Journal of Universal Computer Science, vol. 31, no. 10 (2025), 1042-1079
submitted: 1/8/2024, accepted: 25/2/2025, appeared: 28/8/2025

Residual Energy-Aware Fuzzy-Based Clustering Algorithm for Underwater Wireless Sensor Networks

Sorav Kumar Singh
(Department of Information Technology, Tripura University, Agartala, India,
sksoravsk@gmail.com)

Alak Roy*
(Department of Information Technology, Tripura University, Agartala, India
https://orcid.org/0000-0001-6069-9157, alakroy@tripurauniv.ac.in)

Rajneesh Raushan
(Department of Information Technology, Tripura University, Agartala, India,
rajneeshraushan536@gmail.com)

Abstract: In the field of underwater exploration and research, Underwater Wireless Sensor Networks (UWSNs) play a vital role in understanding the marine environment, oceanography, and marine biology. A key strategy used in UWSNs to aggregate sensor nodes and improve network performance while extending battery life through lower energy usage is clustering. However, available clustering algorithms do not specifically address all the underwater problems, viz., communication is constrained by the limited bandwidth and high latency of acoustic signals, while energy consumption is critical due to the difficulty of recharging or replacing underwater batteries. The harsh underwater environment, with varying pressure, salinity, and movement, affects sensor performance and durability. Accurate localization is difficult without GPS and relies on less precise acoustic methods. So, this paper proposes a Residual Energy-Aware Fuzzy-Based Clustering Algorithm (REAFCA) for UWSNs which presents a novel framework intended to improve network performance and address issues with energy usage. For effective data routing, the REAFCA dynamically arranges clusters based on important factors such as node rank, radius, threshold, angular velocity, and residual energy. To maximize leadership inside the clusters, the adaptive threshold method makes sure that only superior cluster heads are chosen. The algorithm also incorporates dynamic range changes for communication to adapt to changing network circumstances. This algorithm mainly focuses on clustering in an underwater environment while improving the energy efficiency and network life of the nodes. Simulation results demonstrate the superiority of the proposed algorithm over K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER algorithms in terms of energy efficiency and throughput while achieving comparable average delay.

Keywords: Underwater Wireless Sensor Networks, Cluster Head Selection, Residual Energy-Aware Fuzzy-Based Clustering Algorithm, Energy Efficiency, Throughput
Categories: C.2
DOI: 10.3897/jucs.132502

1 Introduction

In recent years, technological developments that allows for previously unheard levels of exploration and scientific discovery have contributed to the fascination with the underwater world. Autonomous Underwater Vehicles (AUVs) and Remotely Operated

Vehicles (ROVs) have developed into indispensable instruments that enable researchers to investigate previously unexplored habitats and descend to extremely low depths. These highly developed devices have been crucial in the discovery of novel marine animals, the documentation of historic shipwrecks, and the mapping of complex underwater ecosystems. In addition to offering a glimpse into Earth's geological past, the discovery of underwater caves and trenches has presented new opportunities and challenges for scientists [Heidemann, 2012, Xie, 2024]. Scientists are solving the secrets of underground environments and learning more about the dynamic processes sculpting our globe as they explore these unexplored regions. Additionally, the study of underwater archaeology has grown rapidly, with digs uncovering submerged relics from marine and ancient civilizations [Roy, 2011]. These findings advance our knowledge of the history of humanity and the interactions between societies and the waters over time. Simultaneously, there has been an increase in the exploration of submerged resources, including mineral and energy deposits. There are unexplored resources on the ocean floor that might be able to supply the expanding global need [Heidemann, 2012, Zeng, 2014]. UWSNs have a broad spectrum of applications that are pivotal for various fields. In environmental monitoring, UWSNs are used to track marine life, monitor water quality, and observe changes in underwater ecosystems. These applications help in understanding the impacts of climate change and human activities on marine environments. In offshore exploration, UWSNs facilitate the detection of underwater oil and gas resources, supporting energy industries [Roy, 2010]. Disaster prevention applications include tsunami and earthquake monitoring, where timely data from UWSNs can provide early warnings and help mitigate the effects of natural disasters. Additionally, UWSNs are utilized in military surveillance for detecting submarines and underwater mines, enhancing national security [He, 2024]. Whereas the unique properties of UWSNs set them apart from terrestrial wireless sensor networks. One significant property is the limited energy resources available to underwater sensor nodes. These nodes are often battery-powered and challenging to recharge or replace, making energy efficiency a critical concern. The underwater environment also affects signal propagation, with water currents, temperature variations, and salinity impacting communication reliability. These factors necessitate the use of robust, waterproof, and durable hardware capable of withstanding the harsh underwater conditions for extended periods. Additionally the communication in UWSNs primarily relies on acoustic signals, as radio waves suffer from rapid attenuation and limited range in water. Acoustic communication, however, comes with its own set of challenges. It typically offers lower bandwidth, resulting in slower data transmission rates. The underwater medium also introduces higher latency due to the slower speed of sound compared to electromagnetic waves. Furthermore, acoustic signals are more prone to errors caused by multipath propagation and ambient noise, which can degrade communication quality. These challenges require the development of specialized communication protocols that can handle high error rates and ensure efficient and reliable data transmission. To address these communication challenges, effective clustering and energy management strategies are essential [Heidemann, 2012, Zeng, 2014]. Clustering algorithms in UWSNs must consider the energy constraints and communication difficulties unique to the underwater environment. Optimal cluster head selection is crucial, as the cluster head is responsible for aggregating and relaying data to the base station. An efficient cluster head should minimize energy consumption, manage data traffic effectively, and maintain robust communication links. Balancing the energy load across all sensor nodes is also vital to prevent premature node failures and extend the network's overall lifetime. Existing clustering algorithms for UWSNs include techniques such as LEACH, TEEN, HEED, PEGASIS, DBSCAN and HEER [Heinzelman, 2000, Younis, 2004, Yi, 2016, Lindsey, 2002, Ester, 1996]. While these algorithms offer various strategies for improving energy efficiency and communication reliability, they often fall short in addressing the specific challenges posed by the underwater environment. For instance, LEACH's frequent cluster head changes can lead to excessive energy consumption, TEEN's threshold-based approach might miss critical data, and HEED's reliance on residual energy does not fully account for the unique propagation delays and error rates in UWSNs. Hierarchical structure of the HEER algorithm increases communication overhead and does not allow for scalability in dense networks and energy efficiency. PEGASIS suffers from heavy delay and inefficiency in dynamic networks due to its communication based on the sequential chain. DBSCAN is sensitive to parameter selection and fails to identify clusters in networks with varying node densities or sparse deployments. There are a few more algorithms as discussed in the Section 3 with research gap justification in the Section 3.3. These drawbacks highlight the need for new clustering algorithms tailored specifically for UWSNs. This paper aims to introduce a novel clustering algorithm that addresses these limitations by considering the unique properties of underwater communication, optimizing energy consumption, and enhancing network lifetime through more effective cluster head selection and load balancing techniques.

1.1 Challenges and Issues

Clustering in UWSNs presents a host of challenges and issues due to the harsh and unique underwater environment. One of the primary challenges is the limited battery life of sensor nodes. These nodes are often deployed in remote and hard-to-access locations, making it impractical to replace or recharge their batteries frequently. This limitation necessitates the development of energy-efficient algorithms to maximize the nodes operational lifespan. Whereas communication problems further complicate clustering in UWSNs. Unlike terrestrial networks, underwater communication primarily relies on acoustic signals, which are inherently slower and have lower bandwidth than radio waves. Acoustic signals are also highly susceptible to environmental factors such as water temperature, salinity, and pressure, leading to high latency and error rates. These communication issues can result in data loss and reduced reliability of the network. The main challenge comes with selecting of an optimal cluster head. The cluster head is responsible for aggregating data from other nodes and transmitting it to the base station. Therefore, it needs to be selected based on criteria that ensure minimal energy consumption, robust processing capabilities, and efficient communication. An inefficient cluster head selection can lead to rapid depletion of the cluster head's battery, causing network fragmentation and reduced network lifetime. Lastly, the overall network lifetime is a critical concern in UWSNs. It is essential to design clustering protocols that not only optimize energy consumption but also ensure balanced load distribution among the sensor nodes. Uneven energy consumption can lead to early node failures, which in turn can create coverage gaps and reduce the network's effectiveness. Thus, the goal is to develop clustering techniques that prolong network lifetime by balancing energy loads, minimizing communication overhead, and selecting optimal cluster heads.

1.2 Motivation of the Research

Clustering in UWSNs is motivated by the need to overcome the challenges and optimize the performance. UWSNs operate in harsh conditions with limited resources, including energy, computational power, and communication bandwidth. Additionally, underwater communication faces specific challenges such as limited range and increased signal attenuation. Clustering addresses these challenges by enabling localized communication within each cluster, reducing overall communication distance, and enhancing the reliability of data transmission. Moreover, clustering facilitates data aggregation and fusion, which are essential in UWSNs due to the redundant and correlated nature of the collected data. Although many algorithms were available in recent literature survey, however none of them is perfect and even now we can make some changes in the algorithms and to make it better and energy efficient.

1.3 Research Objectives

The objective of this research is to design and develop a clustering algorithm for Underwater Wireless Sensor Networks (UWSNs) that can extend the network lifetime by balancing the energy consumption of the sensor nodes. Accordingly, the broad objective is sub divided in to the following sub-objectives given as follows:

- Cluster Formation and cluster hehad selection: Enhance the cluster formation by creating stable clusters which can adapt to underwater conditions and support seamless communication and then cluster head selection by taking into various factors like rank, radius and angular velocity to identify the best cluster heads.
- Adaptive Communication Strategy: Enhance the network's ability to adjust dynamically to changing environmental conditions by introducing Signal Node functions that allow for changes in communication techniques.
- Overall Network Optimization: For enhancing the overall network optimization , the aim is to enhance the reliabilty, efficiency of the network in harsh underwtaer environments and scenarios.

1.4 Research Contributions

The paper proposes a Residual Energy-Aware Fuzzy-Based Clustering Algorithms (REAFCA) for UWSNs which makes a substantial contribution in this field. The proposed algorithm takes into various factors such as node radius, threshold rank, angular velocity, and residual energy to adjust to the conditions of UWSNs. Signal node integration improves network coordination and communication, which raises overall efficiency. A more precise assessment of fitness is possible with a weighted fitness computation that takes into account variables like rank, residual energy and radius. Fitness and energy criteria are used to further narrow the selection in a two-step procedure including likely and candidate cluster heads. The adaptive threshold computation also adapts to changing network properties. Specifically, the algorithm creates channels of communication that allow for the effective distribution of information by broadcasting messages and advertisements from potential cluster heads. Remarkably, the algorithm's scalability allows for various network sizes and settings, providing a theoretical framework for additional investigation and verification via simulations and actual trials. Moreover, AquaSim is used in this paper to provide simulation results which show the performance of the proposed algorithm with K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER, where average throughput, energy-efficiency and average delay is compared. Simulation results demonstrates the superiority of the REAFCA over the other algorithms in terms of energy efficiency and throughput, while achieving comparable average delay.

The paper is organized as follows. Section 1 represents the introduction, motivation, objective and contribution on clustering algorithm for UWSNs in this paper. In Section 2 background details, UWSNs and clustering has been described. A comparative study on existing clustering protocol is discussed in Section 3. The section 4 describes the proposed algorithm for clustering in UWSNs and briefly explains its features. The results of experiments for the clustering algorithm is shown in section 5. And finally conclude the paper with future work in Section 6.

2 Background Details

In the context of underwater there lies a realm very vast and enigmatic, a world with so many mysteries which are waiting to be unveiled. Within this domain, Countless Wonders await to be discovered and only a little fraction of its have been discovered yet [Gkikopouli, 2012].

There exist fragment of knowledge, scattered like pearls in the ocean. While some have captivated people's attention with tales of vanished civilizations and old ruins, others are still hidden beneath the surface, their secrets unsolved. The undersea realm is a patchwork of breathtaking landscapes and species both familiar and strange, from the towering coral reefs that house dynamic ecosystems to the gloomy abyssal plains that conceal creatures beyond comprehension [Luo, 2018]. In this a concept of underwater wireless sensor network is discovered which comes in handy for researcher to explore the vast community of the underwater habitants.

2.1 Underwater Wireless Sensor Networks

UWSNs are specialized networks made for data sharing and communication in submerged settings. These networks are made up of networked sensor nodes containing sensors that track underwater environmental variables like salinity, temperature, and pressure. UWSNs use acoustic waves to transmit data in order to overcome the particular difficulties associated with underwater communication [Zeng, 2014]. Despite being efficient in water, these signals encounter challenges such as varying propagation delays, multi path propagation, and signal attenuation. UWSN communication protocols are created expressly to address these issues and provide dependable data flow between sensor nodes. Sensor nodes perform three main tasks, monitor data, process data and receive data. These nodes are of different types of sensors and monitor different physical or chemical parameters in underwater environments and send the collected data to the central location. One of the main concepts of UWSNs is clustering in which sensor nodes make group and save energy by sharing data smartly to sink node [Roy, 2011, Roy, 2010].

2.1.1 UWSNs Characteristics

UWSNs has many distinct characteristics. Some of which are very important for data transmission in underwater. Firstly, UWSNs rely on acoustic waves for communication due to the limited propagation of radio waves and electromagnetic waves in underwater. Secondly, the underwater environment has many challenges like signal attenuation, delay and noise. All these challenges are solved by the help of UWSNs communication with the help of MAC, Clustering, Routing etc. Also the communication range in UWSNs is shorter than terrestrial networks because of underwater nature. Energy efficiency is crucial in UWSNs as sensor nodes typically operate on limited battery power. Overall, all the characteristics are important to reliably transfer data and save energy of nodes because of limited energy.

2.2 Clustering in UWSNs

Clustering techniques must take into account the peculiarities of sound propagation and the 3D deployment environment. Clustering decisions may be influenced by groupings based on proximity, depth zones, and acoustic communication ranges between nodes. Inter-cluster routing, data aggregation/fusion, and coordination with the surface station or sink are frequently handled by cluster heads. To increase network lifetime, residual energy levels must be taken into consideration by cluster head selection algorithms. It is difficult to maintain cluster stability over time because of the changing undersea conditions that impact connectivity. The quality of the link between sensors and their cluster head can be changed by node mobility or currents. To reorganize clusters and switch up cluster head responsibilities in response to topology changes, adaptive clustering protocols are required. Network scalability and energy consumption are also impacted by choosing the ideal cluster size and number.

Clustering algorithms for UWSNs are still being improved upon in terms of energy economy, scalability, and flexibility to the dynamics of the underwater environment. Machine learning has the potential to facilitate more astute and anticipatory grouping choices that are attuned to the demands of application quality of service. All things considered, clustering is still a crucial method for network self-organization to deal with the difficulties of an underwater environment. A brief structure of clustering is provided in Figure 1.

2.2.1 Characteristics of Clustering

The key component of clustering feature in UWSNs is energy efficiency. Sensor nodes have limited battery in underwater and those battery can’t be replaced. So after the battery is consumed the nodes die. That’s where clustering comes in handy as it makes group of nodes and choose a head which will gather the data from every node in group and send the data to base station while saving the energy of nodes. The flexibility of clustering in UWSNs to adjust to the changing underwater topology is one of its unique characteristics. The undersea environment is constantly shifting due to elements including marine life, tides, and ocean currents [Xie, 2024]. To ensure optimal communication pathways, maintain consistent network connectivity, and maintain network stability, clusters need to dynamically modify their settings. It becomes essential to use adaptive clustering methods to react to changes in the underwater environment. One essential component of clustering features is the cooperative behavior of clustered nodes. Within clusters, cooperative communication solutions improve data transmission reliability and enable coordinated reactions to environmental changes [He, 2024].

2.2.2 Challenges in Clustering

The main challenges which occur during clustering in UWSNs are explained in this section, which shed light on the difficulties encountered during research for developing an algorithm. One of the major difficulties comes in data transfer rates and limited bandwidth. To get over these problems clustering algorithms must maximize the data

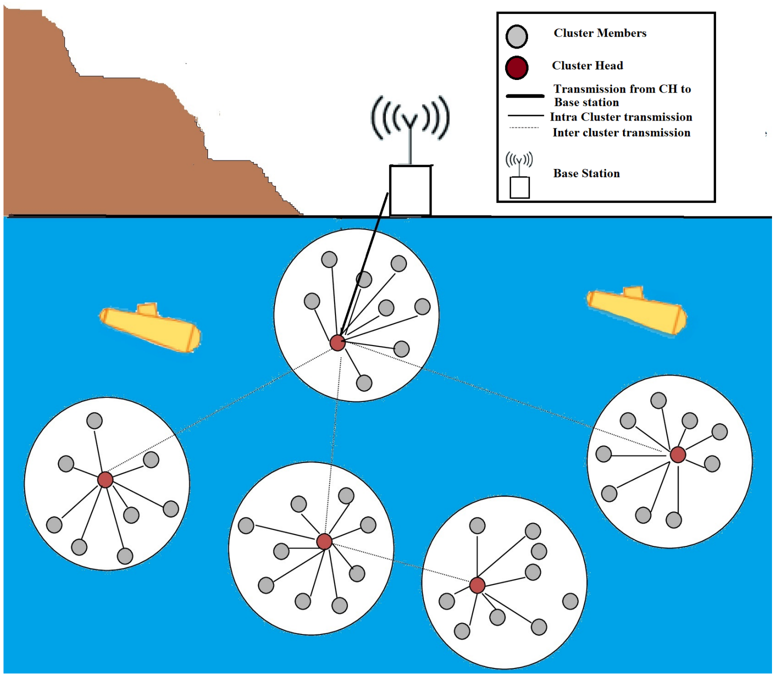

Figure 1: Clustering in UWSNs

aggregation and transmission method [He, 2024]. Another is harsh environment of underwater like high pressure bio fouling and corrosive materials. It is difficult to design a clustering algorithm which takes these aspects into consideration. Another main issue is scalability. In order to handle large number of networks clustering algorithms kave better scalability while preserving optimal performance [Xie, 2024].

3 Related Work

Recent research efforts have tackled numerous critical difficulties in the field of UWSNs, particularly those intended for underwater conditions. Specifically, because sensor nodes have limited energy resources, the problem of optimizing network longevity has become more important. Numerous strategies have been investigated, such as the application of clustering algorithms to arrange nodes and collect information at cluster centers, therefore increasing the network's lifespan overall. So this section provide a brief survey of UWSNs and clustering in UWSNs.

3.1 Literature Survey on Underwater Wireless Sensor Networks

This section provides a brief literature survey on UWSNs as shown below:

- The study in paper [Felemban, 2015] highlights the untapped potential of underwater resources, constituting 70% of Earth, where WSNs face challenges in transitioning to UWSNs. By categorizing UWSN applications into monitoring, disaster response, military, navigation, and sports, the paper emphasizes recent developments and acknowledges challenges faced in adapting communication methods to diverse subsea environments.

- In [Davis, 2012],the authors explores how to modify Terrestrial Wireless Sensor Networks (TWSNs) for use in underwater environments by utilizing low-power and electrical improvements. It demonstrates how adaptable UWSNs are at tracking environmental occurrences. Recognizing obstacles, the study offers a succinct overview of current UWSN topologies implemented to surmount technological barriers.

- In [Luo, J., 2020], a mobility-assisted localization scheme (MALS-TSF) for 3D UWSNs is proposed. It strategically places anchor nodes to optimize cost and introduces a time synchronization-free approach for unknown sensor nodes. In Phase II, it employs two-way TOA for remaining nodes, considering underwater drift. Simulation results show MALS-TSF's effectiveness in achieving high localization without time synchronization.

- Paper [Gkikopouli, 2012] gives a brief overview of UWSN technologies while recognizing their unique traits and weaknesses. With an emphasis on practical applications, it describes important research areas such as architecture, routing, MAC, localization, energy consumption, and security.

- In [Luo, 2018], the paper describes the drawbacks of the present generation of UWSNs and the transition to the next generation. It highlights how important it is to have programmable, resilient, and resource-sharing networks. One disruptive development in technology is the rise of software-defined technologies, such as network function virtualization and software-defined radio. In order to create next-generation UWSNs that are software-based and service-oriented, the article hopes to stimulate more research in these fields.

- A survey in [Zeng, 2014] shows deployment algorithms for UWSN, essential for gathering data on the water and tracking the environment. It discusses the special 3D features of UWSN, classifying deployment techniques into 2D and 3D underwater networks, as well as the deployment of gateway nodes. The study briefly analyzes different schemes and discusses the benefits and drawbacks of each in relation to the deployment of UWSNs.

- Paper [Heidemann, 2012] discusses important applications, acoustic propagation processes, and how they affect communication system design as it examines the architecture and difficulties of underwater wireless sensor networks. It also provides researchers with a synopsis of testbeds, simulation tools, and communications hardware that is currently accessible.

3.2 Literature Survey on Clustering in UWSNs

This section provides a brief literature survey on some of clustering algorithms as shown below:

- The paper [Tiwari, 2023] addresses restricted battery resources by optimizing CH selection and suggests ECERO for UWSN. Energy, distance from the sink, node density, and acoustic path loss are all taken into account by ECERO, which also adds sleep scheduling to lessen the stress on CH nodes. Enhancing EOCSR, ECERO efficiently reduces hot-spot issues brought on by relaying massive amounts of data in UWSN. sink/base station.

- In paper [Khan, 2021] propose ANC-UWSNs using Dragonfly Optimization (DFO). Under difficult underwater conditions, DFO outperforms ACO, CLPSO, GWO, and MFO in terms of cluster size optimization, routing efficiency enhancement, and network lifespan extension.

- In paper [Fei, 2020], authors uses Fuzzy C Means (FCM) for energy-efficient clustering and Moth-Flame Optimization (MFO) for optimal Cluster Head (CH) selection, this study provides FCMMFO, a novel technique to address battery limits and communication delays in UWSNs. The findings of the simulation confirm that FCMMFO is more energy efficient than current algorithms, highlighting its significant contribution to improving UWSN performance.

- Low-Energy Adaptive Clustering Hierarchy (LEACH) [Heinzelman, 2000] is the most common algorithm implemented within the area of wireless sensor networks. Energy-constrained situations in which they will be implemented favor its effectiveness. Frequent changes of the head cluster tend to consume extra energy and sometimes incur poor connectivity among its members.

- In paper [Li, 2007] the authors introduce a refined version of the LEACH clustering technique for hierarchical structures in extensive underwater acoustic sensor networks is presented in this research. Simulation findings demonstrate that the improved algorithm has a good effect on network performance, improving stability and energy utilization.

- Power-Efficient GAthering in Sensor Information System (PEGASIS) [Lindsey, 2002] aims to reduce the number of clusters and improve network lifetime by creating a chain of nodes, which is particularly suitable for large-scale UWSNs. It optimizes energy consumption by reducing the number of cluster heads, but it can struggle with node mobility and network density.

- Hybrid Energy-Efficient Distributed Clustering (HEED) [Younis, 2004] is an improvement over LEACH, where cluster heads are selected based on both energy consumption and node proximity. It provides a balance between energy efficiency and clustering stability. However, it may not be the best choice for highly dynamic environments with variable communication conditions, such as those in UWSNs.

- The Hierarchical Energy-Efficient Routing (HEER) [Yi, 2016] protocol is a clustering-based routing strategy designed to enhance energy efficiency in UWSNs. HEER organizes nodes into hierarchical clusters, where data is aggregated at cluster heads and progressively forwarded to higher levels until reaching the base station. This multi-level architecture balances energy consumption across the network by periodically rotating cluster head roles based on residual energy and proximity. HEER significantly reduces communication overhead and delay by exploiting localized data aggregation without losing its scalability for large-scale deployment. While HEER outperforms other approaches such as LEACH and PEGASIS in terms of energy efficiency and achieved throughput, REAFCA has slightly higher overhead than HEER due to the recursive manner of cluster management.

- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [Ester, 1996] is the algorithm used to find the cluster; it selects the dense area of nodes to form the cluster, thus perfectly suited for underwater wireless sensor networks with dynamics and irregularity. The clusters cannot be pre-define; DBSCAN defines its own parameters as minimum points and neighborhood radius that allow clusters formation according to the node density. This characteristic makes DBSCAN efficient in handling non-uniform node distributions and outlier detection, which is a common phenomenon in UWSNs due to node mobility and environmental factors. However, DBSCAN can be computationally expensive, especially in 3D underwater environments, and may not perform well with varying node densities. While it adapts well to heterogeneous deployments, energy consumption and delay may be higher than protocols like REAFCA due to iterative neighborhood searches and clustering overhead.

- Paper [Krishnaswamy, 2021] presents a fuzzy logic-based approach for Underwater Wireless Acoustic Sensor Networks (UWASNs) trusted cluster head and member node selection. The method uses fuzzy criteria like relative mobility, energy, and distance to determine trust levels, and it uses lightweight XOR encryption for security tests. The strategy, which has been simulated in MATLAB, beats current approaches and improves packet delivery and malicious node identification in UWASNs.

- In paper [Tran, 2014], the authors presents a round-based clustering scheme for Underwater Wireless Sensor Networks, addressing challenges like redundant data transmission. Through initialization, cluster-head selection, clustering and data aggregation phases, the scheme efficiently manages data, improving network throughput and ensuring a minimum percentage of data reaches the sink/base station.

- The study in paper [Bhatti, 2016] suggests a fuzzy c-means (FCM) clustering as a method for energy-efficient cooperative spectrum sensing. Using the factors of sensor location, distance to the fusion center, SNR, and residual energy, the method maximizes both cluster formation and cluster head (CH) selection. As opposed to traditional clustering techniques, it minimizes energy consumption and improves network efficiency in cooperative spectrum sensing with better detection probability under defective channels.

- The paper [Subramani, 2022] proposes a new clustering and routing protocol MCR-UWSN based on meta heuristics that is intended to improve energy efficiency in underwater wireless sensor networks (UWSNs). It efficiently chooses cluster heads and routes data by employing MHR-GOA for multi-hop routing and CEPOC for clustering. Its improved performance over current methods is demonstrated by experimental validation, which makes it a viable option for energy-efficient UWSNs. This paper proposes an equation for fitness of sensor nodes which will be enhanced and used for clustering algorithm which is proposed in this paper. So the equation 1 for fitness of sensor nodes are as follows:

$$F_{\text{fitness}} = w_1 \times a_1 + w_2 \times a_2 \tag{1}$$

  Here F indicate the fitness of the node where w_1 indicates the weight assigned to $\text{delta}_{\text{difference}}$, and w_2 means the weight assigned to $\text{distance}_{\text{neighbour}}$ whereas a_1 and a_2 are coefficient for delta and distance respectively.

- In paper [Priyanga, 2020], authors addresses the challenges of power supply, limited bandwidth, and high propagation delay in UWSNs. The authors propose an Energy Efficient Distributed Unequal Clustering (EEDUC) algorithm with relay node selection(RNS) to enhance network performance and energy efficiency.

  Table 1 shows all the findings, limitations, simulation tools, and techniques used in these papers briefly.

Ref	**Technique Used**	**Simulation tools**	**Findings**	**Limitations**
[Li, 2007]	Improved Clustering Algorithm LEACH-L	NS-2	Enhanced stability and energy utilization of network	Clustering size and consumption of energy
[Krishnaswamy, 2021]	Trust based security scheme in UWASN	MATLAB (2017b)	Performed better from compared algorithms	Security problems of sensor nodes
[Priyanga, 2020]	EEDUC and RNS	NS 2.30	High energy efficiency and better packet delivery ratio	Scalability and adaptive ability
[Tiwari, 2023]	ECERO	MATLAB	Out perform other routing protocols	Network is assumed so results can change
[Khan, 2021]	DFO	MATLAB	Adaptive node Clustering protocols	More consumption of power and residual energy
[Tran, 2014]	Round based Clustering	QualNet 5	High throughput and low energy consumption	Collision can occur
[Subramani, 2022]	MCR-UWSN	NS-2.34	Enhanced energy efficient performance	Data aggregation and underwater object tracking
[Fei, 2020]	FCM and MFO	Matlab	Reduces energy consumption of entire UWSNs	Big time intervals for selecting Cluster Heads

Table 1: A comparative study on Clustering algorithms in UWSNs

3.3 Research Gaps Justification

While much work has been done in resource allocation and clustering for wireless networks, recent work in multi-agent systems and communication networks points to key areas that are still underexplored. Specifically:

- Dynamic Resource Allocation: Traditional approaches do not have the capability to adapt in real-time with changes in network conditions and hence, lead to inefficiencies in power consumption, throughput, and interference management. The recent studies in [Wang, 2024] introduced real-time adaptability that may considerably improve the flexibility of clustering algorithms such as REAFCA in dynamic environments.
- Interference Cancellation and Power Efficiency: The advancement of sophisticated technologies like multi-IRS empowered UAV communication presents a possibility to advance network throughput and power efficiency [Gu, 2024]. When integrated into underwater wireless sensor networks, these advancements help alleviate issues like high latency and signal degradation, ultimately resulting in cluster stability as well as energy efficiency improvement.
- Queue stability and throughput maximization: Research into the area of recent queue stability and dynamic throughput maximization studies, like multi-agent heterogeneous wireless networks, would shed more light on enhancing reliability in communication while minimizing delays in UWSNs [Yang, 2024]. It is essential to achieve optimal performance over an extended duration of network deployment.

Research on [Wang, 2024] may bring dynamic adaptability mechanisms to REAFCA, allowing it to adapt to real-time changes in network load and node behavior, further optimizing energy efficiency and network longevity.

The study on [Gu, 2024] may provide interference management and energy optimization insights which may be applied to make improvements on the performance of REAFCA in difficult underwater environments.

Including results of [Yang, 2024] could help in designing an enhanced throughput and reduced data loss that can be directly used in cluster head selection and aggregation of data in UWSNs.

4 Proposed Residual Energy-Aware Fuzzy-Based Clustering Algorithm (REAFCA)

A brief overview of the proposed Residual Energy-Aware Fuzzy- Based Clustering Algorithm (REAFCA) for underwater wireless sensor networks with system model, assumption, and protocol overview is presented in this section.

4.1 System Model

This section provides a system model for UWSNs which is shown in Figure 2 where few nodes are deployed and scattered in underwater. Cluster formation will be done on this model to form cluster and choose cluster heads according to the proposed work and all the assumptions which is shown below.

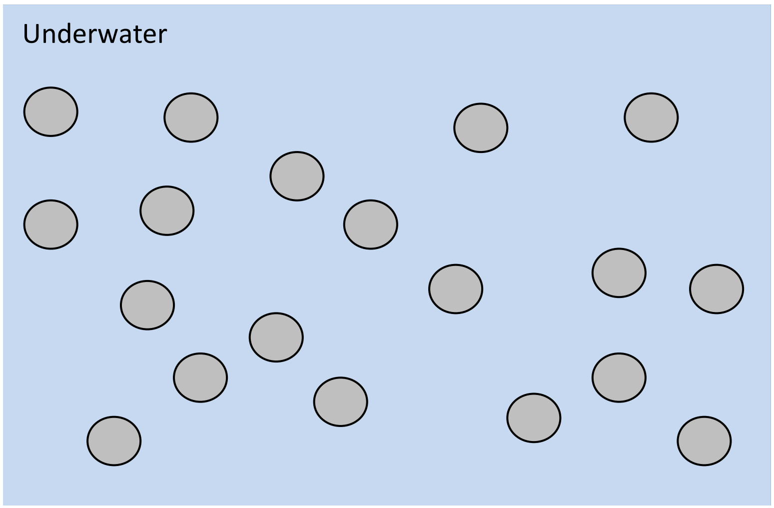

Figure 2: System Model for Clustering in UWSNs

4.2 Assumptions

The following assumptions are used in designing the algorithm:

- N sensor nodes are randomly deployed in a 3D underwater environment with the same initial energy level.
- Each sensor node is stationary after deployment, except for minor movement due to underwater currents which are captured by Angular Velocity. Angular Velocity value reflects the node mobility. It is the rate at which a specific node changes its position. Minimum angular velocity reflects the stationary node and maximum angular velocity reflects the fastest mobile node. The unit of angular velocity is radians/second.
- Nodes use full-duplex links for communication using acoustic signals within a limited range, experiencing propagation delays, noise, and multipath effects that are uniformly distributed across the network area.
- Sensor nodes which act as cluster members consume energy for transmission, reception, and data aggregation. The Cluster heads consume more energy than cluster member because of their additional data aggregation and communication activities.
- Sensor nodes know their residual energy, location, and neighbors; cluster heads are reselected periodically to balance energy consumption in the networks.
- Fuzzy rules are applied for ranking and threshold calculations that adapt dynamically to network conditions.

4.3 Protocol Overview

The Residual Energy-Aware Fuzzy-Based Clustering Algorithm (REAFCA) is divided into three different phases, viz., i) Phase 1: Cluster Formation and Probable Cluster Head

Identification, ii) Phase 2: Fitness Calculation and Cluster Head Selection, and iii) Phase 3: Intra-Cluster Communication, Cluster Head Selection and Finalization as shown in Figure 3.

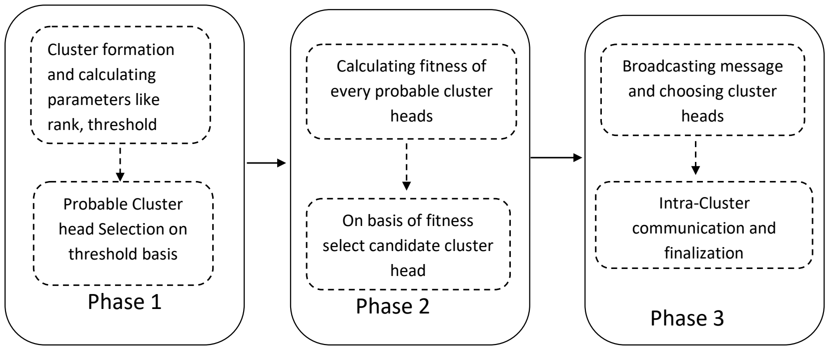

Figure 3: Block Diagram for Residual Energy-Aware Fuzzy-Based Clustering Algorithm (REAFCA)

In ”Phase 1: Cluster Formation and Probable Cluster Head Identification”, it starts by forming clusters and then it identifies probable cluster heads based on node parameters like radius, residual energy, and rank, broadcasting their information to facilitate cluster head selection. After that in ”Phase 2: Fitness Calculation and Cluster Head Selection”, it calculates fitness values for each nodes and then it evaluates received messages from probable and candidate cluster heads, considering factors such as fitness and energy consumption thresholds, to select high-quality cluster heads. It updates the candidate cluster head list accordingly, ensuring efficient cluster head selection. And, in ”Phase 3: Intra-Cluster Communication and Finalization” manages intra-cluster communication tasks within UWSN clusters. It selects cluster heads based on parameters like residual energy and fitness, ensuring optimal cluster leadership. After selecting cluster heads, it performs the necessary finalization steps to ensure the integrity and efficiency of the formed clusters.

A detailed explanation of all these phases is given in the subsequent subsections with a flowchart as shown in Figure 4 depicting the features of all the phases.

4.3.1 Phase 1: Cluster Formation and Probable Cluster Head Identification

The Cluster Formation and Probable Cluster Head Identification phase act as the foundational structure of the proposed REAFCA algorithm. The first step of this algorithm is to form a cluster which helps similar types of nodes to form a connection and helps in data gathering and data transfer. The cluster formation method helps to improve network performance and flexibility in difficult underwater settings by carefully arranging and coordinating network components. This phase clusters the nodes in the network based on proximity, residual energy, and stability. Fuzzy logic plays a very significant role in the decision-making process by managing uncertainties and finding the best possible balance of various parameters to effectively form the clusters. After this, the next step of

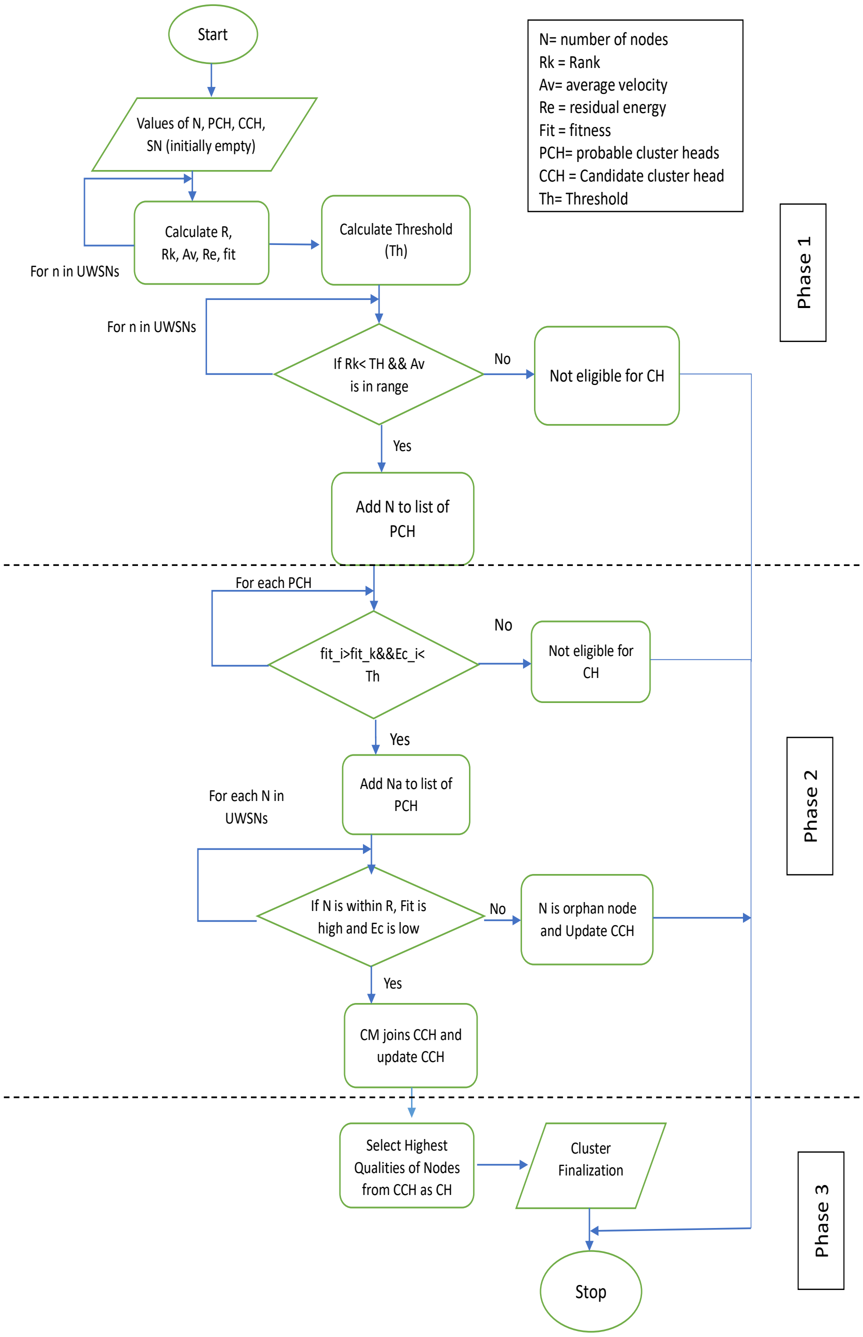

Figure 4: Flow chart for Residual Energy-Aware Fuzzy-Based Clustering Algorithm (REAFCA)

this algorithm is to calculate some parameters to check which sensor node is suitable for the cluster member or list of probable cluster heads. The parameters that are to be calculated are radius (distance of node to the base station), angular velocity as a measure of node mobility) and residual energy of each sensor node. The parameters are briefly explained with suitable equations as follows.

Radius: The radius is the distance of a sensor node from the origin point to the coordinated system as shown below in equation 2. Radius is essential for spatial analysis, localization and or range of the node. It is calculated based on the cartesian coordinates (x, y).

$$\text{Radius} = \sqrt{x^2 + y^2} \tag{2}$$

Let, Radius_{node} is the distance of a specific node from a cluster head and Radious_{max} is the maximum radius across all the nodes in the networks, the normalized radius can be calculated as

$$\text{Normalized Radius} = \frac{\text{Radius}_{node}}{\text{Radius}_{max}} \tag{3}$$

Angular Velocity: In underwater the nodes can not be stable and this affect the nodes so angular velocity is a main aspect while calculating the cluster head. It is calculated on the basis of rate of change of an object with respect to time. By calculating this network manager can predict node trajectories and optimize cluster formation and communication strategies accordingly as shown below in equation 4.

$$\text{Angular Velocity} = \frac{\Delta\theta}{\Delta t} \tag{4}$$

If $\text{Angular Velocity}_{node}$ is the angular velocity of a specific node and Velocity_{max} is the maximum velocity across all the nodes in the networks, the normalized angular velocity can be calculated as

$$\text{Normalized Angular Velocity} = \frac{\text{Angular Velocity}_{node}}{\text{Velocity}_{max}} \tag{5}$$

Residual Energy: It is the total remaining energy left in a sensor node as shown in equation 6. While deployment the nodes are fully charged but the battery is irreplaceable so after some time nodes start to lose there energy and residual energy comes in handy while choosing a cluster head.

$$\text{Residual Energy} = \text{Initial Energy} - \text{Energy Consumption} \tag{6}$$

Let, $\text{Residual Energy}_{node}$ is the residual energy of a specific node, and Energy_{max} is the maximum residual energies across all the nodes in the networks, the normalized residual energy can be calculated as

$$\text{Normalized Residual Energy} = \frac{\text{Residual Energy}_{node}}{\text{Energy}_{max}} \tag{7}$$

Each of the above-mentioned parameters is fed into a fuzzy logic system as input variables, with membership functions. The fuzzy logic system evaluates the parameters and assigns a suitability degree to each node instead of binary thresholds, using gradual transitions, which helps develop more robust and adaptive clusters than the previous method. For example, nodes with moderately high energy and favorable locations will still be included as cluster heads, thus ensuring cluster stability.

The fuzzy logic system calculates the rank of each node and compares it to the threshold value which is calculated from the parameters that are dynamically adjustable to select a balanced number of candidates. Finally, on this entire basis nodes are selected as a probable cluster head. The formula used to calculate rank and threshold are given below.

Rank: It is the primary factor while choosing sensor nodes because the threshold will be compared to rank and on that basis, only the cluster head is chosen.

$$\begin{aligned} \text{Rank} = \text{Weight}_{\text{residual energy}} \times \ \text{Normalized} ResidualEnergy + \\ \text{Weight}_{\text{angular velocity}} \times \ \text{Normalized Angular Velocity} + \\ \text{Weight}_{\text{radius}} \times \ \text{Normalized Radius} \end{aligned} \tag{8}$$

Here for calculating rank residual energy, angular velocity and radius of each node is added with there weighted value and that will give a rank which is shown in equation 8.

If $Rank_{node}$ is the computed rank of a node and $Rank_{max}$ is the maximum rank across all the nodes in the networks, the normalized rank can be calculated as

$$\text{Normalized Rank} = \frac{Rank_{node}}{Rank_{max}} \tag{9}$$

Let, S_{size} be the scaling factor for network size, where $N_{current}$ is the number of nodes in the present network, and N_{max} is the maximum number of possible nodes in the network. Therefore, S_{size} can be calculated as follows:

$$\text{S}_{\text{size}} = \frac{N_{current}}{N_{max}} \tag{10}$$

Let, $S_{density}$ be the scaling factor for network density, where Average Node Degree$_{current}$ is the average degree a node shifts in the network, and Node Degree$_{max}$ is the maximum degree a node can shift in the network. Therefore, $S_{density}$ can be calculated as follows:

$$\text{S}_{\text{density}} = \frac{\text{Average Node Degree}_{current}}{\text{Node Degree}_{max}} \tag{11}$$

Let, Network Density$_{node}$ represents the network density of a node which is defined as the number of neighboring nodes within its communication range divided by the total coverage area by the network. It measures how nodes are distributed in a network. The Network Density$_{node}$ is calculated as

$$\text{Network Density}_{node} = \frac{\text{number of neighboring nodes}}{\pi \times \text{range}^2} \tag{12}$$

Let, Density$_{max}$ is the maximum density across all the nodes in the networks, the normalized network density can be calculated as

$$\text{Normalized Network Density} = \frac{\text{Network Density}_{node}}{\text{Density}_{max}} \tag{13}$$

If Network Load_{node} is the network load of a node and Load_{max} is the maximum load across all the nodes in the networks. The Network Load_{node} can be calculated as

$$\text{Network Load}_{node} = \frac{\text{number of packet handled by a node}}{\text{total time period}} \tag{14}$$

Then, the normalized network load can be calculated as

$$\text{Normalized Network Load} = \frac{\text{Network Load}_{node}}{\text{Load}_{max}} \tag{15}$$

Threshold: The threshold calculation in the algorithm is crucial for identifying probable cluster heads based on a node's rank and angular velocity. The threshold (T_{prob}) is computed using the formula:

$$\begin{aligned} \text{T}_{\text{prob}} = \text{S}_{\text{size} \times \text{S}_{\text{density}} \times (}\text{Weight}_{\text{rank}} \times \text{Normalized Rank}+ \\ \text{Weight}_{\text{angular velocity}} \times \text{Normalized Angular Velocity}+ \\ \text{Weight}_{\text{density}} \times \text{Normalized Network Density}+ \\ \text{Weight}_{\text{load}} \times \text{Normalized Network Load}) \end{aligned} \tag{16}$$

Here, the weights ($\text{Weight}_{\text{rank}}$, $\text{Weight}_{\text{angular velocity}}$, $\text{Weight}_{\text{density}}$ and $\text{Weight}_{\text{load}}$) are parameters that determine the significance of rank, angular velocity, network density and network load in the clustering decision. The normalized values represent the scaled values of rank and angular velocity to ensure a consistent range for comparison. These values are calculated in the algorithm 2 and the formula is shown in equation 16.

Lastly, this algorithm broadcasts a message containing the identity, radius, angular velocity, residual energy, and rank of each probable cluster head to the network.

4.3.2 Phase 2: Fitness Calculation and Cluster Head Selection

The Fitness Calculation and Cluster Head Selection phase uses fuzzy logic to determine the best cluster heads from the probable candidates. This ensures that the selected cluster heads are the best fit for their roles in terms of network coverage, energy efficiency, and stability. The 2nd phase of the algorithm starts with calculating the fitness of the nodes. The fitness function is a weighted combination of normalized values of Radius (F_{nr}), Residual Energy (F_{ne}), Rank (F_{nk}), Normalized Angular Velocity (F_{nv}), Normalized Network Density (F_{nd}), and Normalized Network Load (F_{nl})) respectively. The weights ($Weight_{\text{radius}}$, $Weight_{\text{residual energy}}$, $Weight_{\text{rank}}$, $Weight_{\text{angular velocity}}$, $Weight_{\text{density}}$, and $Weight_{\text{load}}$) determine the relative importance of each parameter in the overall fitness calculation in determining the suitability of a node to act as a cluster head. Here's an explanation of the components in the formula:

- *Normalized Radius (F_{nr})*: For each node, the scaled value of its radius is represented by its normalized radius. This standardization guarantees that, when evaluating fitness, changes in radius amongst nodes are taken into consideration consistently.
- *Normalized Residual Energy (F_{ne})*: The scaled representation of a node's leftover energy is called the normalized residual energy. It is possible to compare energy

Algorithm 1 Cluster Formation and Probable Cluster Head Identification

Input: Random sensor nodes
Output: Adjusted communication ranges, energy-efficient routing, finalized clusters, list of probable cluster heads

```
Function cluster_formation_probable_ch
Cluster Formation and Adjustment
Implement a mechanism for adjusting communication ranges within the cluster
Introduce energy-efficient data aggregation and routing approaches
Perform cleanup and finalization steps
Probable Cluster Head Identification and Broadcast
for Each Node in UWSN do
    Calculate radius, angular velocity, and residual energy
    Calculate rank of the node
end for
T_prob ← Threshold rank to become probable CHs
for Each Node in UWSN do
    if rank of Node < T_prob and angular velocity is within a certain range then
        Add Node to the list of probable CHs
    end if
end for
for Each Probable CH in probable CH list do
    Broadcast Probable CH Message(ID, radius_i, rank_i, angular_velocity_i, residual_energy_i)
end for
```

levels among nodes fairly by normalizing this metric, regardless of their original energy capacity.

- *Normalized Rank (F_{nk})*: A scaled representation of a node's rank is called the normalized rank. Normalization of rank makes an objective evaluation easier by taking network-wide rank variances into account.
- *Normalized Angular Velocity (F_{nv})*: A scaled version of a node's angular velocity normalized to the scale [0, 1] for the evaluation of stability in the network.
- *Normalized Network Density (F_{nd})*: This is a scaled version of the number of neighboring nodes around a node normalized on the scale [0, 1] for measuring local connectivity.
- *Normalized Network Load (F_{nl})*: A scaled measure of the communication load on a node, normalized to [0, 1] to indicate its burden in handling network traffic.
- Weights ($Weight_{\text{radius}}$, $Weight_{\text{residual energy}}$, $Weight_{\text{rank}}$, $Weight_{\text{angular velocity}}$, $Weight_{\text{density}}$, and $Weight_{\text{load}}$): Their relative significance in the overall fitness computation is determined by the weights assigned to these factors. Assigning and adjusting these weights can significantly influence the algorithm's performance. A dynamic weight adjustment mechanism can further enhance adaptability by using

real-time network feedback and fuzzy logic system to adjust the weights of different parameters dynamically according to conditions such as node mobility, energy distribution, network density, or depending on the particular needs of the application or network environment (for example:, i) prioritizing energy efficiency in energy-constrained scenarios, ii) higher weights may be assigned to angular velocity in high mobility environments, or iii) higher weight can be assigned to node radius in sparse networks. One way to provide customized fitness evaluations is by varying the weights, which can affect how each parameter affects the overall fitness value.

The fitness formula is derived from equation 1 and modified according to the REAFCA algorithm. The fitness value (F_n) for each node is calculated using the formula shown in equation 17.

$$\mathrm{F}_n = \text{Weight}_{\text{radius}} \times F_{\text{nr}} + \text{Weight}_{\text{residual energy}} \times F_{\text{ne}} + \text{Weight}_{\text{rank}} \times F_{\text{nk}} +$$

$$\text{Weight}_{\text{angular velocity}} \times F_{\text{nv}} + \text{Weight}_{\text{density}} \times F_{\text{nd}} + \text{Weight}_{\text{load}} \times F_{\text{nl}} \tag{17}$$

After calculating fitness the algorithm receives Probable Cluster Head (CH) Messages from nearby nodes, the algorithm examines each sender node's energy usage and fitness. A sender node is included in the candidate CH list and becomes a candidate for the cluster head position if it demonstrates greater fitness and uses less energy than a predetermined threshold. Once the candidate CH list is filled, the algorithm sends Candidate CH Messages to nearby nodes in order to tell them about possible cluster head candidates. In the meantime, the algorithm determines whether nodes that satisfy the necessary energy and fitness requirements and are within the communication radius of the candidate cluster head are eligible to receive Candidate Cluster Head Messages from other nodes expressing their candidacy. Nodes that satisfy these requirements are invited to join the candidate cluster head and aid in the establishment of the cluster. Nodes that don't match the requirements are considered orphans and aren't included in the clustering process. In order to promote effective cluster formation and administration inside the UWSNs, this methodical approach to cluster head selection guarantees that nodes with ideal fitness and energy characteristics are discovered and assigned as cluster heads.

4.3.3 Phase 3: Intra-Cluster Communication and Finalization

The "Intra-Cluster Communication and Finalization" algorithm plays a crucial role in coordinating communication activities across clusters and securing cluster configurations. The algorithm guarantees the effective execution of intra-cluster communication activities, like data aggregation and exchange, by methodically iterating over every node in the cluster. By using resources as efficiently as possible, this procedure improves the cluster's total data processing and transmission efficiency. Important factors including residual energy, communication range, angular velocity, and fitness are all taken into consideration by the algorithm when evaluating potential cluster heads. Then cluster heads are selected and broadcast their status to all the nearest nodes within their potential communication range and nodes join the nearest cluster head based on fitness value. The use of fuzzy logic during phase 1 and phase 2 ensures that the chosen cluster heads not only are energy efficient but also strategically located in such a way that it can minimize communication overhead and delay. This phase also involves dynamic adjustment

Algorithm 2 Fitness Calculation and Cluster Head Selection

Input: Node parameters, including radius, residual energy, and rank
Output: Fitness values for nodes and selected cluster heads

Function fitness_and_cluster_head_selection
Fitness Calculation
Normalize node parameters (radius, residual energy, rank)
Calculate weighted fitness values for nodes based on normalized parameters
Handle Cluster Head Messages and Selection
for each received Probable CH Message from node k **do**
  if fitness_i $>$ fitness_k **and** energy_consumption_i $<$ threshold **then**
    Add node k to candidate CH list
  end if
end for
if Candidate CH list not empty **then**
  for each Candidate CH c in list **do**
    Advertise Candidate CH Message($c.ID$)
  end for
end if
for each received Candidate Cluster Head Message from node k **do**
  if node within radius r_c **and** meets fitness and energy thresholds **then**
    Node joins the Candidate CH
  else
    Node is orphan node
  end if
end for
Update Candidate CH list

in cluster parameters supported by fuzzy decision-making for maintaining stable and reliable communication despite the fluctuations in the environment. For example, i) nodes with good connectivity and higher energy receive larger ranges to minimize the delays in communication, and ii) nodes having very low residual energy are forbidden to engage in minimal communications. The algorithm carefully selects the best candidate to oversee the coordination and intra-cluster communication after evaluating the potential cluster heads. The cluster heads collect data from member nodes to minimize redundant transmissions. Fuzzy logic assists in the priority of critical data based on metrics such as timestamp, type of data, and load on the network. Then, the cluster head transmits the aggregated data to the base station over the most energy-efficient path. If the direct path is not available, it uses multi-hop communication where the intermediate nodes are selected based on fuzzy-optimized criteria such as residual energy and connectivity. In order to ensure the cluster configurations' robustness and preparedness for operation in the UWSN environment, the algorithm finally completes key finalization processes in which periodic re-clustering maintains energy balance and adaptability in the network. The re-clustering period changes dynamically using fuzzy logic with respect to network conditions (e.g., high frequency in highly mobile environments). In the finalization step, the REAFCA uses multi-path routing, forward error correction, and data aggregation techniques to mitigate communication latency in dynamic underwater environments. The techniques have been discussed in the next sub-section. To put it simply, the ”Intra-Cluster Communication and Finalization” algorithm is essential to improving communication reliability and cluster performance in UWSNs, which in turn improves the network’s overall efficacy.

Algorithm 3 Intra-Cluster Communication and Finalization

Input: Nodes within the cluster of UWSN
Output: Intra-cluster communication tasks executed, cluster heads selected, finalization steps performed

Function intra_cluster_communication_and_finalization
Intra-Cluster Communication
for each Node in the cluster **do**
    Perform intra-cluster communication tasks
end for
Cluster Head Selection
Evaluate candidate cluster heads based on parameters such as residual energy, range, angular velocity, and fitness
Select the highest-quality cluster head from the candidate list
Finalization
Perform necessary finalization steps for the clusters
Function Multi-path routing
Function Forward error correction
Function Data aggregation

4.4 Techniques for Reducing Delay by REAFCA

UWSNs suffer from challenges of communication latency caused by acoustic noise, long propagation delays and multipath effects. Reducing delay is important to enable real-time applications such as underwater surveillance, monitoring, and disaster management. To mitigate communication latency in dynamic underwater environments, REAFCA uses multi-path routing, forward error correction, and data aggregation techniques in the ”Phase 3: Intra-Cluster Communication and Finalization” of REAFCA. The techniques have been discussed as follows.

– Multi-path routing spreads data packets over multiple paths simultaneously to avoid delays due to congestion. This ensures that data reaches its destination promptly even in the case of high interference and node failure. REAFCA integrates multi-path routing by dynamically identifying alternative routing path in phase 3. Then, the REAFCA allocates traffic across these alternative paths based on path quality metrics like delay and fitness value.

– Forward error correction mechanism is used to detect and correct errors at the receiver at the phase 3 to enhance the reliability metrics by reducing retransmissions of lost or corrupted packets due to acoustic noise, signal interference, and multipath propagation.

- Data aggregation technique is used to reduce intra-cluster communication delays. In this technique, the cluster head reprocess data received from its member nodes and aggregate them before transmitting to the nearest base station, reducing overall size of the transmitted data.

4.5 Novelty and Benefits of REAFCA

The Residual Energy-Aware Fuzzy-Based Clustering Algorithm (REAFCA) brings forth significant improvements over the traditional clustering algorithms like LEACH, HEED, and other state-of-the-art methods, especially for the challenges of Underwater Wireless Sensor Networks (UWSNs). By using a fuzzy-based approach, REAFCA achieves better energy efficiency, stability, adaptability, and scalability, making it a perfect solution for the harsh underwater environment. Below are the key points that highlight its novelty and benefits:

1. Multiple Critical Factors for Consideration: Unlike LEACH that is mainly based on residual energy for the selection of the cluster head (CH) and HEED using probabilistic approaches based on residual energy and proximity, the multiple critical factors such as residual energy, node radius, angular velocity, and rank of nodes are considered in CH selection in REAFCA. Using a fuzzy logic system dynamically to evaluate these parameters helps in the making of more judicious decisions well suited for the unpredictable environment under the sea.

2. Adaptive Clustering Using Fuzzy Logic: REAFCA employs fuzzy logic for adaptive adaptation to the dynamic nature of underwater conditions, like changing node mobility, energy, and communication reliability. Such adaptability is contrary to traditional algorithms that use static or threshold-based methods, which in most cases are not equipped to handle the complexity and unpredictability of underwater topology and communication challenges.

3. Increased Energy Efficiency: The key here is that REAFCA optimizes energy consumption by preferring nodes with higher residual energy and better fitness scores, calculated through a weighted combination of parameters. Dynamic CH selection in REAFCA ensures that energy utilization in the network is well balanced, and this has greatly increased network lifetime. LEACH, however, suffers from frequent and energy-intensive CH reassignments, while HEED lacks mechanisms to adjust dynamically to environmental changes.

4. Improved Stability and Longevity: The fuzzy logic-based CH selection ensures that the chosen CHs are stable and well-suited for their roles, thereby reducing the frequency of re-clustering. This stability minimizes communication overhead and delays, addressing the high latency and limited bandwidth inherent in underwater environments.

5. Optimized Communication and Data Aggregation: REAFCA includes features that enhance the efficiency of intra-cluster communication and data aggregation by minimizing redundant transmission and ensuring that nodes and CHs can communicate in a reliable manner. It overcomes specific underwater challenges such as high propagation delays, acoustic noise, and multipath interference.

6. Scalability and Suitability for 3D Deployment: The algorithm targets three dimensional deployments and dynamics of topologies present in the UWSNs; thus, it can scale to varying-size networks. It does better in the complex multi-levelled underwater

environment, because unlike other algorithms which have been especially optimized for 2D terrestrial networks.

7. Simulation-Validated Supremacy: Simulation testing reveals that REAFCA is superior to algorithms like LEACH, HEED, and K-meansA on critical parameters as well: I) REAFCA exhibits significantly less energy usage by uniformly distributing CH selection and hence, reduced re-clustering. ii) The algorithm achieves higher throughput, even in dense networks, by ensuring efficient communication. iii) REAFCA achieves lower delays, making it well-suited for real-time applications like underwater monitoring and surveillance.

8. Unique Contributions to UWSNs: REAFCA addresses underwater-specific challenges—limited energy resources, high latency, and dynamic topologies—more effectively than existing algorithms. Its multi-dimensional evaluation through fuzzy logic ensures robustness, while its dynamic adaptability improves overall network performance and longevity.

4.6 Computational Complexity Analysis

The computational complexity of an algorithm depends on the number of processes it uses. The complexity analysis can guide future real-time applications like underwater monitoring or surveillance where resource limitations are critical. Below is a computational complexity analysis of REAFCA with K-means and K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER.

REAFCA: To analyze the overall computation complexity of the REAFCA, we have to analyze the computation complexity of the three main phases of REAFCA, viz., i) Phase 1: Cluster Formation and Probable Cluster Head Identification, ii) Phase 2: Fitness Calculation and Cluster Head Selection, and iii) Phase 3: Intra-cluster Communication and Finalization. Each node in Phase 1 calculates rank, and threshold to identify probable of cluster heads. This phase includes various numbers of parameters such as node radius, angular velocity, and residual energy and involves applying fuzzy logic rules. Let, P is the number of parameters and F is the number of fuzzy logic rules. Then, the complexity for Phase 1 can be calculated as O(F.P). Let N is the number of nodes in the network, then the complexity for cluster head selection via sorting the fitness scores in Phase 2 can be calculated as O(N logN). This is because the most efficient sorting algorithms like quick sort and merge sort has a computation complexity of O(N logN). Let k1 be the number of average nodes in each cluster. Then, the complexity of Phase 3 scales linearly as O(k1). Thus, the overall complexity of REAFCA is O(F.P) + O(N log N) + O(k1). The O(k1) can be ignored in the overall computation of complexity analysis as k is typically much smaller than N. Moreover, O(k1) has a negligible impact in comparison to dominating factors like fuzzy logic evaluation in phase 1 and sorting in phase 2 complexities. Therefore, the computational complexity of REAFCA is O(F.P + N logN).

K-means: K-means involves two main phase, viz., i) cluster formation and ii) cluster centroids update. The cluster formation is done by assigning N nodes to k number of clusters iteratively for I number of iterations. Thus, the complexity is calculated as O(N . k. I). The cluster centroids update phase requires O (k2 . D) complexity, where D is the dimension of data. The overall complexity of K-means is O(N. k. I).

K-meansA: K-meansA uses similar phases as in K-means but uses adaptive centroid initialization and weight adjustments, adding an extra factor of W (weight updates) for each iteration in cluster formation. Thus, the overall complexity of K-meansA is O(N. k2. I. W).

LEACH: LEACH involves two phases, viz., i) cluster formation and ii) Data Aggregation and Routing. The complexity of cluster formation is O(N) for the random cluster head selection per round. Let, N is the number of nodes in the networks and k is the number of cluster, where k is typically much smaller than N. The complexity for inter clusters communication that scales linearly is O(k), and data forwarding to the base station is O(1). Thus, the overall complexity of LEACH is O(N + k).

PEGASIS: PEGASIS follows two phases, viz., i) cluster formation using a greedy algorithm that has a computational complexity of $O(N^2)$, and ii) data aggregation and forwarding using linear traversal with a complexity of O(N). Thus, the overall computational complexity of PEGASIS is $O(N^2)$.

HEED: HEED uses iterative cluster head selection during cluster formation with probabilistic metrics, viz., proximity and energy efficiency. Let, I be the number of iterations and N is the number of nodes. Then, cluster selection has the complexity of O(I.N). The next phase of HEED is data aggregation from k number of nodes in a cluster with the complexity of O(k). Therefore, the overall computational complexity of HEED is O(I.N+k).

DBSCAN: DBSCAN uses Density-based clustering for cluster formation that requires a complexity of O(N.logN) for sorting neighbor nodes. If all nodes are in one cluster (worst case), then the complexity for region queries is $O(N^2)$. Therefore, the overall complexity of DBSCAN is $O(N^2)$.

HEER: HEER protocols have two phases: cluster formation and hierarchical communication. Let, I be the number of iterations required for cluster head selection and hierarchy formation. N is the number of nodes. Then, the complexity of 1st phase is calculated as O(I.N). Let, k is the total number of clusters in the network and H be the number of hierarchy levels. Then, complexity for data forwarding across all the hierarchy levels in the 2nd phase scales linearly as O(H.k). Therefore, the overall computational complexity of HEER is O(I.N+H.k).

Algorithm	**Complexity**	**Parameters**
REAFCA	O(F.P+N logN)	F=number of fuzzy logic rules, P=number of parameters, N=number of nodes
K-means	O(I.k.N)	I=number of iterations, k=number of clusters, N=number of nodes
K-meansA	O(I.k.N.W)	I=number of iterations, k=number of clusters, N=number of nodes, W=weight updates
LEACH	O(N+k)	N=number of nodes, k=number of clusters
PEGASIS	$O(N^2)$	N=number of nodes
HEED	O(I.N+k)	I=number of iterations, k=number of clusters, N=number of nodes
DBSCAN	$O(N^2)$	N=number of nodes
HEER	O(I.N+H.k)	I=number of iterations, N=number of nodes, H=number of hierarchy levels, k=number of clusters

Table 2: Comparison Table for Computational Complexity Analysis

5 Experimental Results

This section presents a performance comparison of the proposed algorithm ”Residual Energy-Aware Fuzzy-Based Clustering Algorithm (REAFCA)” with K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER clustering algorithms. The performance comparison are carried out to calculate the average throughput, energy consumption and average delay based on the number of nodes while assuming a fixed network topology involving all types of channel and features. All the experiments are done using Aqua-Sim a NS-2 based simulation tool.

5.1 Performance Metrics

To perform the performance comparison the following performance metrics are considered.

– *Average Throughput (THP)*: Throughput is defined as the aggregate number of received packets (TRcp) measured in bytes, divided by the time taken to transmit each packet (TEp) between parties. The results are measured in kbps, obtained by multiplying the outcome by 8 and then dividing by 1000. The formula for throughput is given in equation 18:

$$\text{THP[kbps]} = \frac{\text{TRcp} \times 8}{\text{TEp} \times 1000} \tag{18}$$

– *Energy Consumption (EC)*: EC is the total amount of energy consumed by all the nodes which also includes transmitted power (T_{xp}), receiving power (R_{xp}), and idle power (Idlp) to transmit packets between nodes which is shown in equation 19. The measuring unit is in Joules (J). The formula for energy efficiency is given below:

$$EC\text{[Joules]} = \sum_{i=1}^{n} (T_{xp} + R_{xp} + Idlp) \tag{19}$$

Where, n = Nodes of nodes in total.

– *Average Delay* : It represents the overall time taken by each packet from source to destination, including processing delay, divided by the total number of received packets. It is measured in milliseconds (ms) and can be computed using the following formula which is shown in equation 20 :

$$AverageDelay\text{[ms]} = \frac{\sum_{i=1}^{TS}(RT_i - ST_i) + Pd}{TR} \tag{20}$$

Where, TS = total number of sent packets (TS), TR = total number of received packets (TR), RTi = received time of ith packet, STi = sending time of ith packet, Pd = processing delay.

5.2 Simulation Setup

To compare the performance of REAFCA with K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER, a network topology as shown in Figure 2 is considered.

The topology contains a maximum of 70 nodes where cluster formation and cluster head selection will take place. Cluster heads are used as sink nodes that collects all the data and send that data to base station. Hop-to-Hop routing is used for data transmission in this network topology. The parameter and library files taken from AquaSim simulator used for this simulation are given in Table **??**. The parameters and library files are explained as follows.

- Channel Model: The underwater communication channel is modeled as Channel/UnderwaterChannel.
- Propagation Model: The propagation of signals in underwater follows the Propagation/UnderwaterPropagation model.
- Network Interface: Nodes use Phy/UnderwaterPhy as the physical layer for underwater communication.
- MAC Protocol: Mac/UnderwaterMac/BroadcastMac is used as the MAC layer protocol, supporting broadcast communication.
- Routing Protocol: The Vector-based Forwarding (VBF) protocol is used as the ad hoc routing protocol. The VBF protocol operates in a hop-by-hop manner.
- Queue Management: The network uses Queue/DropTail/PriQueue for packet queuing with a maximum length of 50 packets (ifqlen).
- Link Layer: Link Layer (LL) settings are configured with a minimum delay of 50us, a delay of 25us, and unspecified bandwidth.
- Energy Model: Each node is equipped with an EnergyModel that defines the initial energy as 10000 units. The transmission power (txPower) is set to 1.0 units. The reception power (rxPower) is set to 0.2 units. The idle power consumption is 0.0001 units.
- Antenna Model: Nodes use Antenna/OmniAntenna with specific settings (Gt=1.0, Gr=1.0).

Parameters	**Values**
Channel	UnderwaterChannel
Radio propagation model	UnderwaterPropagation
MAC Protocol	MAC/BroadcastMac
Routing protocol	VBF
Initial energy (Joule)	10000
Transition time (in sec)	0.015
Simulation time (in sec)	1000
Traffic rate (in kbps)	0.05
Type of Traffic	CBR
Packet size (in byte)	60

Table 3: Simulation Parameters

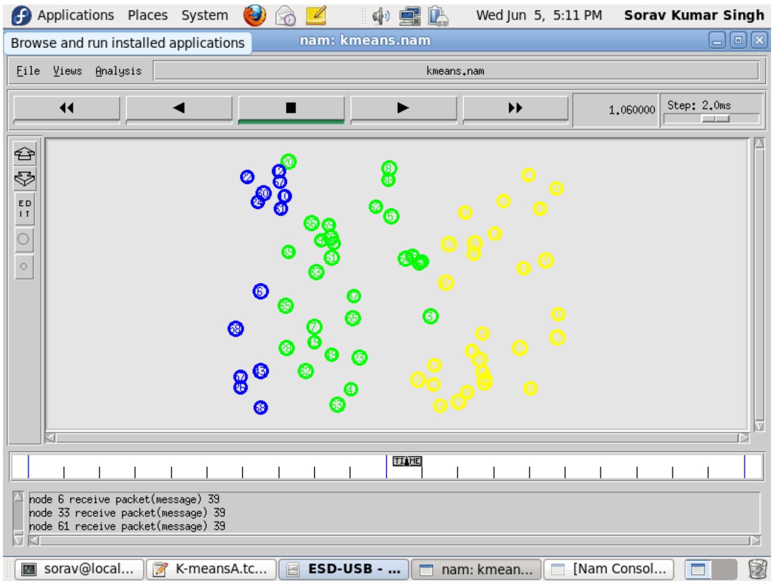

Figure 5: Visualization of K-means algorithm only forming cluster

5.3 Simulation Outcome

This section represents the simulation comparison of the proposed algorithm REAFCA with K-means and K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER. The simulation comparison is done for underwater sensor nodes ranging from 20 nodes to 70 nodes. Network animator (.nam) files and trace files (.tr) are generated for the seven clustering algorithm using the Aqua-Sim simulator. The Nam files are used for the visualization of the network. Figure 5 represents the visualization of the network using K-means algorithm where it only forms cluster and does not select cluster heads. Whereas, Figure 6 represents how K-meansA forms cluster, and Figure 6 shows how cluster heads are selected. Moreover, Figure 8 represents the visualization of the network using REAFCA where nodes in black color are cluster heads and three clusters are formed. Similar results have been obtained for the visualization of the networks for LEACH, PEGASIS, HEED, DB-SCAN and HEER. The generated trace files for K-means, K-meansA, REAFCA, LEACH, PEGASIS, HEED, DB-SCAN and HEER are used to analyse the performance of the network as discussed in the following subsections.

5.4 Result Discussion

This section gives a graphical comparison of REAFCA with K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER comparing their throughput, energy consumption delay as shown below.

5.4.1 Throughput with respect to Number of Nodes

The simulation in this section compares the average throughput of the proposed algorithm REAFCA with K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER

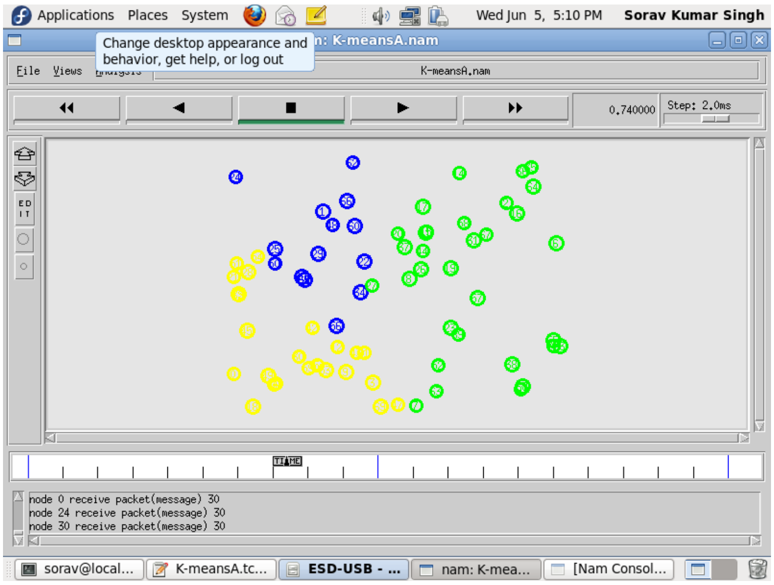

Figure 6: Visualization of K-meansA algorithm for cluster formation

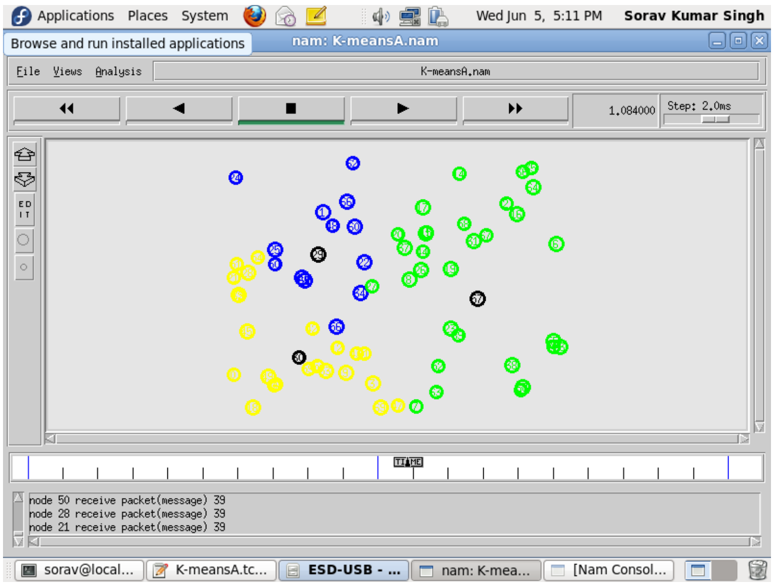

Figure 7: Visualization of K-meansA algorithm for cluster head selection

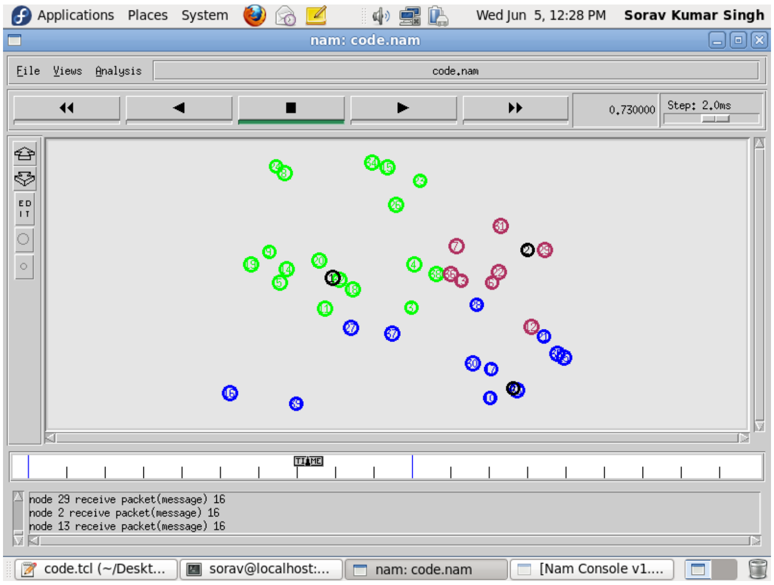

Figure 8: Visualization of REAFCA forming cluster and cluster head selection

with respect to the number of nodes where nodes vary from 20 to 70 nodes which is shown in Table **??** and Figure 9. From the figure, it can be observed that throughput varies from node to node with increasing nodes the throughput also increases.

REAFCA always shows maximum throughput because it selects the cluster head (CH) based on energy-aware mechanism and has the efficient clustering method. For example, at 70 nodes, REAFCA shows the highest throughput of 236.46 kbps, which exceeds HEED of 230 kbps and HEER of 200 kbps. K-means and K-meansA show moderate throughputs of 193.58 kbps and 213.37 kbps, respectively; LEACH and PEGASIS have much lower values in terms of their throughput due to frequent re-clustering and overhead of chain-based communication. DBSCAN performs well at smaller network sizes but struggles as the network grows. These results validate REAFCA's effectiveness in maintaining high data delivery rates in UWSNs.

5.4.2 Energy Consumption with respect to Number of Nodes

The simulation in this section compares the energy consumption of the proposed algorithm REAFCA with K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER concerning the number of nodes where nodes vary from 20 to 70 nodes which is shown in Table **??** and Figure 10. This simulation study compares the energy consumption of the proposed REAFCA algorithm against K-means, K-meansA, LEACH, PEGASIS, HEED, DBSCAN, and HEER over different network sizes. REAFCA shows high energy efficiency since it has a well-optimized cluster head selection and energy-aware clustering approach. For example, at 70 nodes, the energy consumption by REAFCA was 433.56 J while that of K-means is 558.63 J, K-meansA is 521.27 J, LEACH is 540 J, and PEGASIS is 505 J. HEED and HEER also perform well with 445 J and 520 J, respectively, while the highest energy consumption is recorded by DBSCAN because of its density-based

Number of Nodes	K-means (J)	K-meansA (J)	REAFCA (J)	LEACH (J)	PEGASIS (J)	HEED (J)	DB-SCAN (J)	HEER (J)
20	4.89	4.14	5.36	4.7	4.6	5	4.9	5
30	15.6943	18.7214	25.714	14.5	16	24	15.8	22
40	36.6743	33.5127	43.7845	31.5	34	42	36	40
50	74.668	84.2646	77.441	70	75	76	74.5	73
60	119.582	117.208	128.157	110	115	125	118	120
70	193.582	213.373	236.461	190	200	225	192	200

Table 4: Average Throughput of Clustering Algorithms

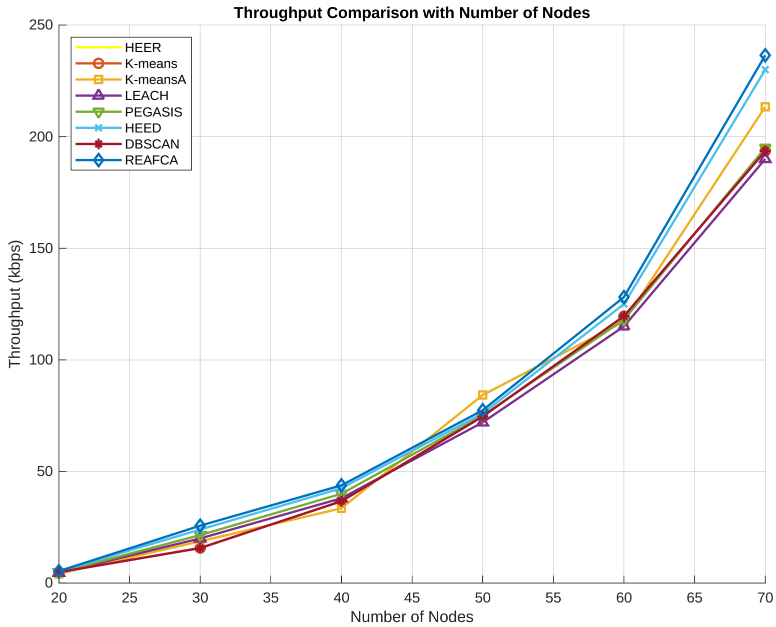

Figure 9: Throughput with respect to Number of Nodes

operations. The results are validated in REAFCA by minimizing energy consumption, thus elongating the network lifetime and performing better than the traditional algorithms for energy-constrained UWSNs.

5.4.3 Average Delay with respect to Number of Nodes

This section compares the average delay of the proposed algorithm REAFCA with K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER concerning the number of nodes where nodes vary from 20 to 70 nodes so as to prove the efficacy for delay-critical applications for UWSNs. Figure 11 and Table 6 represents the comparisons of average delay with respect to number of nodes. As discussed, on the basis of a highly

Number of Nodes	K-means (J)	K-meansA (J)	REAFCA (J)	LEACH (J)	PEGASIS (J)	HEED (J)	DB-SCAN (J)	HEER (J)
20	36.0053	36.1053	31.2388	33	32.5	31.8	36.0053	32.5
30	87.5537	49.8917	29.5918	45	40	30.5	49.8917	38
40	209.895	175.847	163.097	185	170	165	209.895	170
50	273.815	243.573	231.273	260	245	235	273.815	240
60	395.58	371.451	353.26	380	365	355	395.58	370
70	558.637	521.272	433.561	540	505	445	558.637	520

Table 5: Energy Consumption of Clustering Algorithms

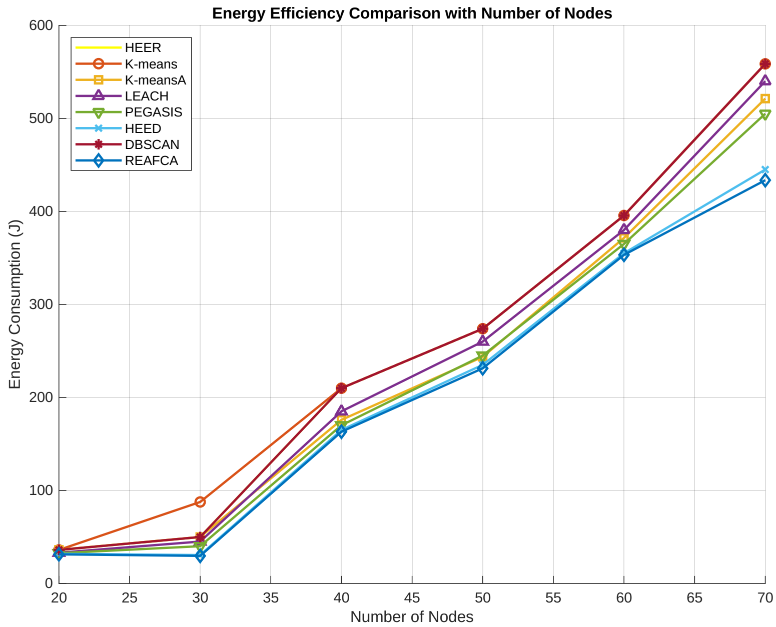

Figure 10: Energy Consumption with respect to Number of Nodes

effective clustering strategy along with energy-aware cluster head selection along with optimized data transmission, it performs consistently lower compared to all. For 70 nodes, REAFCA records an average delay of 1.037 seconds, compared to HEED at 1.0378, HEER at 1.038, and K-meansA at 1.0383 seconds. Comparing with K-means and DBSCAN that were left behind at 1.0416 seconds, it shows that REAFCA outshines both, giving the lowest communication overhead and low retransmissions due to underwater noise and interference in the environment. LEACH also suffers from comparatively higher delays where its delay would be 1.038 second and PEGASIS by 1.039 seconds are recorded due to its frequent time re-clustering and sequential ordering of data in the forwarding, but REAFCA dynamically makes the adaptation at the environmental circumstances of node's mobility and difference in propagation times and ensures swift and reliable transfers for real time underwater monitoring applications.

Number of Nodes	K-means (J)	K-meansA (J)	REAFCA (J)	LEACH (J)	PEGASIS (J)	HEED (J)	DBSCAN (J)	HEER (J)
20	1.00361	1.01304	1.00642	1.0075	1.008	1.0068	1.0036	1.006
30	1.02959	1.02956	1.02956	1.0296	1.0297	1.0295	1.0295	1.029
40	1.06377	1.06376	1.06377	1.0638	1.0638	1.0637	1.0637	1.064
50	1.06256	1.06376	1.06375	1.0638	1.0638	1.0637	1.0625	1.064
60	1.06396	1.06427	1.06026	1.061	1.062	1.0605	1.0639	1.062
70	1.04160	1.03830	1.03728	1.038	1.039	1.0378	1.0416	1.038

Table 6: Average Delay of Clustering Algorithms

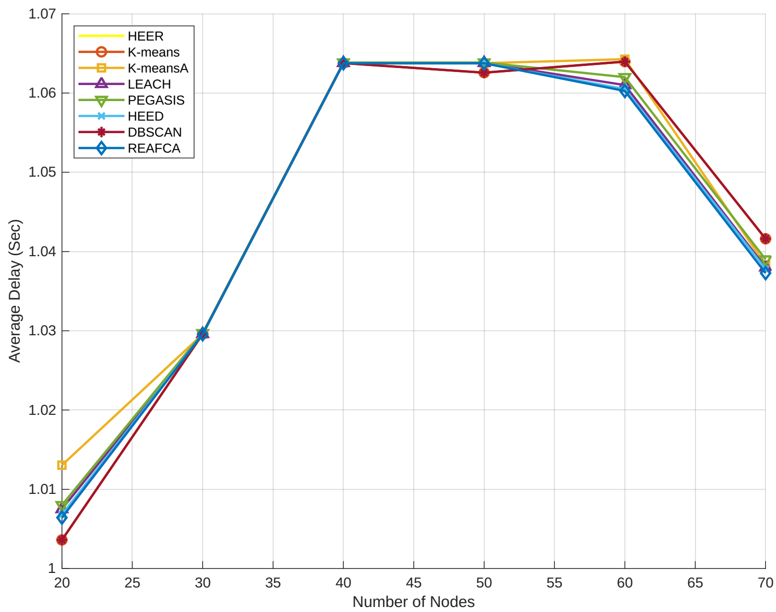

Figure 11: Average Delay with respect to Number of Nodes

5.5 Sensitivity Analysis

In order to evaluate the sensitivity of environmental factors in the formation of clusters and energy consumption in REAFCA, a sensitivity analysis is carried out. In this regard, stability of cluster formation and comparison of the performance of REAFCA with other algorithms, namely, K-means, K-meansA, LEACH, PEGASIS, HEED, DBSCAN, and HEER, is evaluated by Silhouette Coefficient.

5.5.1 Cluster Formation Stability

Cluster formation stability is measured using cluster head variability, cluster lifetime, and silhouette coefficient of clustering algorithms. Cluster head variability is measured by how frequently the cluster head changes across rounds. The unit of cluster head variability is %. Cluster lifetime is measured as the duration a cluster remain stable before reclustering is required. The unit for cluster life time is round. Table 7 represents the cluster formation stability with cluster head variability and cluster lifetime. The cluster formation stability analysis reveals that the significant advantage of REAFCA is in the formation of clusters, with a minimum variability in the cluster head at 5%, thus exhibiting high stability with minimal overhead for re-clustering due to fitness-based selection of CH and its energy-awareness. K-means shows poor stability without energy-aware mechanisms, and K-meansA shows some improvement by considering distance metrics. LEACH exhibits frequent re-clustering and CH variability at 25% and hence decreases the lifetime of the cluster. While forming stable chains, PEGASIS is challenged in dynamic environments, and HEED has an average balance between moderate stability and lifetime. DBSCAN shows good stability but faces problems in sparse networks. HEER equals the performance of REAFCA by exploiting hierarchical and energy-aware strategies.

	K-means	**K-meansA**	**REAFCA**	**LEACH**	**PEGASIS**	**HEED**	**DBSCAN**	**HEER**
Cluster head variation (%)	20%	15%	5%	25%	18%	10%	12%	8%
Cluster life-time (round)	5	8	15	6	7	10	9	12

Table 7: Cluster Formation Stability of Clustering Algorithms

Cluster formation stability with silhouette coefficient of clustering algorithms is presented in the next section.

5.5.2 Cluster Evaluation using Silhouette Coefficient

In addition to the throughput, energy efficiency and delay performance-based metrics, Silhouette Coefficient is another standard technique used for evaluating the quality of cluster formations. It measures the goodness of well-separated and cohesive clusters. Hence, it forms a robust approach for evaluating the performance of the REAFCA's clustering approach against other algorithms like K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER.

Silhouette Coefficient for a node is calculated by considering its cohesion and separation. Cohesion is the average distance between a node and all other nodes in its own cluster, whereas separation is the average distance between a node and all nodes in the nearest cluster to which it does not belong.

Let, i be a node whose cohesion, separation, and silhouette coefficient are defined as a(I), b(I), and S(I) respectively. Therefore, S(i) can be calculated as:

$$S(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{21}$$

The silhouette score of a sensor node i, ranges from -1 to +1, -1 means poorly clustered, 0 means overlapped cluster, and +1 means perfectly clustered. The overall Silhouette Coefficient of an algorithm is the average of S(i) for all the sensor nodes in the network. The silhouette coefficient of all clustering algorithms with their average values is shown in Table 8. From the table, it has been observed that the silhouette coefficient of REAFCA is 0.78 which is higher than the rest of the clustering algorithms, viz., K-means, K-meansA, LEACH, PEGASIS, HEED, DB-SCAN and HEER.

K-means	**K-meansA**	**REAFCA**	**LEACH**	**PEGASIS**	**HEED**	**DB-SCAN**	**HEER**
0.6	0.65	0.78	0.6	0.68	0.7	0.71	0.73

Table 8: Silhouette Coefficient of Clustering Algorithms

Average energy usage for REAFCA with other protocols under varying network sizes is presented in section 5.4.2. Node mobility, network density, and time-varying communication ranges hugely affect energy use. REAFCA's adaptation to such conditions also includes dynamic adaptability in clustering mechanisms through its energy-aware technique in cluster head selection and avoiding overhead in the communication process. For example, in high network densities or increased node mobility, REAFCA balances energy distribution by taking into account residual energy and proximity metrics, thereby reducing energy consumption compared to traditional algorithms. This adaptability makes REAFCA more resilient and efficient in energy-constrained UWSNs.

6 Conclusions and Future Research Direction

The paper presents a Residual Energy-Aware Fuzzy-Based Clustering Algorithm (REAFCA) for Underwater Wireless Sensor Networks (UWSNs) which helps to improve the network lifetime and data transmission with energy-efficiency. By applying a fuzzy logic-based clustering approach, REAFCA adapts dynamically to environmental conditions to optimize the selection of cluster haed and to ensure a balance in the energy consumption across nodes. This algorithm uses angular velocity, residual energy, threshold, rank, and node radius to create an energy-efficient cluster. It first forms clusters and then starts selecting probable cluster heads and then based on fitness it chooses a candidate cluster head and based on that algorithm chooses the cluster heads. After that, it performs intra-cluster communication and cluster finalization. This algorithm also perform better at intra-cluster communications and broadcast every time a sensor node is elected as a cluster head. Moreover, the paper presents simulation studies of REAFCA with K-means and K-meansA LEACH, PEGASIS, HEED, DB-SCAN and HEER for average throughput, energy consumption and average delay.

Simulation results showed that REAFCA outperforms the traditional algorithms, such as K-means, K-meansA, LEACH, PEGASIS, HEED, DBSCAN, and HEER, in terms of energy efficiency and throughput while keeping competitive average delay performance. These results confirm the success of REAFCA in extending the lifetime and reliability of UWSNs.

This work has opened up new avenues for future research. Some of the potential future studies include the following: i) extension of REAFCA to support underwater environments with high mobility and dynamically changing topology and mobile nodes, ii) hybrid acoustic-optical communication techniques for reliable data transmission, iii) incorporation of machine learning algorithm with the REAFCA algorithm to dynamically optimize different parameters using previous data is possible, v) extension of the algorithm to optimize further parameters, including security and fault tolerance simultaneously, and v) testing the scalability of REAFCA in denser and larger networks.

References

[Ali, 2023] Alı-Gburyı, K., Shah, A. S. (2023). Performance Comparison of PEGASIS, HEED and LEACH Protocols in Wireless Sensor Networks. Celal Bayar University Journal of Science, 19(1), 11-18.

[Bhatti, 2016] Bhatti, D. M. S. , Saeed, N. , and Nam, H. . Fuzzy c-means clustering and energy efficient cluster head selection for cooperative sensor network. Sensors, 16(9):1459, 2016.

[Davis, 2012] Davis, A. and Chang, H. . Underwater wireless sensor networks. In 2012 Oceans, pages 1–5. IEEE, 2012.

[Ester, 1996] Ester, M., Kriegel, H. P., Sander, J., Xu, X. (1996, August). Density-based spatial clustering of applications with noise. In Int. Conf. knowledge discovery and data mining (Vol. 240, No. 6).

[Fei, 2020] Fei, W. , Hexiang, B. , Deyu, L. , and Jianjun, W. . Energy-efficient clustering algorithm in underwater sensor networks based on fuzzy c means and mothflame optimization method. IEEE Access, 8:97474–97484, 2020.

[Felemban, 2015] Felemban, E. , Shaikh, F. K. , Qureshi, U. M. , Sheikh, A. A. , and Qaisar, S. B. . Underwater sensor network applications: A comprehensive survey. International Journal of Distributed Sensor Networks, 11(11):896832, 2015.

[Gkikopouli, 2012] Gkikopouli, A. , Nikolakopoulos, G. , and Manesis, S. . A survey on underwater wireless sensor networks and applications. In 2012 20th Mediterranean conference on control automation (MED), pages 1147–1154. IEEE, 2012.

[Gu, 2024] Gu, L., Mohajer, A. (2024). Joint throughput maximization, interference cancellation, and power efficiency for multi-IRS-empowered UAV communications. Signal, Image and Video Processing, 18(5), 4029-4043.

[He, 2024] He, S., Li, Q., Khishe, M., Salih Mohammed, A., Mohammadi, H., Mohammadi, M. (2024). The optimization of nodes clustering and multi-hop routing protocol using hierarchical chimp optimization for sustainable energy efficient underwater wireless sensor networks. Wireless networks, 30(1), 233-252.

[Heidemann, 2012] Heidemann, J. , Stojanovic, M. , and Zorzi, M. . Underwater sensor networks: applications, advances and challenges. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 370(1958):158175, 2012.

[Heinzelman, 2000] Heinzelman, W. R., Chandrakasan, A., Balakrishnan, H. (2000, January). Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd annual Hawaii international conference on system sciences (pp. 10-pp). IEEE.

[Khan, 2021] Khan, M. F. , Bibi, M. , Aadil, F. , and Lee, J.-W. . Adaptive node clustering for underwater sensor networks. Sensors, 21(13):4514, 2021.

[Krishnaswamy, 2021] Krishnaswamy, Vani, and Sunilkumar S. Manvi. "Trusted node selection in clusters for underwater wireless acoustic sensor networks using fuzzy logic." Physical Communication 47 (2021): 101388.

[Li, 2007] Li, X. , Fang, S.-l. , and Zhang, Y.-c. . The study on clustering algorithm of the underwater acoustic sensor networks. In 2007 14th International conference on mechatronics and machine vision in practice, pages 78–81. IEEE, 2007.

[Lindsey, 2002] Lindsey, S., Raghavendra, C. S. (2002, March). PEGASIS: Power-efficient gathering in sensor information systems. In Proceedings, IEEE aerospace conference (Vol. 3, pp. 3-3). IEEE.

[Luo, 2018] Luo, H. , Wu, K. , Ruby, R. , Liang, Y. , Guo, Z. , and Ni, L. M. . Software defined architectures and technologies for underwater wireless sensor networks: A survey. IEEE Communications Surveys Tutorials, 20(4):28552888, 2018.

[Luo, J., 2020] Luo, J. , Yang, Y. , Wang, Z. , Chen, Y. , and Wu, M. . A mobility-assisted localization algorithm for three-dimensional large-scale uwsns. Sensors, 2020. (15):4293, 2020.

[Nigam, 2017] Nigam, G. K., Dabas, C. (2017). Performance analysis of heed over leach and pegasis in wireless sensor networks. In Transactions on Engineering Technologies: World Congress on Engineering and Computer Science 2015 (pp. 259-266). Springer Singapore.

[Panchal, 2023] Panchal, H., Gajjar, S. (2023). Fuzzy□based unequal clustering protocol for underwater wireless sensor networks. International Journal of Communication Systems, 36(16), e5581.

[Priyanga, 2020] Priyanga, M., S. Leones Sherwin Vimalraj, and J. Lydia. "Energy efficient distributed unequal clustering algorithm with relay node selection for underwater wireless sensor networks." Emerging Trends in Computing and Expert Technology. Springer International Publishing, 2020.

[Roy, 2010] Roy, A., Sarma, N. (2020). RPCP MAC: Receiver preambling with channel polling MAC protocol for underwater wireless sensor networks. International Journal of Communication Systems, 33(9), e4383.

[Roy, 2011] Roy, A., Sarma, N. (2021). A synchronous duty-cycled reservation based MAC protocol for underwater wireless sensor networks. Digital Communications and Networks, 7(3), 385-398.

[Subramani, 2022] Subramani, N. , Mohan, P. , Alotaibi, Y. , Alghamdi, S. , and Khalaf, O. I.. An efficient metaheuristic-based clustering with routing protocol for underwater wireless sensor networks. Sensors, 22(2):415, 2022.

[Tran, 2014] Tran, K. T.-M. and Oh, S.-H. . Uwsns: A round-based clustering scheme for data redundancy resolve. International Journal of Distributed Sensor Networks, 10(4):383912, 2014.

[Tiwari, 2023] Kumar Tiwari K, Singh S. Energy□optimized cluster head selection based on enhanced remora optimization algorithm in underwater wireless sensor network. International Journal of Communication Systems. 2023 Oct;36(15):e5560.

[Wang, 2024] Wang, Q., Li, W., Mohajer, A. (2024). Load-aware continuous-time optimization for multi-agent systems: Toward dynamic resource allocation and real-time adaptability. Computer Networks, 250, 110526.

[Xie, 2024] Xie, W., Shen, X., Wang, C., Sun, L., Yan, Y. and Wang, H., (2024). Adaptive Energy-Efficient Clustering Mechanism for Underwater Wireless Sensor Networks Based on Multi-Dimensional Game Theory. IEEE Sensors Journal.

[Yang, 2024] Yang, T., Sun, J., Mohajer, A. (2024). Queue stability and dynamic throughput maximization in multi-agent heterogeneous wireless networks. Wireless Networks, 1-27.

[Yi, 2016] Yi, D., Yang, H. (2016). HEER–A delay-aware and energy-efficient routing protocol for wireless sensor networks. Computer Networks, 104, 155-173.

[Younis, 2004] Younis, O., Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on mobile computing, 3(4), 366-379.

[Zeng, 2014] Zeng, D. , Wu, X. , Wang, Y. , Chen, H. , Liang, K. , and Shu, L. . A survey on sensor deployment in underwater sensor networks. In Advances in Wireless Sensor Networks: 7th China Conference, CWSN 2013, Qingdao, China, October 17-19, 2013. Revised Selected Papers 7, pages 133–143. Springer, 2014.