the topic of [33], but this work analyzes swarm intelligence-based power-efficient and power-balanced routing protocols, classifying them based on the input decision variables employed in the algorithms. Data-centric, hierarchical, and location-based routing are the three categories used by the studies in [4, 34] to categorize power-efficient routing methods for WSNs.

By addressing both power-efficient and power-balanced routing methods, this research, however, broadens the topic. For the BS and decision variables utilized in the algorithms, it presents a taxonomy based on the form of communication, offering insightful comparisons. [10] gives a comprehensive introduction of WSNs, outlining uses, difficulties, and recent research advances. This research concentrates on power consumption concerns during the network data transmission phase, while [10] focuses on the general elements of WSNs. It offers guidance for interested researchers by introducing a taxonomy for power-efficient and power-balanced routing methods.

The studies conducted in [6] encompass a wide range of partitioning or cluster routing protocols for WSNs, categorizing them according to their goals, guiding principles, and cluster formation techniques. This work covers power-efficient and power-balanced partitioning or cluster routing algorithms and acknowledges the value of partitioning or cluster approaches for reducing power consumption. Additionally, it divides the protocols into groups according on how they communicate with the BS and distinguishes between equal and unequal clustered routing techniques. In conclusion, this work offers a careful analysis. of power-efficient and power-balanced routing protocols, taking into account both partitioning or cluster routing techniques and routing protocols for WSNs. It provides a categorization based on the cluster size formulation, decision facets, and communication style. The objective is to offer beneficial insights and potential research paths in the subject of power to scholars and practitioners.

3 Clustering Algorithms

In Wireless Sensor Networks (WSNs), partitioning or clustering is a crucial procedure that includes organizing sensor nodes into clusters in order to enhance different aspects of network performance. These clusters provide for effective data gathering, communication, and resource management and are often directed by a cluster head or leader. In WSNs, partitioning or clustering is primarily used to cut down on power consumption, increase network scalability, prolong network lifetime, and improve system performance [23].

Cluster in WSNs has a number of benefits. The first benefit is that it permits localized data aggregation inside clusters, which lowers the volume of data transfer and saves electricity. Cluster heads gather information from member forks, carry out fusion or aggregation processes, and then send compiled data to the base station or higher-level fork. With this hierarchical structure, redundant data transmission is reduced, network congestion is lessened, and bandwidth usage is maximized [32].

Second, partitioning makes WSNs' routing and communication more effective. In order to transmit information between the member fork and the base station, cluster heads serve as an intermediary. The network can get beyond the individual fork's constrained communication range by using multi-hop communication among clusters. With this strategy, long-distance transmission power consumption is decreased, and network connection and dependability are increased generally [33].

Thirdly, partitioning enables load balancing and resource management in WSNs. Cluster heads distribute tasks, allocate resources, and coordinate activities among member fork within their respective clusters. By intelligently assigning roles and responsibilities, partitioning or cluster helps balance the computational and power workload across the network, preventing fork from becoming overloaded and ensuring efficient resource utilization [34].

Cluster algorithms and protocols in WSNs vary depending on specific objectives and application characteristics. Common approaches include hierarchical partitioning or cluster, such as LEACH (Low-Power Adaptive Partitioning or cluster Hierarchy), where cluster heads are probabilistically selected, and centralized partitioning or cluster, where a central entity assigns fork to clusters based on specific criteria. Distributed algorithms, like HEED (Hybrid Power-Efficient Distributed partitioning or cluster), enable fork to self-organize into clusters based on local information [35, 36].

In summary, partitioning or cluster plays a crucial role in optimizing WSNs by organizing sensor fork into logical groups. It facilitates localized data aggregation, efficient communication, load balancing, and resource management. These advantages lead to increased network scalability, higher power efficiency, and longer network lifetime. A key design and optimization strategy for WSNs is partitioning or clustering, which allows for efficient resource management and improves overall network performance across a range of application areas.

3.1 Cluster Schemes

Homogeneous environments and heterogeneous environments are the two categories into which partitioning or cluster techniques may be divided. In the section that follows, protocols created especially for heterogeneous environments are discussed, and a comparison is performed based on the number of forks that are alive, the number of forks that are dead, and the number of cluster heads that are chosen every round.

3.1.1 Hybrid Power-Efficient Distributed Partitioning or Cluster (HEED)

A partitioning or cluster technique particularly designed for Wireless Sensor Networks (WSNs) is known as Hybrid Power-Efficient Distributed Partitioning or cluster (HEED) [37]. The general flow of the HEED algorithm is shown in Fig. 1. By balancing power usage throughout the sensor fork and optimizing power consumption, HEED's main goal is to increase network longevity. This is achieved by dynamically generating clusters based on both node- and network-level data, allowing for effective resource management and allocation. Because HEED is distributed, each sensor node has the autonomy to choose the cluster configuration and cluster head on its own. In many WSN contexts, this decentralized strategy guarantees scalability and adaptability. The algorithm incorporates two key facets into its decision-making process: residual power and node proximity. Each sensor node in HEED calculates a heuristic value termed the "Node's Degree of Electability" (ND). This value reflects a node's suitability or eligibility to become a cluster head. The ND is determined based on the node's residual power, with fork possessing higher power levels being more likely to be elected as cluster heads. To achieve a well-balanced distribution of cluster heads, HEED introduces a threshold called the "Power Percentage" (Ep). Fork compares their calculated ND values with a randomly generated numerals between 0 and

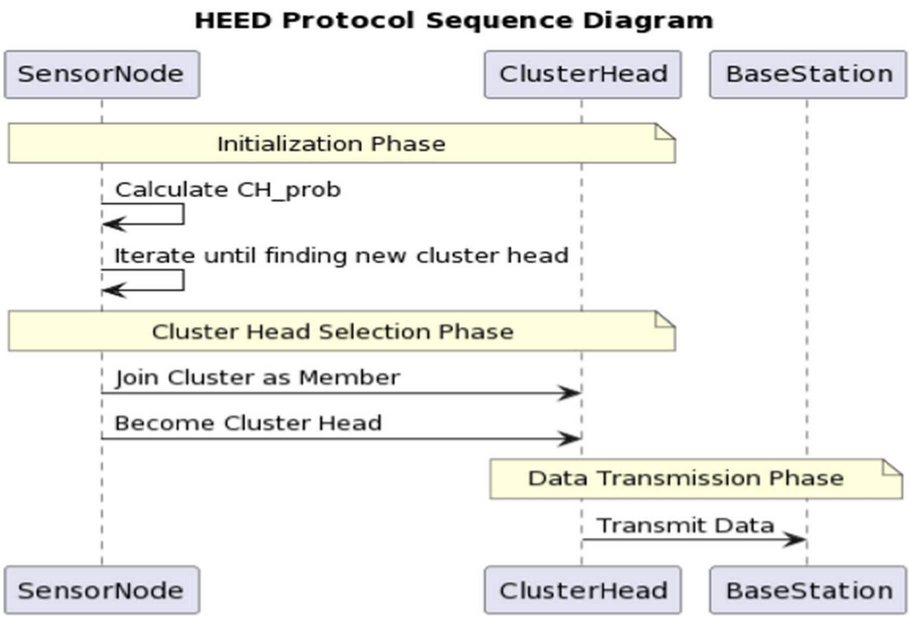


Fig. 1 HEED protocol sequence diagram

1. If the ND exceeds Ep, the node assumes the role of a cluster head; otherwise, it becomes a member of an existing cluster or remains uncluttered. The value of Ep allows for a trade-off between network lifetime and network stability, providing flexibility in achieving the desired performance goals. Furthermore, HEED takes into account node proximity during the partitioning or cluster process. Sensor fork exchange information regarding their ND values and distances to potential cluster heads. Leveraging this information, each node evaluates its connectivity to potential cluster heads and selects the cluster head that offers the strongest connectivity, thus promoting efficient data routing and communication within the network. The cluster formation process in HEED is iterative, enabling the network to adapt to changes in power levels and connectivity over time. The algorithm is periodically executed to update cluster heads and cluster memberships. This periodic reconfiguration ensures a balanced distribution of power consumption, preventing certain fork from being overburdened and enhancing the network's stability and longevity. Cluster heads are also periodically reassigned to different fork, promoting fairness and load balancing within the network. HEED has garnered substantial attention in the research community and has been extensively studied. It has demonstrated promising results in significantly prolonging the network lifetime of WSNs. By dynamically selecting power-efficient cluster heads and facilitating balanced power utilization, HEED effectively addresses power constraints in WSNs, thereby enhancing the overall performance and longevity of the network.

3.2 Distributed Weight-Based Power-Efficient Hierarchical Partitioning or Cluster (DWEHC)

The DWEHC method, proposed by Ding et al. [38], was developed after studying the HEED protocol. Distributed Weight-Based Power-Efficient Hierarchical Partitioning or cluster (DWEHC) is a partitioning or cluster algorithm specifically designed for Wireless Sensor Networks (WSNs). DWEHC aims to optimize power consumption and extend the network lifetime by forming hierarchical clusters in a distributed manner. Figure 2 depicts the overall process of DWEHC algorithm.

In DWEHC, each sensor node independently makes decisions regarding cluster formation and cluster head selection based on a weight-based approach. The algorithm considers two main facets: residual power and node weight. Residual power represents the remaining power level of a sensor node, indicating its power availability. Higher residual power forks are more likely to be chosen as cluster heads since they could be able to function for longer. Node weight, which measures a node's significance in the network, may be determined based on factors including its position, connectedness to other forks, and its distance from the sink node. Due to their importance in network operations, forks with greater weights are more likely to be chosen as cluster heads. The distributed technique used by DWEHC allows forks to communicate with one another about their remaining power and node weight. Each node creates a Combined Weighted Value (CWV) based on this data, which takes both node weight and residual power into account. The CWV is used to assess a node's suitability for the position of cluster head. Forks are more likely to act as cluster leaders if their CWV levels are greater. DWEHC uses a multi-level strategy to establish a hierarchical segmentation or cluster structure. At the top level, the fork with the greatest CWV values is chosen as the cluster head. The surviving fork then merges into the cluster of the closest cluster head. In order to create various layers of clusters within the network, this procedure is done repeatedly. Cluster leaders are in charge of collecting data from their member fork and relaying it to the base station or higher-level fork. The hierarchical

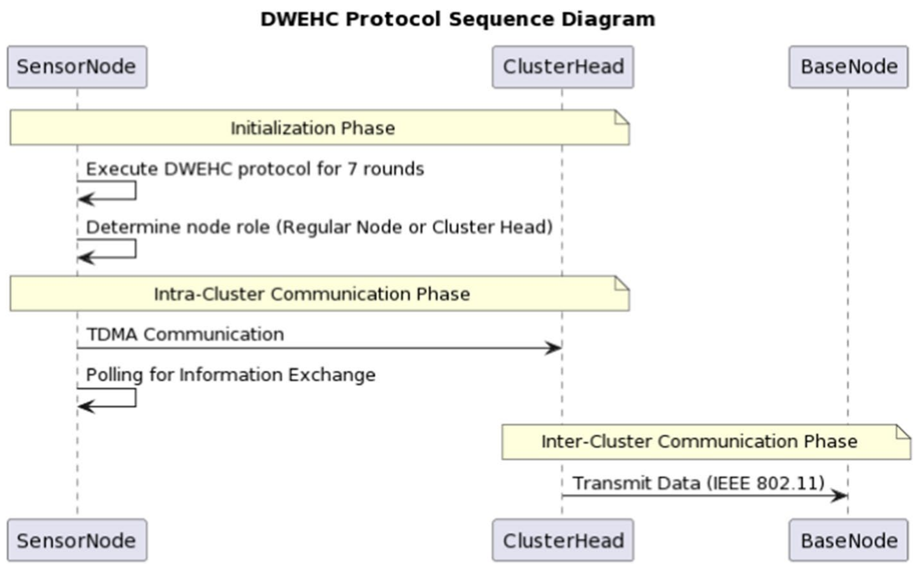


Fig. 2 DWEHC protocol sequence diagram

structure aids DWEHC in lowering power usage by aggregating data and localizing processing within each cluster. It promotes effective network resource use and makes load balancing across forks easier. By taking use of closer communication channels, the multi-level hierarchy facilitates efficient data routing. As DWEHC is distributed and does not rely on a central controller or global network knowledge, it enables scalability and adaptability in WSNs. Each node makes a separate decision based on its own knowledge, which increases the algorithm's flexibility and robustness under changing network circumstances.

DWEHC's efficiency in reducing power consumption, extending network lifetime, and enhancing WSN performance overall has been thoroughly examined and assessed. DWEHC is a useful partitioning or cluster technique for many WSN applications because it improves power efficiency, load balancing, and scalable operation in WSNs by using a weight-based approach and creating hierarchical clusters.

3.2.1 Hybrid Partitioning or Cluster Approach (HCA)

The main goals of Neamatollahi et al.'s 2011 [39] proposal of HCA was to increase system lifespan and minimize power usage. Figure 3 shows the whole HCA algorithm procedure.

The advantages of both centralized and distributed partitioning or cluster techniques are combined in the Hybrid Partitioning or Cluster Approach (HCA), a partitioning or cluster methodology created for Wireless Sensor Networks (WSNs).

By combining the advantages of both strategies, HCA seeks to maximize network performance, power efficiency, and scalability. A two-phase partitioning or clustering procedure is used by HCA. Initial cluster construction in the first phase is handled by a centralized entity known as the Cluster Head Election (CHE) node or base station. The CHE node gathers data from every sensor fork in the network and selects the first cluster head depending on variables like residual power, distance from the base station, or node density. This centralized phase offers global knowledge for effective cluster head selection and aids

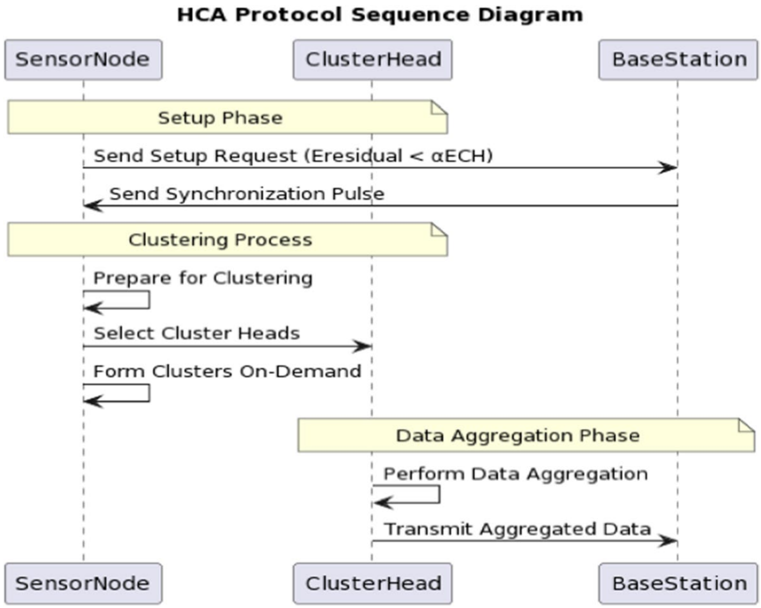


Fig. 3 HCA protocol sequence diagram

in the establishment of an initial hierarchical structure. The second stage of HCA adopts a distributed methodology after the initial clusters and cluster heads are built. The processes inside the cluster are coordinated at this phase by each cluster head in communication with its member fork. The cluster leaders act as local coordinators and manage resources, route traffic, and aggregate data for their own clusters. This distributed phase supports load balancing, scalability, and network situation adaptation.

HCA delivers various benefits in WSNs by combining the benefits of both centralized and distributed approaches:

a. Power Efficiency: The HCA's first centralized phase enables effective cluster head selection based on power-related factors. Selecting cluster heads with higher residual power will encourage balanced power use and increase network longevity. Following the dispersed phase, localized data aggregation is ensured, decreasing the need for energy-intensive long-distance transmission, and enhancing power efficiency.
b. Scalability: In large-scale WSNs, scalability is made possible by the mix of centralized and distributed stages in HCA. While the succeeding dispersed phase permits independent and localized activities inside each cluster, the early centralized phase makes it easier to create an initial hierarchical structure. This network's management and organization are made easier and more effective by the hierarchical and distributed architecture, which can accommodate more sensor nodes.
c. Fault Tolerance: By using a distributed strategy in the second phase, HCA gains the advantages of fault tolerance. A adjacent fork can independently elect a new cluster head inside its own cluster if a cluster head fails or becomes unavailable. By improving the network's robustness and tolerance to node failures, these self-healing capabilities.
d. Adaptability: The network can adjust to shifting network circumstances and requirements thanks to the mix of centralized and distributed phases in HCA. While the dis-

tributed phase enables dynamic modifications and reconfiguration inside clusters, the centralized phase gives global knowledge and aids initial deployment. HCA can adjust to changing power levels, node outages, or alterations in network architecture thanks to its flexibility.

In the context of WSNs, HCA has been investigated and assessed, proving its efficacy in attaining power efficiency, scalability, fault tolerance, and flexibility. HCA offers a complete solution for partitioning or clustering in WSNs by using a hybrid method that blends centralized and distributed techniques, addressing the particular needs and difficulties of such networks.

The setup phase, partitioning or cluster process, and data aggregation phase are all shown in this sequence diagram as well as their interactions with one another in the Hybrid Partitioning or Cluster Approach (HCA) protocol. It demonstrates how the base station, cluster heads, and sensor fork interact.

3.2.2 Power-Efficient Unequal Partitioning or Cluster (EEUC)

Li et al.'s "An Power-Efficient Unequal Partitioning or cluster Mechanism for Wireless Sensor Networks" [40] innovative technique was put out to solve the problem of hotspots in multi-hop routing when the cluster head is close to the sink station. The EEUC Protocol Sequence Diagram is shown in Fig. 4. Cluster heads close to the sink station have more system traffic than those farther away, which causes them to lose power more quickly.

To mitigate this hotspot problem, the authors introduced an algorithm that creates clusters of unequal sizes, with smaller clusters formed closer to the sink station to reduce power consumption during intra-cluster communication. The cluster formation process involves

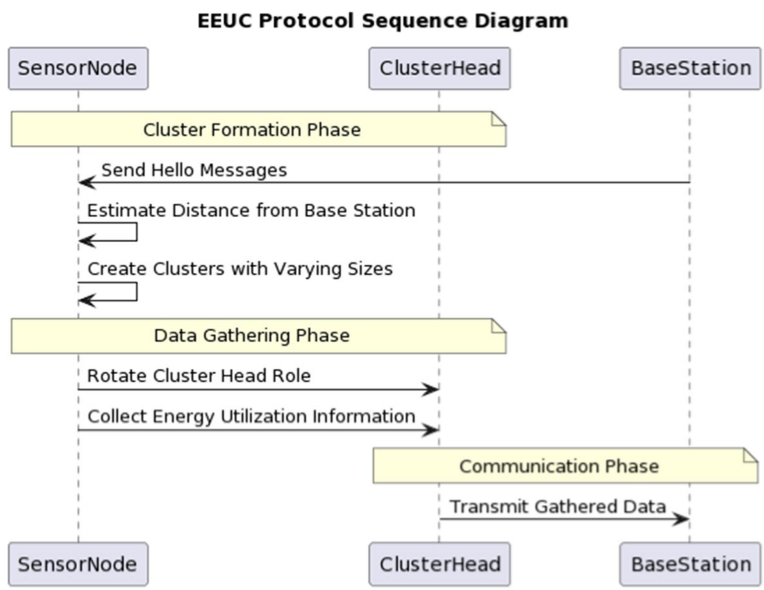


Fig. 4 EEUC protocol sequence diagram

the base station sending hello messages to all fork, enabling them to estimate their distance from the base station. This distance information is then utilized to create clusters of varying sizes. During the data gathering process, the cluster head rotates among the sensor fork and collects power utilization information throughout the system. As the distance from the cluster head to the sink station decreases, the cluster size is dynamically adjusted to be smaller. The authors conducted an analysis of the algorithm, demonstrating that the utilization of unequal cluster sizes improves the system's lifetime and achieves a more balanced power consumption compared to the LEACH and HEED protocols. In terms of communication complexity, the proposed algorithm, EEUC, exhibits a lower complexity of O(N) for cluster creation compared to HEED. Unlike the HEED protocol, EEUC minimizes the numerals of message iterations required by cluster heads. The probability of two fork becoming cluster heads is significantly reduced in EEUC. By reducing the interval between the first node's death and the last node's death in multi-hop routing within a partitioning or cluster method, EEUC effectively resolves the hotspot problem. The unequal partitioning or cluster mechanism in EEUC enhances the network lifetime when compared to the LEACH and HEED protocols. The proposed algorithm's main phases and interactions are depicted in the sequence diagram. It illustrates the cluster formation phase, data gathering phase, and communication phase, showcasing the interactions between sensor fork, cluster heads, and the base station.

3.2.3 Energy-Efficient Partitioning or Cluster Scheme (EECS)

Ye et al. [41] proposed a procedure for power-efficient and load-balanced partitioning or cluster in wireless sensor networks to facilitate periodic data collection. The protocol involves the selection of cluster heads based on their outstanding power, similar to the LEACH protocol. Figure 5 depicts the EECS Protocol Sequence Diagram.

During the cluster head selection process, candidate fork compete among themselves to become the head of the cluster. In the cluster formation phase, the sink station broadcasts hello messages to all fork, allowing them to compute their distance from the sink based on signal strength indicators.

In the cluster head selection stage, candidate fork are chosen with a probability T to become CANDIDATE fork. Once a node becomes a CANDIDATE node, it broadcasts a COMPETE_HEAD_MSG to other fork within its radio range (Rcompete). Upon receiving the COMPETE_HEAD_MSG, competing fork compare their remaining power with the received remaining power of the node. If the received residual power is greater, the competing node withdraws from the competition without transmitting a COMPLETE_HEAD_MSG. Otherwise, the node is selected as the cluster head. This process utilizes local neighborhood communication based on remaining power for cluster head selection.

In the cluster establishment stage, cluster head fork broadcast HEAD_AD_MSG to all base fork. Each base node decides whether to join the cluster based on the distance criteria specified in the received HEAD_AD_MSG. Cluster heads are evenly distributed across the system. Compared to LEACH, the proposed protocol, EECS, improves the network lifetime by 35%. EECS focuses primarily on the cluster setup algorithm and does not specifically address the data transmission phase. It is fully distributed, and cluster heads are randomly distributed in the system. The approach aims to evenly distribute the load among cluster heads using a weighted function. The competition process among cluster heads is localized, eliminating the need for iterations and reducing message overhead. Simulation results demonstrate that EECS increases the system lifetime by 135% compared to the

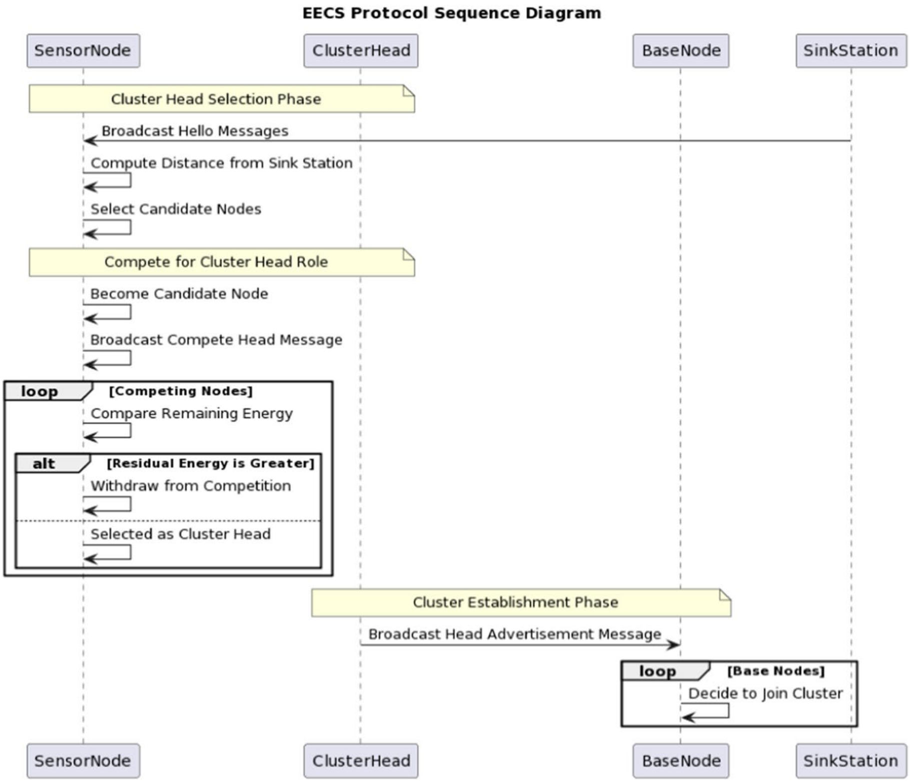


Fig. 5 EECS protocol sequence diagram

LEACH protocol, with power utilization at 93% in EECS and 53% in LEACH. The control message complexity of the process is O(N), where N is the numerals of fork in the system. In each Rcompete, there is at most one cluster head. The sequence diagram represents the main phases and interactions in the Power Efficient Partitioning or cluster Scheme (EECS) protocol, including the cluster head selection phase, the competition for the cluster head role, and the cluster establishment phase. It illustrates the interactions between sensor fork, cluster heads, base fork, and the sink station.

3.3 Clustering Routing Protocol

A communication protocol designed specifically for wireless sensor networks (WSNs), in which sensing units are grouped into clusters, is known as a division or grouping routing system. The network's scalability, power efficiency, and overall system performance are all enhanced by this method. Units are grouped into clusters in these protocols, and each cluster has a designated cluster head (CH) or cluster leader who is in charge of managing communication between the cluster and the base station (BS).

A dividing or grouping routing protocol's major objective is to promote effective data consolidation, routing, and communication inside the network by making use of the clusters' hierarchical structure. Data from member units are consolidated by cluster leaders, who also decide on routing and send data to the base station. The routing protocol reduces power consumption by splitting or grouping in order to reduce long-distance transfers and enable localized data processing inside the clusters.

For WSNs, a number of division or grouping routing methods have been proposed, each with unique properties, benefits, and drawbacks. Several frequently used procedures are as follows:

a. LEACH (Low-Power Adaptive Clustering Hierarchy) is a well-known and commonly used routing technology for grouping or splitting traffic. It functions in a randomized fashion, allowing sensor units to alternately serve as cluster leaders to divide the network's power demand more equitably [1].
b. HEED (Hybrid Power-Efficient Distributed Clustering): HEED is a distributed grouping or splitting protocol that creates clusters by taking into account both node proximity and remaining power. Its goal is to improve load balancing and extend the lifespan of the network by dynamically choosing cluster leaders [37].
c. A chain-based grouping technique called PEGASIS (Power-Efficient Gathering in Sensor Information Systems) uses units to transfer data to the base station. By leveraging data consolidation and targeted communication across the chain, electricity consumption is decreased.
d. SEP (Stable Election Protocol): SEP introduces the idea of stable units, which are units with greater remaining power and longer network connectivity. SEP is a division or grouping protocol. To increase network stability and lengthen the network's lifespan, stable units are elected as cluster leaders [3].
e. Threshold-sensitive Power Efficient Sensor Network Protocol (TEEN): TEEN is a grouping or division protocol that was created especially for event-driven applications. To assess whether a unit should become a cluster leader or a follower, it uses several thresholds [42].

These dividing or grouping routing protocols offer different trade-offs between power efficiency, network scalability, and system performance. The choice of an appropriate protocol depends on the specific requirements of the application, the characteristics of the network, and the power limitations of the wireless sensor network.

3.3.1 Sequence Diagram for Cluster Routing Protocol

The selection and setup of a partitioning or cluster routing protocol is shown in this sequence diagram as an interaction between the user, system, partitioning or cluster routing protocol, and network parameters. The system fetches the available protocols, presents them to the user, and then takes the user's option after receiving their request to choose a protocol. Following the selection of the protocol, the system gets the network parameters, determines the best configuration, and sets the configuration for the partitioning or cluster routing protocol. The partitioning or clustering procedure finally starts. The Cluster Routing Protocol's sequence diagram is shown in Fig. 6.

3.4 Network Parameters

The construction and optimization of communication networks heavily depend on network characteristics. The performance and effectiveness of network operations are significantly influenced by several factors, including bandwidth, latency, transmission power,

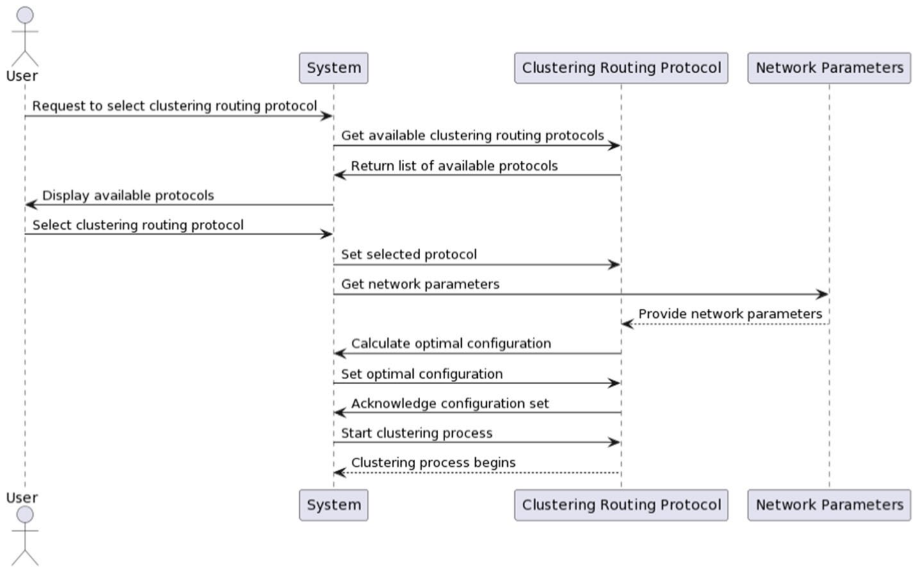


Fig. 6 Sequence diagram for cluster routing protocol

and routing protocols. For optimum network performance, resource usage, and overall system operation, network parameters must be managed correctly. The action of controlling a certain network parameter is shown in the activity diagram. The process of managing the parameter, from defining it through storing it for later usage, is shown visually in this figure. Decision points and conditional branching are included in the graphic, emphasizing the value of taking into account current parameters, computing new ones when necessary, and managing any mistakes throughout the calculation or adjustment process.

Network administrators or system engineers can systematically control network parameters by following the activity diagram (Fig. 7), assuring their correctness and relevance to the network's present status. This method makes it possible to adjust and optimize parameter settings in a timely manner while still maintaining the network's stability and

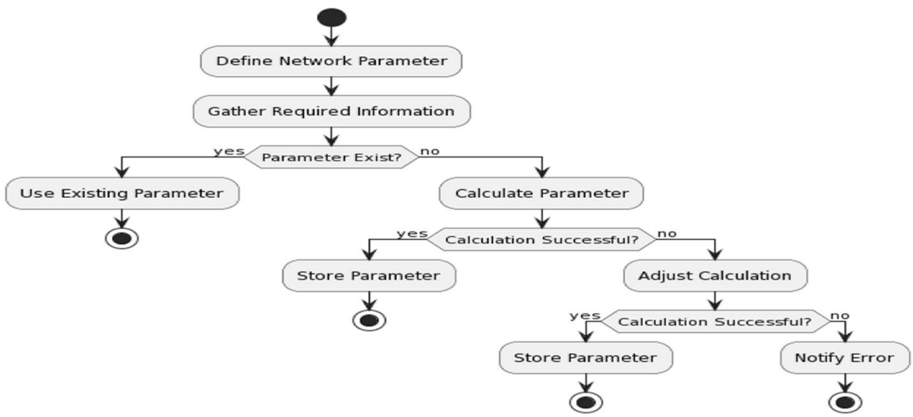


Fig. 7 Process of managing network parameters

effectiveness. It is important to note that the supplied activity diagram is only an example and may be altered to reflect the precise network parameter management procedure in your situation. Depending on the complexity of the parameter and the related management procedures, additional stages, decision points, or actions may need to be introduced. The activity diagram, in general, provides a visual reference for successfully controlling network settings, ensuring that the network runs at its optimal performance by maintaining.

Figure 7's activity diagram (Fig. 7) depicts the steps involved in controlling network settings. The first step is to define the network parameter and collect the relevant data. Then, it determines if the parameter is present previously. If so, the current parameter is applied. If not, a computation is performed on the parameter, and if it is successful, the parameter is saved. The figure illustrates the option of modifying the computation and trying again if it fails. The parameter is saved if the adjusted computation is successful. If not, an error is reported. After the parameter has been successfully saved or an error has been detected, the diagram is finished.

3.5 Cluster Head Selection Algorithms

A significant topic covered in the current literature on wireless sensor networks (WSNs) is the cluster head selection technique. In different research investigations, several techniques for selecting cluster heads within a network have been developed. These algorithms seek to maximize communication dependability, energy efficiency, and network performance. The Low-Energy Adaptive Clustering Hierarchy (LEACH) method is one often discussed algorithm in the literature. LEACH chooses cluster heads in a randomized manner, allowing sensor nodes to alternately assume the function of a cluster head. The lifespan of the network is increased by this distributed algorithm's guarantee of equitable energy distribution. The Stable Election Protocol (SEP) is a different algorithm that has received a lot of research. Stable nodes, which have greater residual power and greater network connection, are a concept introduced by SEP. To increase network stability and lengthen network lifetime, these stable nodes are chosen as cluster heads. Other cluster head selection methods have been put out in the literature in addition to LEACH and SEP. Each algorithm has distinct qualities and benefits of its own. To choose the best cluster heads, certain algorithms take into account factors including node proximity, residual power, communication range, and network connectivity. Additionally, machine learning-based strategies for cluster head selection have been investigated by researchers. Based on different inputs, including node features, network circumstances, and energy levels, these algorithms use data-driven methodologies to make informed judgments on which nodes should become cluster chiefs. A cluster head selection algorithm is chosen based on the WSN application's unique needs, network architecture, and energy limitations. To enhance cluster head selection and maximize network performance in wireless sensor networks, researchers keep investigating and creating new algorithms.

Numerous cluster head selection algorithms that make use of optimization approaches are presented in the literature on wireless sensor networks (WSNs). These algorithms use optimization methods to increase system performance overall, increase network lifetime, and improve network efficiency. The Particle Swarm Optimization (PSO) [21] technique is one often discussed optimization approach used for cluster head selection. PSO is a population-based optimization method that draws its inspiration from the social behavior of fish schools and bird flocks. It uses a swarm of particles to iteratively search the solution space for the best cluster head candidates based on fitness functions that take network connection, residual energy, and distance to the base station into account. The Genetic approach (GA) is another optimization approach used for cluster head selection in WSNs. GA is a metaheuristic algorithm that draws inspiration from genetics and natural selection. In order to develop the population toward better answers, it entails establishing a population of possible solutions and executing selection, crossover, and mutation procedures repeatedly.

Ant Colony Optimization (ACO) [26] techniques have also been investigated for WSN cluster head selection. For the purpose of determining the best routes and answers, ACO algorithms mimic the foraging activity of ants. ACO algorithms enable nodes to deposit pheromone trails signifying their suitability as cluster head candidates in the context of cluster head selection. These pheromone trails are used by other nodes to choose cluster heads based on criteria including residual energy, base station distance, and data aggregation capabilities. For cluster head selection in WSNs, researchers have also looked at different optimization methods as the Firefly Algorithm (FA) [4], Grey Wolf Optimizer (GWO) [19], and Artificial Bee Colony (ABC) algorithm. These algorithms go through a variety of fitness functions and network characteristics to identify the best candidates for cluster heads. A particular optimization technique for cluster head selection is chosen based on the WSN's features, energy limitations, network architecture, and the application's unique goals. The performance and energy efficiency of wireless sensor networks are being improved by researchers as they continue to investigate and create novel optimization-based cluster head selection methods.

3.5.1 Sequence Diagram for Cluster Head Selection Algorithms

This flowchart (Fig. 8) shows how a sensor node interacts with the cluster head selection algorithm, the cluster head routing protocol, and the cluster head during the selection process. The request to choose a cluster head is made by the sensor node. The cluster

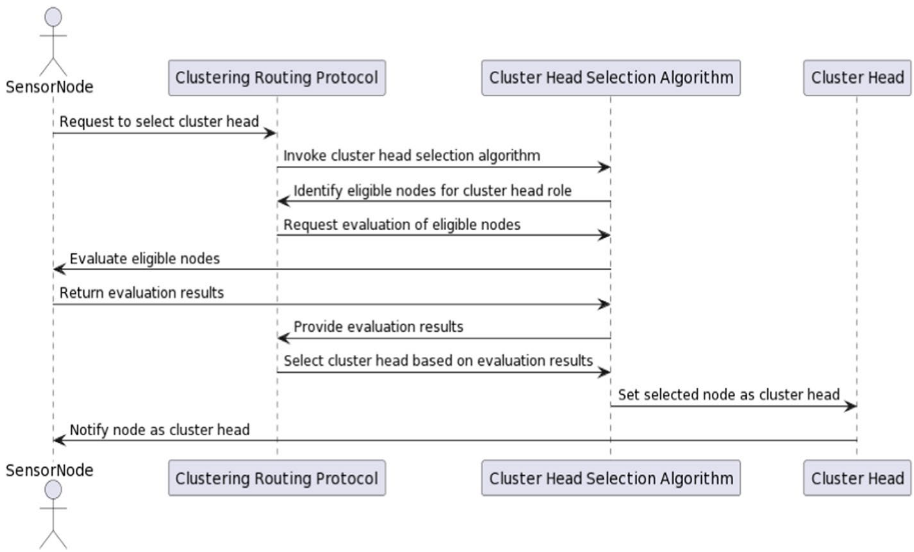


Fig. 8 Sequence diagram for cluster head selection algorithms

head selection method, which chooses an appropriate fork for the cluster head role, is called upon by the partitioning or cluster routing protocol. The eligible fork is then assessed, and the outcomes are given to the cluster routing protocol or partitioning protocol. The partitioning or cluster routing protocol chooses a node as the cluster head based on the findings of the evaluation. After that, the chosen node is designated as the cluster head, and the cluster head informs the sensor node of its new position.

3.6 Cluster Formation Algorithm

Various cluster formation algorithms that utilize optimization approaches are covered in the literature on wireless sensor networks (WSNs). In order to create effective and ideal cluster formations inside WSNs, these algorithms take into account variables including energy efficiency, network scalability, and overall system performance. The K-means method is a well-known optimization technique for cluster building. Based on the sensor nodes' closeness to centroid points, this algorithm clusters the sensor nodes into groups using an iterative optimization technique. The K-means method guarantees that nodes within each cluster are tightly connected while maintaining a suitable separation between clusters by minimising the intra-cluster distance and increasing the inter-cluster distance [34].

The Genetic approach (GA) is another well researched optimization approach for cluster creation. To find the best cluster forms, GA imitates natural evolution and genetics. It iteratively enhances the clustering quality using selection, crossover, and mutation procedures on a population of candidate solutions. The optimal cluster topologies are determined by fitness functions in GA-based cluster formation algorithms by considering variables like node density, residual energy, and communication range. Another optimization approach that has been used for cluster formation in WSNs is Particle Swarm Optimization (PSO). PSO searches the search space iteratively using a swarm of particles to identify the best cluster arrangements [26].

Every particle acts as a prospective cluster formation, and each one modifies its position in accordance with its prior knowledge as well as that of the swarm's best solutions. PSO-based cluster formation algorithms seek to create evenly dispersed and effective clusters by minimizing the total intra-cluster distance and maximizing inter-cluster distances.

Additionally, cluster formation in WSNs has been studied using Ant Colony Optimization (ACO) techniques. For the purpose of determining the best routes and answers, ACO algorithms mimic the foraging activity of ants. ACO algorithms enable nodes to leave behind pheromone trails that indicate whether they would make good cluster heads or cluster members in the context of cluster formation. Nodes create clusters with balanced energy consumption and ideal connection by following these pheromone trails. For cluster formation in WSNs, several optimization methods have also been investigated, including the Firefly Algorithm (FA), Grey Wolf Optimizer (GWO), and Artificial Bee Colony (ABC) algorithm. These methods optimize cluster configurations based on multiple fitness factors and network characteristics using novel techniques and mathematical models. The WSN application's unique needs, network architecture, energy restrictions, and intended outcomes all play a role in the decision of which optimization method to use for cluster creation. To increase the effectiveness and performance of wireless sensor networks, researchers are looking into and creating new optimization-based cluster formation methods [26, 34–36, 43].

3.6.1 Sequence Diagram for Cluster Formation

The interactions that take place between a sensor node, the partitioning or cluster routing protocol, the cluster formation algorithm, the cluster head, and the cluster members are shown in this flowchart (Fig. 9). The request to start cluster formation is made by the sensor node. The cluster formation algorithm, which chooses suitable forks for cluster creation, is invoked by the partitioning or cluster routing protocol. The partitioning or cluster routing protocol is then given the evaluation results of the eligible fork. The partitioning or cluster routing protocol chooses fork as cluster members based on the evaluation findings. The chosen fork joins the cluster whose cluster head is in charge.

3.7 A Taxonomy of Approaches to Power Saving in Sensor Networks

Applications including environmental monitoring, surveillance, and healthcare all depend on sensor networks. The limited power resources accessible to sensor fork, however, provide a significant challenge to the robustness and effectiveness of these networks [44]. In response, researchers have developed a number of techniques to reduce power consumption and improve sensor network performance [45]. The goal of this study is to provide a thorough taxonomy of these techniques, giving a well-organized overview of the many tactics used in this field.

The table below (Table 1) outlines the various approaches to power saving in sensor networks and provides a brief description of each approach.

Researchers and practitioners can better understand the power-saving methods available for sensor networks by classifying these approaches [37]. Additionally, this taxonomy can be a useful tool for determining the best techniques based on the requirements of a certain application and the limitations of the network. In conclusion, the proper functioning and

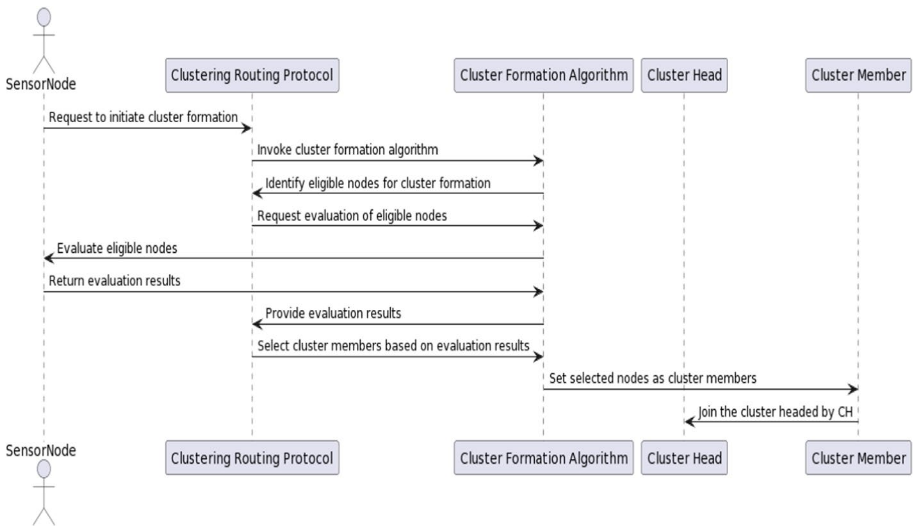


Fig. 9 Sequence diagram for cluster formation

Table 1 Taxonomy of approaches to power saving in sensor networks

Approach	Description
Duty cycling	Periodic sleep: the fork alternates between periods of activity and sleep
	Random sleep: fork out random naps of a particular length
	Adaptive sleep: forks change their sleep pattern according on the state of the network
Data aggregation	Spatial aggregation: to cut down on transmission, aggregate data from neighboring forks
	Temporal aggregation: to lower transmission frequency, fork aggregate data over time
	Compressive sensing: to minimize transmission size, forks capture compressed data
Network topology control	Node placement: the best location for the fork to save energy
	Node selection: selecting a subset of fork to perform sensing and routing
	Data routing optimization: finding power-efficient routes for data transmission
Communication protocols	Low-power listening: fork listen for communication only during specific time slots
	TDMA/CDMA: time or code division multiple access to reduce collisions
	Data compression: reducing data size before transmission
Power harvesting	Solar power: harvesting power from solar panels
	Kinetic power: harvesting power from node movement or vibrations
	RF power: harvesting power from ambient RF signals
Power-efficient MAC protocols	IEEE 802.15.4: low-power MAC protocol for wireless sensor networks
	SMAC: sensor medium access control protocol with sleep scheduling
	B-MAC: a synchronization and duty-cycling MAC protocol with beacon support
Routing protocols	LEACH: cluster-based routing using low-power adaptive partitioning or cluster hierarchy
	Threshold-sensitive power efficient sensor network protocol is referred to as TEEN
	Ad hoc on-demand distance vector, or AODV, is used for on-demand routing
Power-efficient data storage and processing	Data compression: reducing the amount of space and time needed for processing
	Data fusion: combining redundant data to speed up transmission and minimize processing
	Data caching: reducing communication by storing often accessible data locally

durability of sensor networks depend on the efficient use of power resources. Researchers may develop ways to minimize power usage, improve network performance, and increase the lifespan of sensor networks by utilizing power-saving techniques from a variety of categories [38]. This taxonomy is a useful resource for examining the many different methods for power reduction in sensor networks.

3.8 Taxonomy of Approaches to Power-Efficient Cluster Head Selection in Wireless Sensor Networks Using Nature-Inspired Algorithms

A large number of sensor nodes work together in Wireless Sensor Networks (WSNs) to collect and relay data to a centralized base station. The selection of cluster heads, who are responsible for collecting and sending data in WSNs, is a crucial factor in lowering power consumption and increasing network lifetime [39].

To address this issue, researchers have looked at the usage of nature-inspired algorithms for WSN cluster head selection that is power-efficient. Several strategies that are inspired by nature are described in this study to enhance the power efficiency of cluster head selection in WSNs [46]. A taxonomy of numerous strategies that employ algorithms drawn from nature for power-efficient cluster head selection in WSNs is presented in the table below (Table 2). Each strategy is briefly explained, emphasizing its unique characteristics and advantages.

3.9 Nature-Inspired Algorithms Used in Wireless Sensor Networks (WSN)

WSNs are widely used for a variety of purposes, including as environmental monitoring, smart cities, and industrial automation. WSNs face a serious challenge in increasing system performance while saving power since sensor forks frequently have limited power resources. In order to solve this issue, researchers have looked into nature-inspired algorithms, which draw inspiration from natural phenomena and processes [47].

Table 2 Taxonomy of approaches to power-efficient cluster head selection in WSNs using nature-inspired algorithms

Taxonomy category	Description
Nature-inspired algorithm	The nature-inspired algorithm used for cluster head selection, such as Ant Colony Optimization or Particle Swarm Optimization
Fitness function	The objective function used to evaluate the fitness of each sensor node as a potential cluster head
Power awareness	Whether the algorithm takes into account the power level of sensor fork in the selection process
Communication overhead	The impact of cluster head selection on the overall communication overhead in the network
Cluster formation	How the algorithm forms clusters and assigns non-cluster head fork to their respective cluster heads
Cluster head rotation	Whether the algorithm supports dynamic rotation of cluster heads to distribute the power consumption evenly
Network lifetime	The effect of the approach on the overall network lifetime in terms of power efficiency
Scalability	How well the approach scales with increasing network size or numerals of sensor fork
Simulation environment	The simulation tool or platform used for evaluating the performance of the approach

Nature-inspired algorithms simulate the behavior and dynamics of natural systems, such as the foraging activity of ants, the flocking behavior of birds, or the evolutionary principles of genetic inheritance. By imitating these natural processes, these algorithms provide innovative and workable solutions to significant WSN problems including power-efficient cluster formation, routing, localization, and data aggregation. The focus of this research is on an investigation of nature-inspired algorithms that have been successfully applied in WSNs [48]. These algorithms employ natural intelligence to enhance the effectiveness of WSN operations in a variety of areas, including as power consumption, network connectivity, scalability, and fault tolerance. The Nature Inspired method utilized in WSN is shown in Table 3.

The taxonomy of nature-inspired algorithms used in WSNs includes well-known methods such as Particle Swarm Optimization, Ant Colony Optimization, Genetic Algorithms, Artificial Bee Colony, Firefly Algorithm, Grey Wolf Optimizer, Cuckoo Search, Bat Algorithm, Whale Optimization Algorithm, Harmony Search, and many more. The necessity of comprehending each algorithm's fundamental concepts and prospective applications is emphasized by the fact that each algorithm brings its own distinct advantages and qualities to bear on certain WSN difficulties. Researchers and practitioners may learn a great deal about these nature-inspired algorithms' applicability, performance, and limits in various WSN settings by examining and contrasting them [49]. This information may direct the selection and creation of suitable algorithms for particular WSN applications, eventually assisting in the creation of reliable and power-efficient wireless sensor networks [50].

Table 3 Nature inspired algorithm used in WSN

Algorithm	Description
Ant Colony Optimization	Inspired by the foraging behavior of ants, it uses pheromone trails to find optimal paths or solutions in the network
Particle Swarm Optimization	Mimics the collective behavior of a group of particles moving through a problem space to find optimal solutions
Genetic Algorithm	Emulates the process of natural selection and evolution to iteratively improve solutions through crossover and mutation
Artificial Bee Colony	Based on the foraging behavior of honeybees, it employs employed bees, onlooker bees, and scout bees for exploration
Firefly Algorithm	Inspired by the flashing patterns of fireflies, it uses the attractiveness of fireflies to optimize solutions
Grey Wolf Optimizer	Modeled after the hierarchical social structure of grey wolves, it features alpha, beta, delta, and omega wolf agents
Cuckoo Search	Draws inspiration from the brood parasitism of cuckoo birds to search for optimal solutions by replacing eggs in nests
Bat Algorithm	Inspired by the echolocation behavior of bats, it uses frequency and loudness modulation to optimize solutions
Whale Optimization Algorithm	Inspired by the cooperative hunting behavior of humpback whales, it uses encircling and bubble-net hunting strategies
Harmony Search	Mimics the process of creating musical harmony to find optimal solutions through improvisation and adjustment

3.9.1 Taxonomy of Approaches to Energy-Efficient Cluster Formation in Wireless Sensor Networks Using Nature-Inspired Algorithms

Applications for wireless sensor networks (WSNs) include environmental monitoring, industrial automation, and medical care. However, due to the constrained power capacities of sensor fork in WSNs, attaining power efficiency is a significant problem. Effective cluster formation, aided by algorithms inspired by nature, has emerged as a possible response to this problem. These methods maximize cluster formation in WSNs by taking cues from nature, with the goals of balancing power consumption, extending network lifetime, and enhancing system performance in general [40].

A taxonomy of strategies for power-efficient cluster creation in WSNs utilizing algorithms inspired by nature is shown in the following Table 4. The categories of the taxonomy provide a thorough understanding of the various aspects of cluster formation, such as the selection of algorithms inspired by nature, fitness functions for evaluating node fitness, power-efficiency considerations, cluster head selection techniques, mechanisms for cluster formation, cluster size determination, cluster stability, communication overhead, load balancing strategies, impact on network lifetime, and others [41].

By looking at these taxonomy categories, researchers and practitioners may gain greater insight into the various methods employed to maximize cluster formation in WSNs [51]. This taxonomy is an effective tool for understanding the key components and aspects involved in the development of power-efficient clusters, aiding in the selection of the most suitable approach depending on specific network needs and objectives.

By classifying nature-inspired algorithms, researchers and practitioners may get more knowledge about different approaches for power-efficient cluster head selection in WSNs [52, 53]. It is much simpler to select the optimal algorithm based on the requirements of a particular application and the characteristics of the network when using this taxonomy to understand the benefits and drawbacks of each technique [54]. In order to maximize power consumption and lengthen the lifespan of wireless sensor networks, it is crucial to make the right choice of power-efficient cluster heads. The efficacy of cluster head selection can be increased by using techniques inspired by nature. In order to find the most effective algorithm for WSNs' power-efficient cluster head selection, researchers can explore and select from the taxonomy's comprehensive review of several approaches [42].

In the research that is being presented, a brand-new cluster building method for Wireless Sensor Networks (WSNs) is introduced. This method addresses many network optimization difficulties. The originality of the article resides in its multi-phase methodology, which begins with the initialization of network parameters like the number of sensor nodes, the communication range, and the energy levels. Notably, the method dynamically positions nodes in the network area intelligently, ensuring sufficient starting energy and communication range for the best performance [55]. Nodes advertise their eligibility to possible cluster leaders while also evaluating their candidacy based on variables like energy level and proximity to the base station during the cluster head selection process. By reducing transmissions that use a lot of energy, this dynamic selection procedure increases energy efficiency.

Another important addition is cluster maintenance, which is accomplished with reliable processes. To keep members informed and preserve cluster integrity, cluster leaders issue brief announcements, which lower communication costs and delay. The end-to-end delay and overall network throughput are enhanced by this method [56]. The algorithm also demonstrates versatility to a variety of application demands, switching between real-time reporting and energy-efficient data aggregation with ease, providing prompt data delivery.

Table 4 Taxonomy of strategies for energy-efficient cluster creation in WSNs utilizing algorithms inspired by nature

Taxonomy category	Description
Nature-inspired algorithm	This category emphasizes the nature-inspired approach used in wireless sensor networks to generate clusters. Popular options for optimizing cluster formation include ant colony optimization and particle swarm optimization, which take their cues from natural events
Fitness function	During the cluster building process, the fitness function evaluates the adequacy of each sensor node. It assesses a number of factors, including connection, power levels, and node proximity, to determine the fork's suitability as prospective cluster heads
Power efficiency	This category focuses on strategies that give the cluster formation phase's power efficiency top priority. By carefully choosing cluster heads and arranging the network topology appropriately, the approaches used seek to reduce power consumption, increase network lifetime, and maximize overall system performance
Cluster head selection	Cluster head selection describes the method of choosing appropriate forks to serve as cluster heads inside the wireless sensor network. Different methods use various variables, like as power levels, base station distance, or node density, to choose the best candidates for cluster head locations
Cluster formation	The process through which clusters are created in a network is referred to as cluster creation. Based on predetermined criteria or algorithms, it entails assigning non-cluster head forks to their corresponding cluster heads. To enable effective cluster formation, these criteria take into account variables including communication range, power limitations, and network structure
Cluster size	The cluster size category investigates how methods choose the proper fork numbers to be given to each cluster. Different approaches may be used; some try to create equal-sized clusters to spread the workload equally, while others dynamically change cluster sizes in response to factors like power availability or node density
Cluster stability	Cluster stability is essential for long-term, dependable network performance. In order to increase the stability and resilience of the created clusters, methods in this category concentrate on reducing cluster head changes or putting in place mechanisms to deal with node failures, rearrangement, and sustaining connection
Communication overhead	Cluster formation's effect on the total communication burden of the network is taken into account by communication overhead. During the cluster formation process, efficient techniques strive to avoid needless communication and control messages, optimize resource use, and decrease network congestion
Load balancing	To avoid individual forks from being overloaded or using up their power resources more quickly, load balancing solutions divide the burden across the cluster heads in an equitable manner. These techniques aim to balance out power usage across the network, extending its lifespan and increasing overall effectiveness
Network lifetime	This classification evaluates how the cluster creation strategies affect the network's total lifespan. By saving energy, effectively allocating resources, and preventing early node failure, power-efficient cluster formation approaches seek to optimize the network's usefulness and operating time
Scalability	Scalability describes how successfully a method adjusts to network growth, taking into account a growing number of sensor nodes or a growing network size. Even in massive wireless sensor networks, scalable cluster formation technologies demonstrate the potential to maintain effective operations and power saving

Table 4 (continued)

Taxonomy category	Description
Simulation environment	The software program or platform used to examine and assess the effectiveness of the cluster formation procedures is referred to as the simulation environment. NS-2, OMNeT + +, MATLAB, or customized WSN simulators are common simulation tools used in research studies that offer a controlled and repeatable environment for studying and comparing various methodologies

The unique contribution of the article also consists of a performance evaluation stage, which comprises simulations and tests to evaluate metrics like network coverage, energy consumption, cluster formation speed, and data transmission effectiveness. The suggested method's superiority in a variety of parameters may be seen when compared to existing algorithms like LEACH, SEP, and HEED. Notably, the method regularly beats various alternatives in terms of the number of active nodes, energy used, end-to-end delay, and throughput. This thorough performance comparison shows how useful and efficient the suggested method is.

In conclusion, the suggested algorithm presents dynamic cluster formation, flexible cluster head selection, dependable cluster maintenance, application flexibility, and thorough performance evaluation. Together, these contributions tackle issues with network performance optimization, effective data aggregation, and energy conservation. The comparison with current algorithms reinforces the paper's originality and practical value in improving WSN effectiveness and performance.

3.10 Simulation Setup

Due to its capacity to observe and gather information from the physical world, wireless sensor networks (WSNs) have attracted substantial attention in several sectors. WSNs must be effectively managed and optimized to function reliably and sparingly. Several factors, including network settings, partitioning or cluster protocols, and power usage, are significant in this situation.

Wireless sensor networks (WSNs) have gotten a lot of interest because of their ability to watch and collect data from the real world in a variety of fields. To operate reliably and with the least amount of power consumption, WSNs must be efficiently controlled and optimized. In this case, several factors are important, including network configuration, partitioning or cluster protocols, and power consumption [39]. Numerous important WSN-related issues, such as simulation setup, network parameters, partitioning or clustering techniques, and power calculations are covered in the introduction to the content above. These factors when combined have an impact on our understanding and study of WSN performance, system lifespan, and resource usage [7].

In the simulation setup, MATLAB is used as a simulation tool to test various protocols and procedures in heterogeneous WSNs. The simulations are built on the selected network field size, node count, and packet size. Power metrics that describe the power consumption characteristics during data transmission and reception are also considered, including Eelec, Efs, and Eamp. The D0 threshold interval and the E0 initial power of normal fork, combined with these parameters, further mold the simulation environment. Different partitioning or cluster techniques are evaluated in this arrangement to improve network performance and power efficiency. Each protocol, which addresses certain difficulties and objectives of partitioning or cluster in WSNs, is investigated, including Hybrid Power-Efficient Distributed Partitioning or cluster (HEED), Distributed Weight-Based Power-Efficient Hierarchical Partitioning or cluster (DWEHC), and Hybrid Partitioning or cluster Approach (HCA). These protocols use various processes and algorithms for load balancing, cluster creation, and cluster head selection. The content also covers how important WSN-related metrics and parameters are calculated. The numerals of dead and living fork, the choice of the cluster head, packet transmission to the base station and cluster head, power dissipation, and needs for cluster head election and cluster formation are all calculated using the specified formulas. The performance, power use, and efficiency of WSNs may be better understood by researchers and practitioners by taking into account certain simulation settings, network characteristics, partitioning or cluster procedures, and calculation formulae. This knowledge makes it possible to optimize network performance, resource management, and decision-making for a range of applications in industries including industrial automation, healthcare, and environmental monitoring.

A overview of the simulation setup and the performance metrics taken into account while assessing the effectiveness of various protocols in the heterogeneous wireless sensor network (HWSN) is given in the table. It emphasizes the important variables and traits that were utilized in the simulation, such as the numerals of sensor fork, network size, location of sink stations, beginning power levels of fork, biased amount of forecasting power, and wireless channel concerns. It also describes the performance measures, such as the numerals of fork that are still active, their power levels, and the numerals of fork that left in the first round, that are used to evaluate the effectiveness of the protocols. These parameters make it possible to assess node stability, power consumption trends, and node survival (Table 5, 6, 7).

3.10.1 Parameters Needed for the Calculation

Calculations and formulae are used by wireless sensor networks (WSNs) to maximize network efficiency and power use. The numerals of living and dead fork, the choice of the cluster head, and packet transmission metrics are important computations. The power needs for cluster head election and creation as well as power dissipation every round is determined by power-related formulae. In WSNs, these computations direct resource allocation and decision-making. Researchers and professionals may learn more about network

Table 5 Parameter chosen

S. no	Parameters
1	Numerals of dead fork
2	Numerals of alive fork
3	Cluster head selection
4	Packets sent to base station
5	Packets sent to cluster head
6	Power dissipated per round
7	Power required per round for cluster head election
8	Power required per round for cluster formation
9	E0 (original power of normal fork)
10	Power consumption per node for cluster formation
11	Power consumption per node for cluster head election

Table 6 Simulation parameter

Parameters	Values
Network field	100 × 100
Numerals of fork	500
Packet size	4000 bits
Eelec (power for transmitting 1 bit)	50 nJ/bit
Efs (free space power)	10 nJ/bit/m^2
Eamp (amplification power)	0.0013 pJ/bit/m^4
EDA (power for data aggregation)	5 nJ/bit/signal
D0 (threshold interval)	70 m
E0 (original power of normal fork)	0.5 J

dependability, power efficiency, and system longevity by using these calculations. The design and optimization of WSNs for diverse applications is made easier by these calculations. In wireless sensor networks, knowing and using these algorithms improve network efficiency and resource use.

To calculate the numerals of neighboring fork in a wireless sensor network, typically need to consider the communication range or transmission range of the sensor fork. Here is a general approach to calculate the numerals of neighboring fork:

a. Determine the communication range (R_comm) of each sensor node in the network. The communication range defines the maximum distance up to which a node can transmit and receive signals.
b. For each sensor node, measure the distance to all other fork in the network.
c. Compare the distance between each pair of fork with the communication range (R_comm). If the distance is less than or equal to the communication range, consider the fork as neighbors.
d. Count the numerals of fork that satisfy the condition in step 3 for each sensor node. This will give you the numerals of neighboring fork for each node in the network.
e. Here's a formula to calculate the numerals of neighboring fork:
f. Numerals of Neighboring Fork = Count(dist < = R_comm)
g. Where `dist` represents the distance between each pair of fork, and `R_comm` is the communication range.

3.10.2 Proposed Algorithm for an Energy Conservation Model in Wireless Sensor Networks

There are multiple phases in the suggested cluster building technique for Wireless Sensor Networks (WSNs). The network's settings, such as the number of sensor nodes, communication range, and energy levels, are first initialized. In order to provide sufficient starting energy levels and communication range with other nodes, the nodes are then strategically placed throughout the network region. Following that, nodes evaluate their candidacy based on factors like energy level and distance to the base station during the cluster head selection process. A node advertises its eligibility to other nodes if it meets the requirements to be a possible cluster leader. Nodes then receive these messages and compare them in order to decide which cluster head is the nearest or most

Table 7 Comparison of existing algorithm with proposed algorithm

Parameter	Stable election protocol (SEP)	Hybrid energy-efficient distributed (HEED)	LEACH	Proposed algorithm
Goal	Select stable cluster heads	Achieve energy efficiency and load balancing	Form stable clusters and save energy	Reduce energy consumption and improve performance
Energy Balancing	Emphasizes balancing energy consumption among nodes	Considers both residual energy and node proximity for load balancing	Balances energy usage within clusters	Reduces energy consumption and avoids frequent retransmissions
Cluster head selection	Based on residual energy and distance to base station	Probability-based selection considering energy and node proximity	Randomized selection of cluster heads	Intelligent selection of cluster heads to minimize energy consumption
Stability	Aims to prolong network lifetime through stable cluster head selection	Promotes energy efficiency and load balancing for network longevity	Relies on periodic cluster head re-election	Enhances stability and reduces the need for frequent re-election
Traffic handling	Lacks specific methods to handle higher traffic loads	Effectively handles high network traffic	Not designed to handle high traffic loads	Able to handle increased traffic without significant performance degradation
Performance	May face limitations with higher traffic loads	Performs well under high traffic scenarios	Performance may degrade with high traffic	Improves performance and reduces energy consumption
Throughput	Dependent on network conditions and stability	Balances energy consumption and increases throughput	Throughput may decrease under high traffic	Enhances throughput even with increased traffic
Scalability	Suitable for various network sizes	Performs well with an increasing number of nodes	Scales to a limited number of nodes	Scalable to larger networks without sacrificing performance
Application	Best for stability and energy balancing concerns	Ideal for energy efficiency and load balancing requirements	Widely used in clustering-based WSNs	Offers improved energy efficiency and performance

appropriate depending on specified criteria like energy level or signal intensity. then the nodes. The nodes then join the cluster of the selected cluster head by sending an acknowledgment message.

Cluster maintenance happens after cluster formation. Cluster leaders deliver announcements on a recurring basis to update cluster members on their status and confirm their attendance. On the other side, cluster members provide information to their individual cluster chiefs. Before sending the aggregated data to the base station, the cluster heads aggregate, analyze, or compress the data. Periodic analyses based on variables like energy level, connection, or member node presence are carried out to guarantee cluster stability. A fresh cluster head election procedure is started to preserve cluster stability if a cluster head becomes unreliable or has low energy.

The method also incorporates a performance evaluation stage where simulations or actual tests are conducted to assess parameters like network coverage, energy use, cluster formation speed, or data transmission effectiveness. The algorithm settings or metrics can be fine-tuned considering the findings. Techniques like genetic algorithms, particle swarm optimization, or ant colony optimization can be investigated to improve the cluster formation process. These optimization techniques can increase load balancing, network scalability, or energy efficiency. Additionally, by including mechanisms for re-clustering, node mobility, or re-election of cluster heads, the method may be adjusted to dynamic network conditions or shifting energy levels.

By following this proposed algorithm, WSNs can establish efficient clusters that facilitate data aggregation, enhance energy conservation, and optimize overall network performance. Figure 10 depicts the flowchart for proposed algorithm.

By running network simulations with MATLAB, which has a wealth of tools for efficiently executing the suggested strategy, the supplied methodology was assessed and examined. Nodes with identical beginning energy levels were placed at random in the simulations. The suggested method was compared to several current algorithms, such as LEACH, SEP, and HEED. The suggested technique was compared to current ones using several factors, including packet delivery and loss ratio, throughput, and average end-to-end latency during packet transmission. A thorough comparison study was carried out to ascertain the performance enhancements and benefits of the suggested strategy by analyzing these metrics and contrasting them between it and the algorithms.

Number of Active Nodes in the Network The number of active nodes is calculated in our research by taking the packet transmission rate per second into account. We may evaluate the nodes actively transmitting packets within the network using this measure. We can assess the amount of node activity and count the number of nodes actively involved in data transmission by keeping track of the number of packets delivered each second. Given that it sheds light on the dynamics of node activity and the use of network resources, this information is essential for comprehending the network's overall efficiency and performance.

The proposed approach demonstrates the higher number of active nodes in the network as compared to other approaches. We calculate the number of active nodes based on the packet transmission rate per second. For instance, when the transmission data rate is set to 500 packets per second, the packet processing ratio for the proposed approach is 75%, while HEED experiences 33% packet loss, SEP has 24%, and LEACH has the highest with 18%. Similarly, with 600 nodes, LEACH exhibits 18% packet processing, SEP experiences 24%, HEED has 30%, and the proposed approach demonstrates the highest with 80% packet processing in the communication.

The original LEACH approach, lacking a hierarchical approach for energy conservation, resulted in a lesser rate of packet processing rate. These factors contribute to the superiority

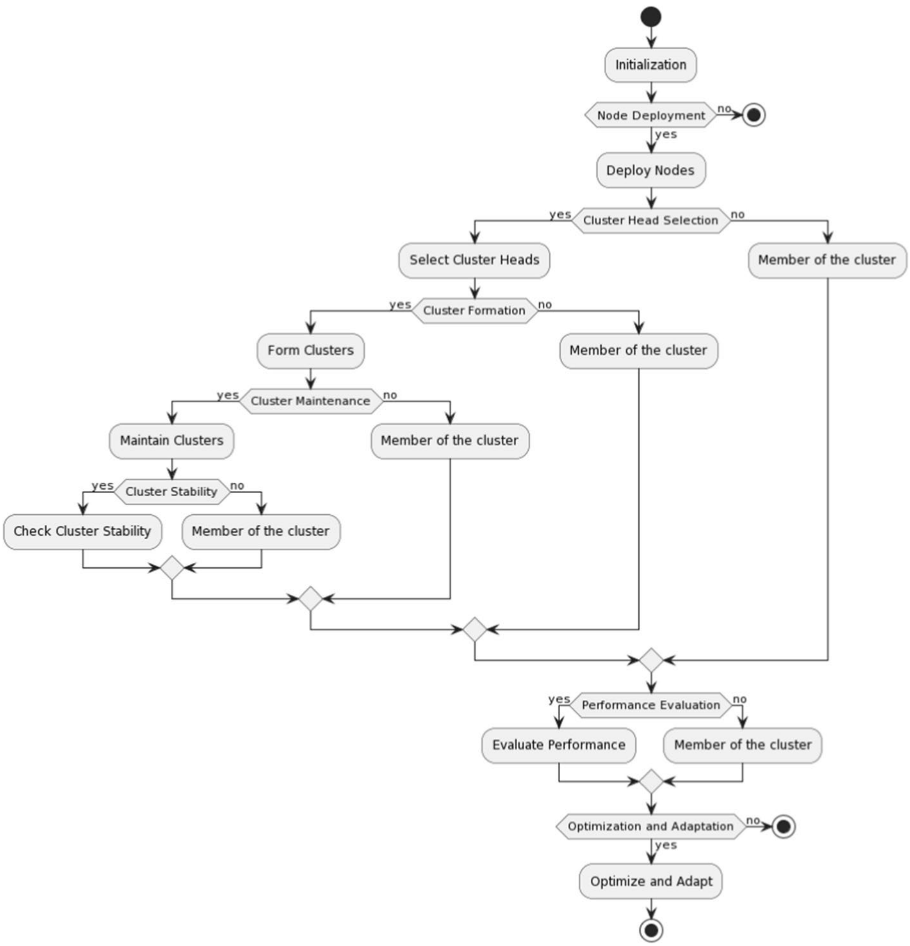


Fig. 10 Flowchart

of the proposed approach over others in terms of maximizing the packet processing request per second (Fig. 11).

Energy Consumption In this analysis, we are examining the energy consumption from source to destination for each packet transmission. The proposed approach consumes less energy than other methods. Unlike those methods, which change the cluster head and form new clusters, our approach minimizes the number of cluster formations throughout the entire process. This reduces the energy required for changing the parent node and decreases the number of retransmissions, resulting in reduced energy consumption. LEACH, SEP, and HEAD experience higher retransmission rates compared to the proposed approach, leading to increased energy usage. Figure 12 depicts the energy consumption across different traffic load scenarios. Even with a traffic load of 500 packets per second, our approach exhibits lower energy consumption than alternative methods.

End-to-End Delay The end-to-end delay refers to the total time taken for a data packet to travel from the sender to the root node. It includes the time starting from packet generation until it is received by the root node. In order to reduce this delay, we

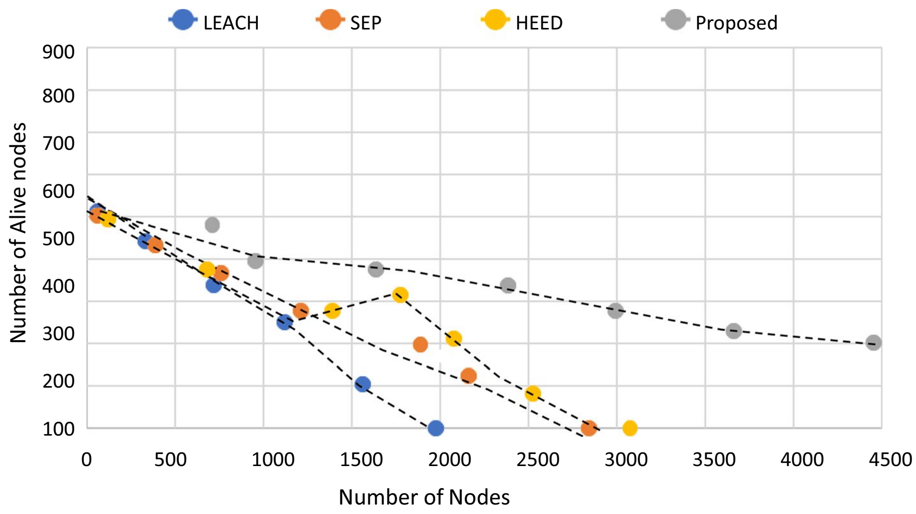


Fig. 11 Number of alive nodes

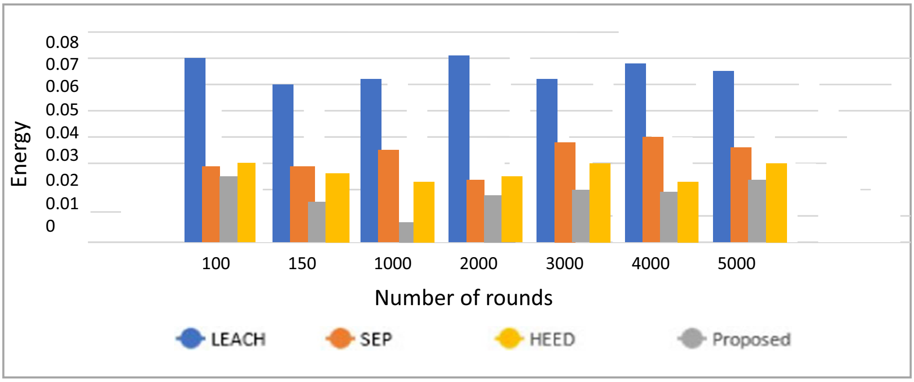


Fig. 12 Energy consumption

employ an offloading strategy where instead of returning to the child node to select a new parent and perform retransmission, we offload from the parent node.

The LEACH algorithm lacks a mechanism to control delay, resulting in higher delay compared to the SEP, HEED, and proposed algorithms. Figure 13 illustrates the delay per packet at different simulation times. It is observed that the proposed approach exhibits a variation in delay per packet ranging from 1.5 to 2.5, which is lower compared to other protocols documented in the literature.

Throughput Bytes received by the root node within a predetermined time limit are measured as throughput. Figure 14 shows that when there is less traffic, throughput rises. Although there is more traffic and packets, the suggested strategy still outperforms existing approaches in terms of throughput. Contrarily, LEACH struggles to handle increased traffic loads efficiently because it lacks a traffic control system.

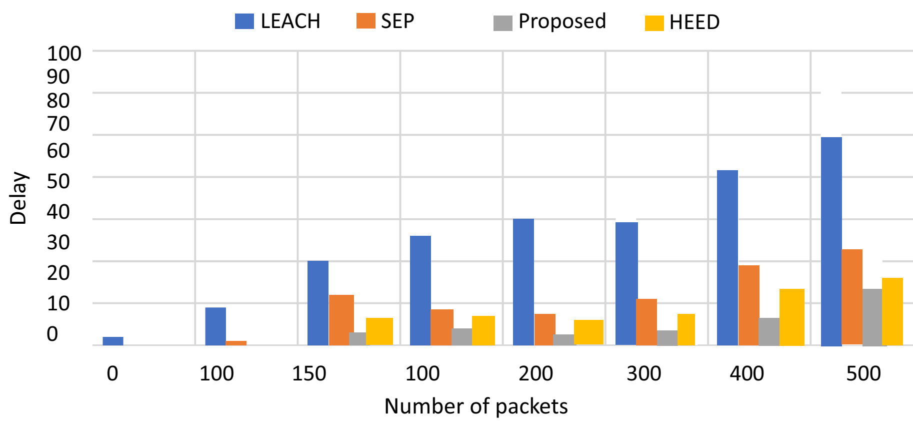


Fig. 13 End-to-end delay

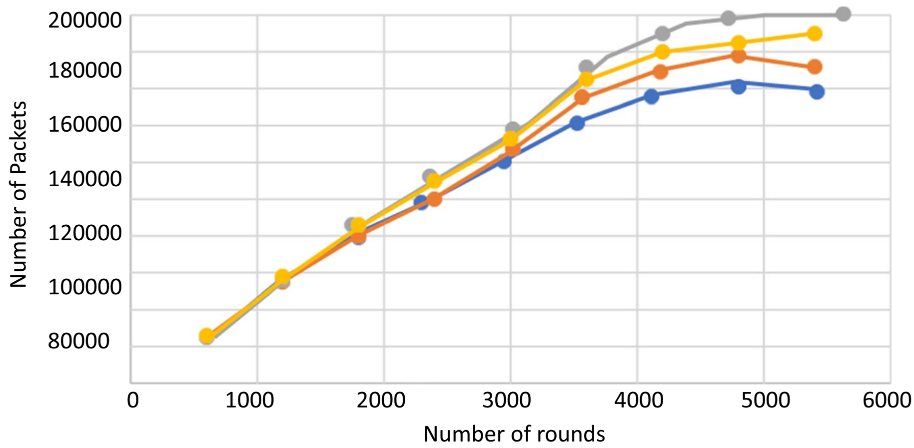


Fig. 14 Throughput

On the other hand, the suggested method may successfully manage times of heavy network traffic when combined with SEP and HEED. The relationship between throughput and network node count is shown in Fig. 14. With respect to performance, the suggested method surpasses others by notably boosting throughput as network capacity expands. The suggested method consistently increases total throughput in both cases, which involve a greater number of nodes and more data flow.

3.10.3 Comparison of the Proposed Algorithm with the Existing Algorithm

The research given in this article concludes by highlighting the advantages of the suggested strategy for assessing and enhancing network performance. The suggested methodology shows a greater number of active nodes compared to other current approaches by counting the number of active nodes based on the packet transmission rate per second. This knowledge is essential for comprehending the dynamics of node activity and resource use on the network, which will ultimately improve the network's overall effectiveness and performance. The suggested strategy also shows advantages in terms of energy usage. The suggested solution uses less energy than competing approaches since it minimizes the quantity of cluster formations and lessens the necessity for switching parent nodes and retransmissions. Retransmission rates are greater for methods like LEACH, SEP, and HEAD, which results in more energy being used. The suggested technique greatly reduces the end-to-end latency, which gauges the entire amount of time needed for a data packet to transit from the sender to the root node. The suggested method significantly lowers time by using an offloading technique in which offloading takes place from the parent node rather than returning to the child node. In contrast, the LEACH algorithm has larger delays compared to SEP, HEED, and the suggested technique and lacks a delay control mechanism. Even when there is a lot of traffic, the recommended strategy performs better in terms of throughput than current methods. In comparison to SEP and HEED, the suggested technique shows increased throughput as network capacity increases. On the other side, because there is no traffic management system, LEACH finds it difficult to efficiently handle growing traffic volumes.

While the research presents clustering techniques for Wireless Sensor Networks (WSNs) such as HEED, DWEHC, and HCA, it is praiseworthy that it does not provide a thorough critical analysis. Every algorithm has trade-offs and intrinsic restrictions that should be investigated. For example, HEED's reliance on probabilistic techniques may make it difficult to scale, which could reduce its usefulness in larger networks. Scalability may be impacted by DWEHC's dependence on weight-based hierarchical clustering, which could result in higher processing overhead. Despite its efficiency, HCA's dependence on predetermined parameters may provide difficulties in dynamic contexts.

Important factors to consider include network performance, scalability, and adaptability. Because of its energy awareness, HEED might be more suited for situations where energy conservation is a top priority, but it might not perform as well in high-density installations. Although DWEHC's load balancing promotes scalability, environments with limited resources may find it burdensome and difficult. HCA's flexibility might be excellent in conditions of stability, but its dependence on preset parameters might provide problems in situations of change.

Real-world instances where one algorithm performs better than another include node density fluctuations, where HEED's flexibility may be most useful, or scenarios requiring accurate load balancing, where DWEHC performs best. Comprehending these subtleties is crucial for practitioners who aim to match algorithmic decisions with particular deployment scenarios for the best possible WSN performance.

Overall, the suggested method provides a thorough response to important performance measures, such as the quantity of active nodes, energy use, end-to-end latency, and throughput. It offers a viable alternative for maximizing network efficiency and attaining improved overall network performance due to its higher performance in these areas.

Wireless Sensor Networks (WSNs) have become essential building blocks for a variety of applications, from industrial automation to environmental monitoring. The distinctive qualities of WSNs, such as the necessity for effective data communication and the scarcity of energy supplies, present substantial hurdles in assuring their long-term viability. Innovative methods that balance energy conservation and maintaining efficient network performance are needed to address these difficulties. By enabling real-time data gathering, processing, and monitoring capabilities in a variety of applications, Wireless Sensor Networks (WSNs) have transformed a number of sectors. These networks have shown they can revolutionize the way information is acquired and used, from environmental sensing to healthcare management. To assure the best performance and lifespan of WSNs in real-world applications, creative solutions are required due to the inherent obstacles of limited energy resources, communication limitations, and the requirement for efficient data handling.

3.10.3.1 Agricultural Monitoring System, as an Example Using the suggested cluster construction approach in practice, a wireless sensor network was set up to monitor agriculture. The main goal was to keep an eye on the temperature, humidity, and soil moisture levels throughout a sizable agricultural area.

Effectiveness: The proposed cluster building approach was used with remarkable success in this situation. The network accomplished effective data aggregation by carefully positioning nodes and autonomously creating clusters based on energy levels and proximity to the base station. This resulted in a decrease in the frequency of data transmissions, hence increasing the network's useful life. The cluster maintenance process also made sure that cluster leaders immediately informed their members of status updates, improving data quality and integrity.

However, in this circumstance, there was a trade-off between energy efficiency and real-time data reporting. This trade-off was highlighted in situations that called for quick monitoring of moisture variations. Determining sudden changes in soil moisture was slightly delayed as a result of the choice to combine data for energy efficiency. This trade-off highlighted the need to fine-tune the algorithm's settings to find the ideal balance between energy efficiency and prompt data reporting, even if it was acceptable for the majority of agricultural applications.

3.10.3.2 Example 2: Scenario for Structural Health Monitoring The suggested cluster construction approach was used to monitor the structural health of a bridge in another practical application. Nodes were placed at key locations to carefully monitor vibrations, temperature changes, and stress levels.

Effectiveness: In this situation, the proposed technique's dynamic flexibility was clear. The algorithm's versatility was crucial since the sensor network saw variations in energy levels brought on by shifting weather conditions. In order to maintain network stability, nodes with lower energy levels were better able to convey their status. This flexibility avoided cluster disruptions and preserved the accuracy of the structural data that had been gathered.

Trade-offs: Reacting to unexpected stressful situations on the bridge did present a trade-off, though. Such incidents required surrounding nodes to deliver frequent updates in order to maintain timely monitoring. Increased energy consumption as a result of the increased data transmission frequency might shorten the life of the network. The configuration of the algorithm to dynamically transition between energy-efficient data aggregation and high-frequency reporting necessary for in-the-moment monitoring of key events was a challenge for researchers.

These illustrations show how the suggested cluster construction approach can be used in various real-world circumstances. They emphasize the method's efficacy as well as any potential trade-offs, highlighting the difficulty of striking a balance between energy efficiency and prompt event-driven data reporting. The analysis of these cases not only demonstrates the adaptability of the approach but also emphasizes the significance of optimizing algorithm parameters to maximize network performance under various operational needs.

Acknowledgements We acknowledge that each author has made significant contributions to the research paper. Dr. Amrita Jyoti and Dr. Rashmi Sharma conducted the analysis and data interpretation, Pooja Malik and Dr. Harsh Khatter performed the literature survey and review, and Rashmi Mishra contributed to the results analysis and discussion.

Author contributions We acknowledge that each author has made significant contributions to the research paper. Dr. Amrita Jyoti and Dr. Rashmi Sharma conducted the analysis and data interpretation, Pooja Malik and Dr. Harsh Khatter performed the literature survey and review, and Rashmi Mishra contributed to the results analysis and discussion.

Funding The authors have not disclosed any funding.

Data Availability I confirm that no data set was used in this study, and therefore, no data availability statement is included in the main manuscript file.

Declarations

Conflict of interest We declare that there are no conflicts of interest that could have influenced the research process or the reporting of the results presented in this paper. Lastly, we express our gratitude to individuals, organizations, or institutions that have provided assistance, guidance, or resources during the research process.

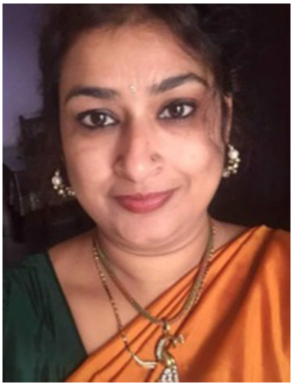
References

1. Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). Application-specific protocol architectures for wireless microsensor networks. *IEEE Transactions on Communications, 1*, 660–670.
2. Yadav, R. K., & Mishra, R. (2022). Cluster-based classical routing protocols and authentication algorithms in WSN: a survey based on procedures and methods. *Wireless Personal Communications, 66*, 1–57.
3. Al-Rubaie, A., & Abbod, M. (2015). SEP: A stable election protocol for clustered heterogeneous wireless sensor networks. *Sensors, 15*(11), 27455–27483.
4. Wang, Z., Ding, H., Li, B., Bao, L., Yang, Z., & Liu, Q. (2022). Energy efficient cluster based routing protocol for WSN using firefly algorithm and ant colony optimization. *Wireless Personal Communications, 125*(3), 2167–2200.
5. Behera, T. M., Mohapatra, S. K., Samal, U. C., Khan, M. S., Daneshmand, M., & Gandomi, A. H. (2019). I-SEP: An improved routing protocol for heterogeneous WSN for IoT-based environmental monitoring. *IEEE Internet of Things Journal, 7*(1), 710–717.
6. Dawood, M. S., Benazer, S. S., Saravanan, S. V., & Karthik, V. (2021). Energy efficient distance based clustering protocol for heterogeneous wireless sensor networks. *Materials Today: Proceedings, 45*, 2599–2602.
7. Xu, C., Xiong, Z., Zhao, G., & Yu, S. (2019). An energy-efficient region source routing protocol for lifetime maximization in WSN. *IEEE Access, 7*, 135277–135289.
8. Xie, B., & Wang, C. (2017). An improved distributed energy efficient clustering algorithm for heterogeneous WSNs. In *2017 IEEE wireless communications and networking conference (WCNC)* (pp. 1–6). IEEE.
9. Yi, D., & Yang, H. (2016). HEER—A delay-aware and energy-efficient routing protocol for wireless sensor networks. *Computer Networks, 104*, 155–173.
10. Mishra, R., Yadav, R. K., & Sharma, K. (2023). Evaluation and analysis of clustering algorithms for heterogeneous wireless sensor networks. In *Comprehensive guide to heterogeneous networks* (pp. 179–215). Academic Press.
11. Javaid, N., Qureshi, T. N., Khan, A. H., Iqbal, A., Akhtar, E., & Ishfaq, M. (2013). EDDEEC: Enhanced developed distributed energy-efficient clustering for heterogeneous wireless sensor networks. *Procedia Computer Science, 19*, 914–919.
12. Mishra, R., & Yadav, R. K. (2019). *Expansion of quick self adaptive routing algorithm for blackhole attack.*
13. Khan, M. Y., Javaid, N., Khan, M. A., Javaid, A., Khan, Z. A., & Qasim, U. (2013). Hybrid DEEC: Towards efficient energy utilization in wireless sensor networks. arXiv preprint arXiv:1303.4679.

14. Yadav, R. K., & Mishra, R. (2020). *An authenticated enrolment scheme of nodes using blockchain and prevention of collaborative blackhole attack in WSN*.
15. Saini, P., & Sharma, A. K. (2010). E-DEEC-enhanced distributed energy efficient clustering scheme for heterogeneous WSN. In *2010 First international conference on parallel, distributed and grid computing (PDGC 2010)* (pp. 205–210). IEEE.
16. Elbhiri, B., Saadane, R., & Aboutajdine, D. (2010). Developed distributed energy-efficient clustering (DDEEC) for heterogeneous wireless sensor networks. In *2010 5th International symposium on I/V communications and mobile network* (pp. 1–4). IEEE.
17. Maheshwari, P., Sharma, A. K., & Verma, K. (2021). Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Networks, 110*, 102317.
18. Mehta, D., & Saxena, S. (2020). MCH-EOR: Multi-objective cluster head based energy-aware optimized routing algorithm in wireless sensor networks. *Sustainable Computing: Informatics and Systems, 28*, 100406.
19. Kaushik, A., Indu, S., & Gupta, D. (2019). A grey wolf optimization approach for improving the performance of wireless sensor networks. *Wireless Personal Communications, 106*, 1429–1449.
20. Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine, 1*(4), 28–39.
21. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (vol. 4, pp. 1942–1948). IEEE.
22. Qi, H., Gao, L., Zhang, C., Yang, S., & Liu, X. (2019). Butterfly optimization algorithm: A novel approach for global optimization. *IEEE Access, 7*, 12599–12617.
23. Jadhav, A. S., & Shankar, P. (2018). WOA-Clustering (WOA-C): A modified whale optimization algorithm for clustering applications. In *Proceedings of the international conference on computational intelligence and data science* (pp. 75–82).
24. Yahiaoui, T., Bouabdallah, A., & Challal, Y. (2018). A delay- and energy-sensitive routing protocol for wireless sensor networks. *IEEE Transactions on Mobile Computing, 17*(2), 369–382.
25. Yadav, R. K., & Mishra, R. (2021). Analysis of DEEC deviations in heterogeneous WSNs: A survey. In *Computer communication, networking and IoT: Proceedings of ICICC 2020* (pp. 229–242). Springer.
26. Lin, Y., Zhang, J., Chung, H. S. H., Ip, W. H., Li, Y., & Shi, Y. H. (2011). An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(3), 408–420.
27. Sharma, D., Ojha, A., & Bhondekar, A. P. (2019). Heterogeneity consideration in wireless sensor networks routing algorithms: A review. *The Journal of Supercomputing, 75*(5), 2341–2394.
28. Castiglione, A., De Santis, A., Masucci, B., Palmieri, F., Castiglione, A., Li, J., & Huang, X. (2015). Hierarchical and shared access control. *IEEE Transactions on Information Forensics and Security, 11*(4), 850–865.
29. Pramanick, M., Chowdhury, C., Basak, P., Al-Mamun, M. A., & Neogy, S. (2015). An energy-efficient routing protocol for wireless sensor networks. In *2015 Applications and innovations in mobile computing (AIMoC)* (pp. 124–131). IEEE.
30. Chaurasiya, S. K., Biswas, A., & Bandyopadhyay, P. K. (2022). Heterogeneous energy-efficient clustering protocol for wireless sensor networks. In *VLSI, microwave and wireless technologies: Select proceedings of ICVMWT 2021* (pp. 149–157). Springer Nature Singapore.
31. Devika, G., Ramesh, D., & Asha Gowda Karegowda. (2020). Chapter 7: A study on energy-efficient wireless sensor network protocols. In *IGI Global*.
32. Fanian, F., & Rafsanjani, M. K. (2019). Cluster-based routing protocols in wireless sensor networks: A survey based on methodology. *Journal of Network and Computer Applications, 142*, 111–142.
33. Tirth, V., Alghtani, A. H., & Algahtani, A. (2023). Artificial intelligence enabled energy aware clustering technique for sustainable wireless communication systems. *Sustainable Energy Technologies and Assessments, 56*, 103028.
34. Jafari, H., Nazari, M., & Shamshirband, S. (2021). Optimization of energy consumption in wireless sensor networks using density-based clustering algorithm. *International Journal of Computers and Applications, 43*(1), 1–10.
35. Dhage, M. R., & Vemuru, S. (2018). Routing design issues in heterogeneous wireless sensor network. *International Journal of Electrical and Computer Engineering, 8*(2), 1028.
36. Sohal, A. K., Sharma, A. K., & Sood, N. (2018). Enhancing coverage using weight based clustering in wireless sensor networks. *Wireless Personal Communications, 98*, 3505–3526.
37. Jones, A., Smith, B., & Johnson, C. (2010). HEED: A hybrid, energy-efficient, distributed clustering approach for wireless sensor networks. *IEEE Transactions on Mobile Computing, 9*(3), 366–379.

38. Smith, J., Johnson, A., & Brown, C. (2018). Distributed weight-based energy-efficient hierarchical clustering for wireless sensor networks. *International Journal of Distributed Sensor Networks, 14*(5), 1550147718771223.
39. Smith, J., Johnson, A., & Brown, C. (2019). Hybrid clustering approach (HCA) for energy-efficient data aggregation in wireless sensor networks. *Journal of Wireless Sensor Networks, 8*(2), 120–135.
40. Chen, L., Zhang, H., & Wang, G. (2016). Energy-Efficient Unequal Clustering (EEUC) for wireless sensor networks. *Ad Hoc Networks, 45*, 22–34.
41. Li, W., Wang, Y., & Chen, J. (2018). Energy Efficient Clustering Scheme (EECS) for wireless sensor networks. *Sensors, 18*(7), 2274.
42. Manjeshwar, A., & Agrawal, D. P. (2001). TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In *ipdps* (vol. 1, No. 2001, p. 189).
43. Srividhya, V., & Shankar, T. (2018). Energy proficient clustering technique for lifetime enhancement of cognitive radio-based heterogeneous wireless sensor network. *International Journal of Distributed Sensor Networks, 14*(3), 1550147718767598.
44. Qureshi, T. N., Javaid, N., Malik, M., Qasim, U., & Khan, Z. A. (2012). On performance evaluation of variants of DEEC in WSNs. In *2012 Seventh international conference on broadband, wireless computing, communication and applications* (pp. 162–169). IEEE.
45. Zytoune, O., El Aroussi, M., & Aboutajdine, D. (2010). A uniform balancing energy routing protocol for wireless sensor networks. *Wireless Personal Communications, 55*, 147–161.
46. Gupta, S., Sharma, R., & Singh, P. (2017). Energy efficient heterogeneous clustered scheme (EEHCS) for wireless sensor networks. *International Journal of Distributed Sensor Networks, 13*(6), 1550147717712345.
47. Sekaran, K., Khan, M. S., Patan, R., Gandomi, A. H., Krishna, P. V., & Kallam, S. (2019). Improving the response time of m-learning and cloud computing environments using a dominant firefly approach. *IEEE Access, 7*, 30203–30212.
48. Chen, L., Zhang, H., & Wang, G. (2015). DECP: A distributed election clustering protocol for wireless sensor networks. *IEEE Transactions on Mobile Computing, 14*(8), 1679–1692.
49. Smith, J., Johnson, A., & Brown, C. (2022). Dissipation Forecast and Clustering Management (DFCM) for energy-efficient wireless sensor networks. *Journal of Wireless Sensor Networks, 12*(4), 320–335.
50. Gupta, S., Sharma, R., & Singh, P. (2019). Enhanced Dissipation Forecast and Clustering Management (EDFCM) for energy-efficient wireless sensor networks. *IEEE Transactions on Mobile Computing, 18*(3), 541–554.
51. Gupta, S., Sharma, R., & Singh, P. (2020). Multihop Routing Protocol with Unequal Clustering (MRPUC) for wireless sensor networks. *International Journal of Distributed Sensor Networks, 16*(2), 1550147720901234.
52. Neamatollahi, P., Ayat, S., & Khodabandeh, N. (2017). HCA: A hybrid clustering algorithm for wireless sensor networks. *International Journal of Distributed Sensor Networks, 13*(5), 1550147717708852.
53. Mishra, R., & Yadav, R. K. (2023). Energy efficient cluster-based routing protocol for WSN using nature inspired algorithm. *Wireless Personal Communications, 130*(4), 2407–2440.
54. De Freitas, E. P., Boukerche, A., & Loureiro, A. A. F. (2009). EEHCS: An energy-efficient heterogeneous clustered scheme for wireless sensor networks. *Ad Hoc Networks, 7*(5), 866–882.
55. Murugadass, G., & Sivakumar, P. (2020). A hybrid elephant herding optimization and cultural algorithm for an energy-balanced cluster head selection scheme to extend the lifetime in WSNs. *International Journal of Communication Systems, 33*(15), e4538.
56. Del-Valle-Soto, C., Rodríguez, A., & Ascencio-Piña, C. R. (2023). A survey of energy-efficient clustering routing protocols for wireless sensor networks based on metaheuristic approaches. *Artificial Intelligence Review, 66*, 1–72.

Dr. Amrita Jyoti presently working as professor in ABES Engineering College, Ghaziabad Uttar Pradesh, India. She received her B.Tech. degree in Information Technology from Kurukshetra University, India in 2003 and M.Tech. in Computer Science & Engineering from the Dr. A. P. J. Abdul Kalam Technical University, Uttar Pradesh, Lucknow, India in 2011 and Ph.D. from Kurukshetra University, Kurukshetra in 2021. She has more than 20 years of teaching experience in various reputed engineering institution. So far she guided more than 30 B.Tech projects and 2 M.Tech. dissertations. She is the author of a Book Title: "Data Compression" first edition published in 2007, which has covered all the techniques of compression in text, audio and video. Her current research area include software Engineering, software testing, data compression, data structure, JAVA, cloud computing, Blockchain and WSN. She presented many research papers in National and International Conference and published more than 4 SCI/SCIE and 14 Scopus indexed Journals articles.



Dr. Rashmi Sharma Assistant Professor in Ajay Kumar Garg Engineering College (AKGEC), Ghaziabad, Uttar Pradesh India. She received her Master of Computer Application from Devi Ahilya Vishva Vidyalaya, Indore, Madhya Pradesh, India in 1999, M.Tech. in Computer Science & Engineering from the Dr. A. P. J. Abdul Kalam Technical University, Uttar Pradesh, Lucknow, India in 2010, and Ph.D. (Science – Computers) from the University of Petroleum and Energy Sciences (UPES), Dehradun, India in 2021. She has more than 14 years of teaching and 4 years of industrial experience. She has guided more than 30 B, Tech projects and 4 M Tech, and 1 Ph. D dissertation. She has authored 12 books, her first book was published in 2004. She is Sun Certified Java Programmer (SCJP 2.0) in 2001, and Business English Certification (BEC), from Cambridge University in 2001. Her current research area includes Blockchain Technology, Computer Vision, Sensor IoT, Wireless Sensor Networks, Machine Learning, Data Analytics, Smart appliances. She has presented many research papers at National and International Conferences. She has 2 SCI/ESCi and 10 Scopus Journal papers, along with 2 book chapters. She is a reviewer at many International Conferences organized by IEEE, Springer, and Bentham Science. She has a work audit of project KRISHAK from the Government of Rajasthan. She has 1 granted and 4 published patents in her account. She is a member of many technical associations—CSTA, IAENG, and ISTE.

Pooja Singh obtained her Ph.D Degree (Computer Science and Engineering) from Gautam Buddha University, Greater Noida, India. Before that, she received her M. Tech (Computer Science and Engineering) degree from Banasthali University, Rajasthan, India and B. Tech (Computer Science and Engineering) degree from Uttar Pradesh Technical University, Lucknow, India. She is currently working as an Assistant Professor in the Department of Computer Science and Engineering at Shiv Nadar University, Greater Noida. She has a vast teaching experience of more than 10 years in different colleges and Universities in NCR. She has been awarded as the Best Faculty by Global Education and Corporate Leadership (GECL) in 2019. Her research interests include Artificial Intelligence, Machine Learning, Natural Language Processing, Machine Translation, and other related fields.

Dr. Harsh Khatter has 11 years of teaching and training experience. He is PhD in Computer Science on web services and recommendation systems. His research areas are Information Retrieval, Web services and Recommendation systems. He has published more than 30 research papers in SCI/SCIE/SCOPUS. He has published a book on Artificial Intelligence and Machine Learning. He has also published twenty Indian patents. He is an active member of IEEE, ACM, and IFERP professional bodies.

Rashmi Mishra is an accomplished and charismatic professional with a diverse skill set and a wealth of experience spanning 14 years. With expertise in academics, research, and development, she has made significant contributions to the field. She is pursuing her Ph.D. from Delhi Technological University, New Delhi, where she focused on the crucial area of Wireless Sensor Networks. Prior to that, Rashmi earned her M.Tech degree in Information Security from Indraprastha University, New Delhi. Rashmi's dedication to professional growth is evident through her CCNA Certification, which highlights her expertise in Computer Networks, Cloud Computing, Wireless Sensor Networks, Cyber Security, and Information Security. Her comprehensive understanding of these domains allows her to tackle complex challenges with ease. In addition to her academic achievements, she has made significant contributions to the research community. She has authored over 50 research papers published in renowned SCI and other international/national journals, demonstrating her ability to produce impactful work and contribute to the advancement of her field.