

# Optimizing Data Survivability in Unattended Wireless Sensor Networks: A Machine Learning Approach to Cluster Head Selection and Hybrid Homomorphic Encryption

Haritha K Sivaraman<sup>1,\*</sup>, <sup>2</sup>Rangaiah L<sup>2</sup>

<sup>1,\*</sup>Research Scholar, Department of Electronics & Communication Engineering, VTU, Belagavi, Raja Rajeswari College of Engineering, Bangalore, India.

<sup>2</sup>Professor, Department of Electronics & Communication Engineering, Raja Rajeswari College of Engineering, Bangalore, India.

---

## Article Info

### Article history:

Received Oct 25, 2024

Revised Feb 1, 2025

Accepted Feb 26, 2025

---

### Keyword:

UWSN;  
Data Survivability;  
Deep Q-Networks;  
optimized hybrid homomorphic encryption;  
Seagull optimization  
Algorithm;  
Whale Optimization Algorithm.

---

## ABSTRACT

The research relies on machine learning-based Cluster Head (CH) selection and optimised Attribute-Based Encryption (ABE) with Homomorphic Encryption to improve data survivability in Unattended Wireless Sensor Networks (UWSNs). Integrating blockchain technology would enable tamper-proof data storage and provenance. The suggested method uses machine learning techniques like Deep Q-Networks (DQNs) or other models for intelligent and adaptive CH selection in UWSNs. Dynamically selecting CHs takes into account energy efficiency, network coverage, communication dependability, and node characteristics. The second part protects data using optimised Attribute-Based Encryption (ABE) and Homomorphic Encryption. ABE offers fine-grained attribute-based access control to restrict data access to authorised entities. Secure processing of encrypted data using homomorphic encryption protects privacy and integrity. These encryption algorithms are optimised to balance security and computational performance for efficient data processing and transmission while guaranteeing data privacy and integrity. Blockchain technology is suggested for tamper-proof data storage and provenance. To optimise the suggested solution's performance, the study uses the Seagull Optimisation Algorithm (SOA) and the Whale Optimisation Algorithm (WOA). These algorithms fine-tune system parameters, optimise CH selection, and boost UWSN performance. This holistic strategy uses machine learning-based CH selection, optimised ABE with Homomorphic Encryption, and blockchain technology for tamper-proof data storage and provenance to improve UWSN data survival. Optimisation algorithms boost the solution's efficacy and efficiency, protecting UWSN data, latency, and energy usage.

Copyright © 2025 Institute of Advanced Engineering and Science.  
All rights reserved.

---

## Corresponding Author:

Haritha K Sivaraman,  
Department of Electronics & Communication Engineering, VTU, Belagavi, Raja Rajeswari College of Engineering, Bangalore, India.  
Email: haritharesearchscholar@gmail.com

---

## 1. INTRODUCTION

Wireless Sensor Networks (WSNs) have become the most emerging technology in the field of research for the advanced development of digital networks in recent years. WSN constitutes an extensive collection of sensors, forming a distributed network for sensing, self-organization, and data propagation [1]. In a distributed environment, WSN nodes serve as compact, self-contained devices with limited resources. They play a crucial role in processing information, communication, and sensing mechanisms, enabling the detection of environmental conditions in their immediate surroundings. These networks rely on batteries as their energy source [2]. The effectiveness of sensor nodes is limited by several factors, including storage

capacity, processing speed, battery life, and more [3]. As a result, ensuring adequate security provisions [4] becomes an undertaking challenge. In the WSN, the sensors generate data and operate in a multi-hop fashion, relaying it from one node to another. Their primary objective is to efficiently gather a relevant set of information and relay it to the Base Station (BS) [5]. These types of sensors find application in vast fields, including industry, army operations, residential settings, and scientific endeavors. They serve various purposes, such as healthcare, freight services, combat field, catastrophic recovery, construction industrialization, security, astronautics, and industrial sectors [6]. The majority of existing WSN systems consist of a considerable quantity of sensor nodes which are small in size, cost-effective, and limited in energy resources while possessing sensing capabilities. These sensor nodes are susceptible to damage caused by energy depletion or natural calamities. As sensor nodes within a WSN are constructed in a multi-hop fashion to keep track of natural surroundings and the malfunction of certain nodes can lead to a decline in network coverage and connectivity, potentially rendering the entire network ineffective [7] and [8]. Unattended Wireless Sensor Networks (UWSNs) refer to a specific category of Wireless Sensor Networks characterized by the sporadic presence of a sink node. As a result, enhancing the robustness of the network has been a significant focus of research in the realm of UWSNs. Typically, sensor nodes are randomly distributed across the monitored area, continuously collecting sensor data.

The UWSN is responsible for accomplishing its mission promptly, even when faced with challenges such as intrusions, attacks, accidents, and failures within a hostile environment. The challenge at hand is commonly referred to as the survivability problem. The survivability of a system can be defined as its capacity to successfully achieve its mission within the designated timeframe, even when confronted with intrusions, attacks, accidents, and failures [9]. [10] introduced a mathematically constrained definition of survivability, which establishes a framework for quantifying the level of network survivability. In this research article, we proceed with the analysis of the data survivability of a UWSN. Drawing inspiration from [11], the concept of UWSN survivability is defined based on the frequency and impact of failures. Distribution of UWSNs is typically focused on a particular field and individual user, which significantly restricts the flexibility and scalability of task execution. Consequently, the resource utilization of sensor nodes remains below 20 percent, rendering them incapable of meeting the diverse Standards about quality of service (QoS) associated with various applications. Conventional software embedding methods, which involve incorporating new application components into the existing network architecture, often result in a cumbersome and inflexible network structure. However, the utilization of network virtualization technology [12, 13] offers an effective solution to this issue. It not only resolves the aforementioned problem but also presents substantial economic advantages, including minimal installation and operational expenditures associated with resource leasehold to external clients. Additionally, virtualization of network technology enables swift configuration and recovery of physical network resources in the event of failures, while also providing the ability to scale resources on demand.

The energy constraints faced by sensors in UWSNs also pose a significant challenge, leading to various failures and issues regarding survivability. To conserve energy, a commonly employed technique involves transitioning the nodes between sleep and awake states. Typically, the network is designed such that multiple nodes are deployed to ensure coverage of each region. In such scenarios, it is feasible to maintain region coverage by keeping one node awake while the remaining nodes are in sleep mode. During sleep mode, the sensors cease radio broadcasts and environmental sensing activities. In a UWSN, radio transmission is recognized as the primary power-consuming function, consuming a significant amount of power. As the intensity of an awake mode sensor is fully drained, one of the sensors in the sleep mode can be activated to assume the role of coverage provision in that particular area. The pioneering work of [14] introduced the utilization of the dominant set of algorithms to regulate the Nodes' sleep and wake intervals within a UWSN. Building upon this, [15] introduced the concept of generating a maximum number of disjoint dominating sets, known as the domatic partition problem, in unit disk graphs. Domatic partition involves dividing the sensor nodes into clusters or groups in such a way that the energy consumption is balanced across the network. By achieving a balanced energy consumption, the authors aim to extend the overall lifetime of the network. Over time, advancements in UWSN technology have led to the widespread adoption and practicality of rechargeable nodes. In this research paper, we primarily focus on data survivability in UWSNs. Here are the key **unique contributions** that differentiate the existing methods from our proposed work,

#### **Contributions of the Proposed Approach:**

##### **1. Intelligent & Adaptive CH Selection Using ML**

- Unlike traditional heuristic-based CH selection, the proposed approach leverages Deep Q-Networks (DQNs) or other ML models to dynamically select Cluster Heads (CHs) based on real-time network conditions, energy efficiency, and security factors.

- This adaptive learning mechanism ensures resilience against Sybil attacks, blackhole attacks, and inefficient CH rotation, which are common vulnerabilities in UWSNs.
2. Optimized Hybrid Encryption Mechanism (ABE + Homomorphic Encryption)
    - Most existing approaches either use Attribute-Based Encryption (ABE) for access control or Homomorphic Encryption (HE) for privacy-preserving computations.
    - The proposed hybrid encryption mechanism is optimized to balance security and computational efficiency, ensuring:
      - Fine-grained access control (via ABE) to prevent unauthorized access.
      - Secure computations on encrypted data (via Homomorphic Encryption) without decryption, enhancing data privacy.
    - The encryption scheme is tailored for resource-constrained UWSNs, minimizing energy consumption while maintaining strong security guarantees.
  3. Blockchain Integration for Tamper-Proof Data Storage & Provenance
    - Unlike conventional methods that rely solely on encryption, this research integrates blockchain technology to achieve tamper-proof storage, auditability, and provenance tracking.
    - The use of blockchain:
      - Prevents unauthorized data modifications and ensures integrity.
      - Provides decentralized trust management without requiring a centralized authority.
      - Reduces the impact of Man-in-the-Middle (MitM) attacks and data injection attacks.
  4. Multi-Objective Optimization via SOA & WOA
    - Existing optimization methods for UWSNs often focus on single parameters (e.g., energy efficiency or latency).
    - The proposed approach employs Seagull Optimization Algorithm (SOA) and Whale Optimization Algorithm (WOA) for multi-objective optimization, fine-tuning:
      - CH selection efficiency
      - Encryption parameter optimization (balancing security vs. computational cost)
      - Blockchain transaction performance (minimizing delays and energy usage)
    - This optimization enhances overall system performance while reducing energy overhead, making the solution practical for real-world deployment.

## 2. LITERATURE REVIEW

Data survivability in UWSNs is closely linked to the frequent occurrence of failures, which can disrupt the network's ability to carry out its intended tasks, such as data aggregation. Several research papers in the past and recent literature have addressed the challenges and issues related to survivability analysis in networked systems. These papers have offered a comprehensive explanation of the concept of survivability and provided a theoretical framework that has inspired our focus on conducting a specific analysis that caters to the unique technical specifications of UWSNs. In 2020, Yang Z. *et al.* [16] addressed the issue of failure of a single link protection in hybrid IP/SDN (Software Defined Networking). In such networks, a combination of traditional IP-based routing and SDN-based control is utilized. The authors focus on selecting suitable SDN candidates, which are network nodes that can dynamically reconfigure the network after a link failure to ensure continued connectivity and efficient traffic routing. In 2010, Li, *et al.* [17] proposed a multi-path protocol that optimizes the selection of paths based on factors such as link quality, energy consumption, and congestion that used almost all the resources available in the network. The protocol aims to minimize data loss, reduce latency, and balance energy consumption across the network.

In 2019, Raj, Jennifer S. *et al.* [18] conducted a study utilizing fuzzy logic and Convolutional Neural Networks (CNN) to achieve this optimization. The paper focuses on enhancing QoS constraints: energy consumption, data delivery, and network lifetime of client-based sensor networks by applying fuzzy logic including CNN techniques. The approach aims to strike a balance between energy efficiency and QoS requirements, enabling efficient and reliable data transmission in IoT sensor networks. In 2016, Jammu, Srikanth, *et al.* [19] presented a grid-based approach to address the hot spot problem in wireless sensor networks. The hot spot problem refers to the imbalance of energy consumption and network load distribution that can occur in UWSNs, leading to premature battery drain of certain sensor nodes. The proposed algorithm utilizes a grid-based clustering and routing technique to evenly distribute the load and energy consumption across the network. By dividing the network into a grid structure and employing clustering and routing strategies, the algorithm aims to mitigate the hot spot problem and prolong the network's overall lifetime. In 2016, Smys, S., and Robert Bestak *et al.* [20] provided an introductory overview of a special issue in the journal *Wireless Personal Communications*. The objective of the special edition is cutting-edge network architectures for wireless personal computing systems of the future. The paper sets the context for the special edition and highlights the importance of developing novel network architectures to meet the evolving needs

of wireless personal systems. It serves as an introductory piece, providing an overview and setting the stage for the collection of articles in the special edition that investigate novel network architectures for wireless personal computing systems in the future.

In 2021, Qing Fan *et al.* [21] presented a scheme that utilizes blockchain technology to provide secure authentication and efficient data sharing in the context of the Internet of Things (IoT). The scheme aims to address the challenges of ensuring authentication and data integrity in IoT environments by leveraging the decentralized and tamper-resistant nature of blockchain technology. In 2019, Smys *et al.* [22] introduced a huge dataset of application-specific energy-conscious security routing protocols for UWSN. Its aims to balance energy efficiency and security requirements, addressing the challenges associated with resource constraints and the need for secure data transmission in UWSNs. In 2014, Wang *et al.* [23] implemented PWDGR, a Pair-Wise based Directional Geographical Routing algorithm for wireless sensor networks. By exploiting geographical information and employing directional routing, the algorithm aims to improve routing efficiency, reduce energy consumption, and enhance the reliability of data transmission in UWSNs. In 2019, Sivaganesan, D *et al.* [24] focused on the establishment of a collision-avoidance routing algorithm that is effective for automotive networks. Vehicular networks involve vehicles communicating with each other to exchange information and enable various services such as traffic management and safety applications. In 2018, Deng *et al.* [25] proposed the concept of survivability in wireless sensor networks that build upon the foundation of security, emphasizing that security serves as the fundamental concept for researching network survivability.

In the context of analyzing this data survivability of UWSN, the primary focus lies in analyzing the connectivity of communication links based on the network's topology. Selecting suitable indicator parameters is crucial when examining the data survivability aspect of these networks. Graph theory offers a range of indicators that describe graph connectivity, including but not limited to measures such as network connectivity, coritivity, and uniformity. By addressing these gaps, future research could significantly contribute to the advancement of data survivability and security in Unattended Wireless Sensor Networks.

- Explore advanced machine learning techniques for Cluster Head (CH) selection beyond Deep Q-Networks (DQNs).
- Assess the scalability of optimised Attribute-Based Encryption (ABE) and Homomorphic Encryption in large-scale UWSNs.
- Investigate the impact of environmental factors on UWSN performance and CH selection efficiency.
- Address challenges of implementing blockchain in resource-constrained UWSNs, including energy consumption and latency.
- Conduct empirical studies or real-world testing of the proposed methods in actual UWSN deployments.
- Explore user privacy concerns related to data access and sharing in sensitive applications like healthcare.
- Perform comparative analysis of the proposed solution against existing methods in UWSNs.

### 3. PROPOSED METHOD

#### 3.1. Clustering

The method we propose consists of two primary elements: clustering of sensor nodes and secure data transmission based on cryptographic algorithms. The objective of these components is to enhance the effectiveness, velocity, and security of Wireless Sensor Networks.

**Sensor Node Clustering:** In large-scale Unattended Wireless Sensor Networks (UWSNs), clustering plays a vital role in extending the network's lifespan. In this research, we propose an adaptive approach to selecting Cluster Heads (CHs) using a Reinforcement Learning algorithm, specifically Deep Q-Networks (DQNs). This approach incorporates dynamic clustering principles into the CH selection process. By considering the residual energy amplified by the occurrence region, an efficient election of the cluster head is achieved.

To address the complexity of this problem, we employ Reinforcement Learning (RL), an Artificial Intelligence (AI) technique. Within the RL framework, the master node, acting as an agent, incorporates a packet scheduler, while the remaining components of the UWSN represent the surrounding environment. The environment provides feedback in the manner of status updates and rewards as the master node executes a variety of scheduling operations over time. The DQN-based packet scheduler is guided by well-designed incentive functions as it learns across several episodes to converge on the desired configuration. To evaluate the effectiveness of the enhanced DQN scheduler, we ran tests in a variety of network scenarios. Numerous parameters, such as node population, mobility, how data arrives, and packet lifetime, have an impact on the scheduler's policy in Unattended Wireless Sensor Networks (UWSNs). We thoroughly experimented to examine the DQN scheduler's flexibility in dynamic network situations. We evaluated the DQN scheduler's

performance against that of other currently used techniques for longevity of the network and Quality of Service, or QoS, reliability. In addition, we sought to learn more about RL-based scheduling methods by probing the optimized DQN scheduler's policy. Through reverse engineering, these efforts not only improve our grasp of the underlying workings of solutions based on artificial intelligence but also give us useful information for designing non-AI strategies.

To reduce energy consumption and extend the network's lifespan, we integrate an advanced data aggregation technique called Distributed Compressive Sensing (DCS). Within each cluster, sensor nodes intelligently aggregate their data before transmitting it to the Cluster Head (CH), thereby reducing redundant information and conserving energy. DCS utilizes the principles of compressive sensing to efficiently collect and process sensor data. Unlike traditional data aggregation methods where each sensor node transmits its raw data to the CH, resulting in redundant and energy-intensive transmissions, DCS allows sensor nodes to collectively compress their data before transmission. This significantly reduces the volume of data sent to the CH.

**Cryptographic algorithm-based secured data transmission:** To tackle the challenge of ensuring data security during communication, we introduce a hybrid cryptographic solution. The sender encrypts its data using an innovative hybrid homomorphic encryption technique known as optimized Attribute-Based Encryption (ABE) with Homomorphic Encryption, which combines the Seagull Optimization Algorithm (SOA) and the Whale Optimization Algorithm (WOA). The computational complexity of this proposed approach relies on factors such as the size of the Unattended Wireless Sensor Network (UWSN), the number of sensor nodes, and the intricacy of the optimization strategy and cryptographic algorithm employed. Nonetheless, it can be implemented in real-time or near real-time, depending on the computational power and resources available within the UWSN. The utilization of sensor node clustering and optimal Cluster Head (CH) selection reduces communication overhead, leading to improved efficiency and speed in data transmission. Moreover, the hybrid cryptographic algorithm enhances both data transmission speed and the overall security of the UWSN. To ensure data storage integrity and provenance, we integrate blockchain technology into the UWSNs. Sensor data undergoes cryptographic hashing and is recorded on the blockchain, creating a transparent and immutable ledger that facilitates data verification. Overall, the proposed approach offers advancements in efficiency, speed, and security for UWSNs. It has the potential for real-time implementation, depending on the available computational resources. Refer to Figure 1. for a visual representation of this approach.

### 3.2. Network Model

The Unattended Wireless Sensor Network (UWSN) comprises  $N$  mobile sensors, which are uniformly distributed throughout the network. These sensors are programmed to perform periodic sensing and data collection tasks. The dynamic cluster formation model is adopted in which sensors move at speeds that enable them to reach any point within the deployment area within a single round. The passage of time is divided into rounds, during which sensors gather data and encrypt it using the public key of the trusted sink. The sink, assumed to be a trusted authority, visits the network once every  $\nu$  round to collect the data. After transferring the data to the sink, the sensor node's entire memory is promptly erased. It is assumed that the communication between sensors and between sensors and the sink is reliable. Additionally, each sensor possesses knowledge of its location as well as the locations of its immediate neighbors.

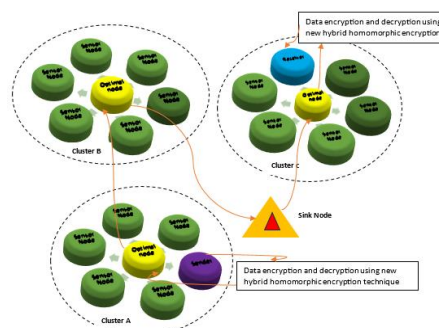


Figure 1. The architecture of the data survivability model in UWSN

To define the problem, we examine a UWSN model depicted in Figure 1, which comprises a sender node and multiple receiver nodes. Once a connection is established, the sender node and receiver nodes engage in communication within a predetermined connection interval (CI). The communication persists until

the battery of a receiver node is depleted. Consequently, we define the lifetime of the  $i$ th receiver, denoted as  $t^{(i)}$ , as the duration of its operational capability until the battery is completely drained.

$$t^{(i)} = \sum_{k=1}^N L_k, \text{ for } e_N^{(i)} < E_s^{(i)} < e_{N+1}^{(i)} \quad (1)$$

Here,  $L_k$  represents the duration of the  $k^{\text{th}}$  connection interval (CI),  $N$  indicates the directory of the final CI,  $e_k^{(i)}$  indicates the total amount of energy consumed by the  $i^{\text{th}}$  receiver node til the  $k^{\text{th}}$  CI, and  $E_s^{(i)}$  shows the battery's preliminary charge level of the  $i^{\text{th}}$  receiver. The duration required for each of the nodes to use up all of its battery power is the network lifetime in this study. Given that the sender consumes far more energy than the receiver, the receiver node essentially determines the network's overall lifetime. As a result, the network lifetime is solely determined by the receiver node that has the shortest operational duration.

Our objective is to optimize regulating the network lifetime  $(L_k, \{n_k^{(i)}\}_{i=1}^N)$  for  $k=1, 2, \dots$ , while simultaneously meeting the norms for Quality-of-Service  $q(i)$  of each receiver node, given by,

$$\begin{aligned} & \text{maximize} \\ & L_k, \{n_k^{(i)}\}_{i=1}^N = \min \{v(i)\}_{i=1}^N \\ & \text{subject to} \quad q(i) > \eta(i) \end{aligned} \quad (2)$$

In the above situation,  $t^{(i)}$  denotes the envisioned QoS criterion for the  $i^{\text{th}}$  recipient, whereas  $\eta^{(i)}$  indicates the QoS rating corresponding to packet transmission for that receiver.

### 3.3. Dynamic Cluster Formation

The dynamic generation and regulation of clusters are the main topics of this research. We particularly propose a dispersed, agent-oriented strategy for setting up tracking regions on the work floor. Every tracking zone consists of a group of Anchor Nodes (AN) that are close to the sink component and a sink node itself. When the environment changes, such as when sensor nodes are added or removed or when signals are blocked, the clusters adapt dynamically. Through inter-cluster trade-offs, the Unattended Wireless Sensor Network (UWSN) as a whole allows workload sharing. Our focus was on creating a dynamic and effective cluster formation technique to facilitate adaptive behaviors. To enable autonomous cluster creation and account for modifications to the production environment, such as sensor additions and removals, and signal loss, a clustering method is first required. Second, to reduce interaction distance and corresponding energy consumption, the resultant clusters needed to include AN that were situated close to their (sink). An anchor node sends a Call for Proposal (CFP) transmission to each of the nearby sink nodes that are within its radio frequency (RF) reach as soon as it is installed in the network field. In their bids, sink nodes provide distance values between themselves and the AN. The optimal bid, which matches the nearest sinks within all the received bids, is chosen by the AN after a predetermined amount of time has passed. According to formal rules, the AN selects the superlative nearest sink using the given process: The selection procedure can be expressed as follows if  $A_j$  signifies the position of anchoring  $j$  in  $R^n$ ,  $M_i$  indicates the origin of the cluster's mean  $i$ , and  $d$  denotes a distance parameter.

$$\arg \min_i d(A_j, M_i) \text{ for all anchor} \quad (3)$$

To put it simply, each AN selects the cluster in which the mean of the sink node is located the closest to it among all clusters. The distributed k-means algorithm is successfully implemented by this process. The contribution of this method is that the k-means's computing cost is decreased. The algorithm can be dispersed across the network, which allows the clustering issue to be addressed in an acceptable amount of time. Additionally, this strategy enables adaptable behavior within the system in addition to facilitating effective cluster formation. The Unattended Wireless Sensor Network (UWSN)'s cluster creation process can be started at any time, allowing for dynamic response to changes like the inclusion or exclusion of sensor nodes. The strategy also works well in crowded regions in which signal may be jammed by obstructions. In these situations, the bid procedure makes sure that the barred AN get allocated to the nearest CH that they can interact with other than just the closest that might be subjected to blocking.

### 3.4. Deep Q-Networks (DQN) Algorithm

By experimenting with multiple behaviors and viewing the rewards that ensue, Reinforcement Learning (RL) is one type of Machine Learning (ML) technique that helps an agent determine the best course

of action within a given environment. Figure 2 depicts the RL framework, which consists of an agent and its environment. The agent observes the condition of the environment throughout every stage and then acts in a way that affects the environment. The environment then changes into a unique state and gives the agent feedback in the manner of a reward. Without having any existing understanding of the environment, the agent learns a series of behaviors that optimizes the cumulative reward through numerous iterations of this procedure.

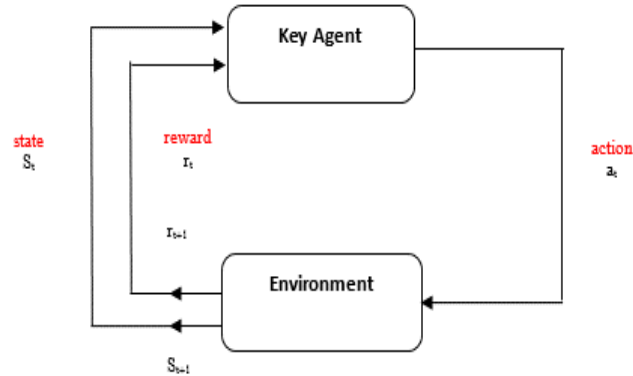


Figure 2. Within the framework of reinforcement learning, the dynamics occur through the interaction between a key agent and its surrounding environment

At each time step, denoted as  $t$ , the agent perceives and observes the current state of the environment, represented by  $s_t$ . Based on this observation, the agent selects and takes an action, denoted as  $a_t$ . Subsequently, the environment shifts to an entirely different state, denoted as  $s_{t+1}$ , and in response, provides the key agent with a reward, denoted as  $r_t$ . Among the various reinforcement learning algorithms, Q-learning is widely recognized as one of the most popular approaches. It aims to find an optimal policy for action selection by utilizing a Q-function, which estimates the importance of a particular action in a specific situation. The Q-function, denoted as  $Q(s, a)$ , undergoes updates according to the representation stated below,

$$Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \max_{a'} Q(s',a) - Q(s,a)] \quad (4)$$

In the formula,  $r$  stands for the compensation received as a result of action  $a$ ,  $\alpha$  stands for step size, for discount factor, and  $s'$  stands for the following state. The tabular update approach, when used to solve real-world issues, confronts difficulties such as an exponential increase in the state-action  $(s, a)$  space and sluggish convergence. A remedy known as Deep Q-Network (DQN) has been put forth as a response to these difficulties. To roughly represent the Q-function, DQN uses a neural network, making it possible to handle complicated issues more effectively. The DQN architecture also has a duplicated Q-network and experience replay, two crucial elements, to improve learning stability. Experience replay updates the Q-network using mini-batches, which enhances training stability. To further improve learning stability, the duplicated Q-network, which is derived from the primary Q-network, is used. In this arrangement, the main Q-network is trained using Q-values.

### 3.5. Distributed Compressive Sensing Technique

Compressive Sensing (CS) is a concept that originated in the field of signal processing. Its key strength is the capacity to reassemble scarce or compressed signals with just a few observations, without needing to know the signal's structure beforehand. When signals are sparse on a known basis, measures at the sensor end are expensive, and calculations at the receiving end are reasonably priced, compressive sensing is useful. Unattended Wireless Sensor Networks (UWSNs) specifications are perfectly met by these qualities.

Distributed Compressive Sensing (DCS) represents one of the best widely recognized ways among the different CS algorithms for correlated signals that have been suggested. To allow for the reconstruction of various signals captured by sensor nodes in a UWSN, DCS provides a joint signal restoration technique that utilizes a greedy algorithm. This approach presupposes that these signals follow established joint sparsity models.

A combined weak signal recovery technique is used in distributed compressive sensing to rebuild sparse signals. This method claims that the scanty description of every signal contains a novel component unique to each signal as well as a universal component maintained by all signals.

$$X^i = Z_c^i + Z_m^i \quad (5)$$

In this approach, the variable  $Z_c^i$  represents the common component shared by all  $X^i$  signals, and its measure of sparseness is determined by the least sparseness measure among all signals as foundation  $\Psi$ . On the other hand, the signals  $Z_{in}^i$  correspond to the distinct parts of the  $X^i$  signals, each having its sparseness measure on the same foundation. The recovery process in this methodology primarily emphasizes reconstructing the common component with high precision. As the proportion of the common component becomes significantly larger than the individual components, the reconstruction error tends to decrease.

### 3.6. Homomorphic Encryption

By allowing aggregation procedures to be carried out on encrypted data, homomorphic encryption techniques offer a workable approach for assuring secure data aggregation. But computationally demanding and time-consuming tasks like encryption and decryption. Information is encoded by the recipient, decoded at intermediate nodes, aggregated, and then encrypted again before being sent to the following hop in link layer cryptography. Queues could become overcrowded as a result of this process, and resource use might go up. Contrarily, homomorphic encryption enables the direct application of some aggregation functions to encrypted data, including sum and average. As a result, the network's sensors are subject to much less work. Every sensor throughout the path applies the aggregate algorithm to the protected information before it is transferred and headed for the base station. The aggregated result is encrypted when it is sent to the base station, which then decrypts it to determine the entire aggregated value. Homomorphic encryption schemes enable arithmetic operations to be performed on ciphertexts. For instance, multiplicatively homomorphic schemes allow efficient manipulation of two ciphertexts, resulting in the multiplication of the corresponding plaintexts upon decryption. Homomorphic encryption proves particularly useful in scenarios where a party lacks the decryption keys but still needs to perform arithmetic operations on a set of ciphertexts.

Consider a probabilistic encryption scheme denoted as  $\text{Enc}()$ , where  $M$  and  $C$  represent the unencrypted text and coded text spaces, respectively. If the set  $M$  creates a grouping under the operation  $\oplus$ , been refer to  $\text{Enc}()$  as a  $\oplus$  homomorphic encryption algorithm. In this context, for any instance  $\text{Enc}()$  of the encryption algorithm, given  $c1 = \text{Enc}(k1, m1)$  and  $c2 = \text{Enc}(k2, m2)$  for some plaintext values  $m1$  and  $m2$  belonging to  $M$ , there exists an effectual algorithm capable of generating a valid ciphertext  $c3 \in C$  from  $c1$  and  $c2$ , using a specific key  $k3$ . This process ensures that the following holds:

$$c3 = \text{Enck3}(m1 \oplus m2) \quad (6)$$

### 3.7. Attribute-Based Encryption (ABE)

Client access control stipulates that, by an access policy, an individual must have exclusive access privileges to a particular set of data. To meet this criterion, Sahai and Waters devised the Attribute-Based Encryption (ABE) cryptographic concept, which expands on Shamir's original identity-based encryption notion. In ABE, an individual, referred to as  $U_i$ , creates an encryption key and a ciphertext that has several properties that describe it. Another user,  $U_j$ , who has a key that matches or overlaps with the properties of the ciphertext from  $U_i$  above a specific threshold, can decrypt this ciphertext.

In this context, let  $M1$ ,  $M2$ , and  $M3$  represent prime order multiplicative cyclic groupings  $p$ . Additionally, considering  $m1$  and  $m2$  being the generators of  $M1$  and  $M2$ , respectively. The Attribute-Based Encryption (ABE) scheme follows four key steps for its execution.

**Setup:** During the setup phase, the system chooses various parameters, including a prime order  $p$  for the bilinear group  $M1$ , a generator  $m$  within  $M1$ , a group of characteristics  $I$ , a bilinear map  $e$ , prime randomized numbers  $t_i$  given to all variable  $i$ , and a single prime randomized number. The algorithm generates the hidden Master's Key (MK) and the publicly available key (PK) using these inputs.

**Key generation:** The inputs consist of User Access (UA) tree  $P$  and the MK. Based on the feature set associated with the leaf nodes of the end UA tree, the algorithm produces either a secret/decryption key SK.

**Encryption:** The message 'is fed as input, a set of features  $I_i$ , and the PK. It utilizes the public key PK to encrypt the message 'g' and produces the ciphertext  $E$  as the output.

**Decryption:** Here, an end user is provided with the ciphertext  $E$ , which is encrypted using the attribute set  $I_i$ , along with the secret key SK (derived from  $P$ ) and the PK. If the attribute set  $I_i$  corresponds to the UA structure  $P$ , the user the  $E$  (ciphertext) using their SK and returns the initial input message 'g' as the output.

The Base Station (BS) assigns each end user a policy for accessing using its UA tree. A user can then decrypt messages returned from a sensor or CH only if their attributes match those of the sensor. For instance, a user  $U_i$ , as shown in Figure 3, would decrypt sensor data if the sensor detects in-body diseases such as lung disease or Spinal disorders and possesses on-body measuring attributes like oxygen rate or bone density. Additionally, the user must have at least two out of four specific expert attributes, such as being a



doctor, medic staff, nurse, or insurance person. On the other hand, the user U<sub>j</sub>, with the UA tree depicted in Figure 4, is unable to decrypt the sensor node’s message. This is because the sensor fails to content the '2/4' threshold geometry of the user's attributes. Specifically, the 'chief' trait has a single common data “doctor”, leading to a wrong output for the AND gate logic.

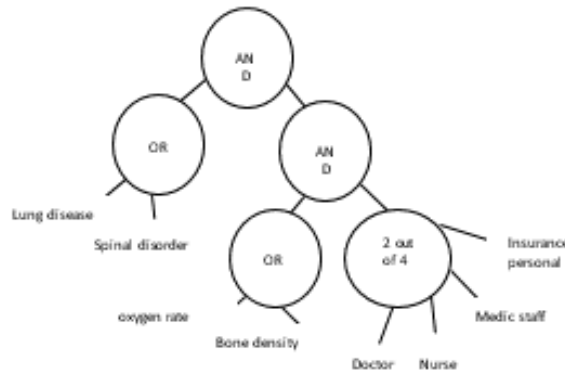


Figure 3. Accessible user structure enabling the decryption of sensor node data

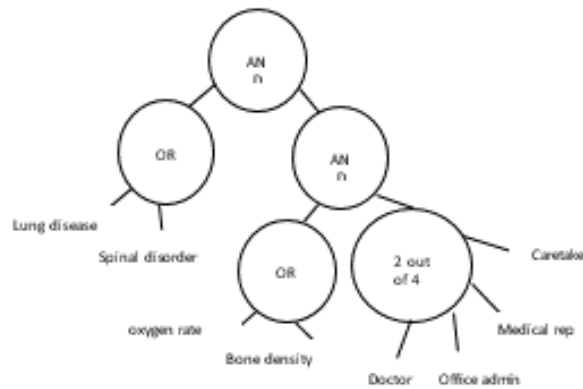


Figure 4. Inaccessible user structure preventing decryption of sensor node data

**3.8. Seagull Optimization Algorithm (SOA)**

In this context, the Cluster/Routing Progression Leader is selected using the Seagull Optimization Algorithm (SOA). In this part, the SOA is explained in comprehensive information. The Laridae family of birds, which includes seagulls, lives in a variety of environments. Seagulls stand out among the various seabird species because they are persistent and have a strong desire to catch prey. Due to these characteristics, seagulls are well renowned for their cognitive abilities and distinctive migration and foraging patterns. Seagulls are picked among other seabirds for their unique qualities and quick judgment. Two crucial procedures are involved in the Seagull Optimization Algorithm: migration and attack.

**3.8.1. Migration**

During the migration process, the seagull must account for and adapt to several different scenarios.

- **Collision avoidance:** To prevent any interference among neighboring agents in the Seagull Optimization Algorithm (SOA), additional constraints are incorporated when determining the optimal position of the exploration agent. This is mathematically represented by the equation provided below.

$$C_s = A \times P(X) \tag{7}$$

The characteristics of the Search Agent's (SA) drive, denoted as *A*, are captured in the equation, taking into account the current iteration *X* and the SA's present location *P<sub>s</sub>*. It is important to note that the movement patterns of the SA, as described by the equation below, are not influenced by residual agents *C<sub>s</sub>*.

$$A = F_c - (X \times (F_c / \text{maxiteration})) \tag{8}$$

In the given equation,  $X$  represents the current iteration, ranging from 0 to *max iteration*. The value of  $F_c$  is fixed at 2, while  $A$  is linearly scaled and reduced from  $F_c$  to 0. The frequency of the constraint is controlled by the parameter  $F_c$ .

- **Directional movement toward ideal neighborhood:** Once the collision among neighboring agents is successfully resolved, the exploration agents are directed toward the optimal movements of their neighbors. This direction is expressed in the following formulation.

$$M_s = B \times (P_b(X) - P_s(X)) \quad (9)$$

In the given equation, the SA and its position are represented by  $P(X)$  and  $M_s$ , respectively.  $B$  denotes a random agent responsible for effective evaluation between examination and manipulation. The exploration agent with the highest fitness is denoted as  $P(X)$ . The calculation of the random variable is expressed in the equation provided below.

$$B = 2 \times S = A^2 \times RD \quad (10)$$

In the given equation,  $RD$  represents a random variable that ranges between 0 and 1, encompassing a diverse set of values.

- **Stay close to the best SA possible:** The equation below illustrates how the position of the revised SA and the optimal SA are related

$$DS = |C_s + M_s| \quad (11)$$

In this equation,  $DS$  represents the separation between the current and most suitable SA s.

### 3.8.2. Prey Attacking

The main motivation behind employing this algorithm is its ability to minimize computational requirements during the exploration phase. In the attacking process, seagulls adjust their exile state by prioritizing altitude preservation, taking into account factors such as air currents and weight. When attacking prey, seagulls may execute twisting movements while in midair. These twisting motions can be characterized by the equations provided below.

$$X = R \times \cos K$$

$$Y = R \times \sin K$$

$$Z = R \times K$$

$$R = U \times e^{KV} \quad (12)$$

In the given equations, the natural logarithm base is denoted as  $e$ . The quantities  $u$  and  $v$  represent the spiral shape.  $k$  is an arbitrary value within the limits  $[0 \leq k \leq 2\pi]$ , and  $R$  represents the spiral's extension after each iteration. The updated movement of the SA is computed using the equation provided below.

$$P(X) = (DS \times X \times Y \times Z) + P_{bs}(X) \quad (13)$$

In this equation,  $P_b(X)$  refers to the optimal response, representing the position of the remaining SAs.

### 3.9. Whale Optimization Algorithm (WOA)

WOA is an intelligent algorithm based on swarm behavior, designed to tackle continuous optimization problems. It has demonstrated exceptional performance when compared to other meta-heuristic methods. Notably, WOA differs from other algorithms that are inspired by nature in that it is straightforward to construct and robust. The method simply has to have a single control factor (time interval) adjusted on the majority of occasions. A colony of humpback whales in WOA searches a multi-dimensional space for food. The locations of the individual whales stand in for various decision-making factors, and the separation among the whales, as well as the food, represents the objective cost. Three operational processes have an impact on a whale's location: the shrinkage of encircling prey, the bubble-net attack technique (exploitation phase), and the hunt for food (exploration phase). The general flowchart of WOA is depicted in Figure 5.

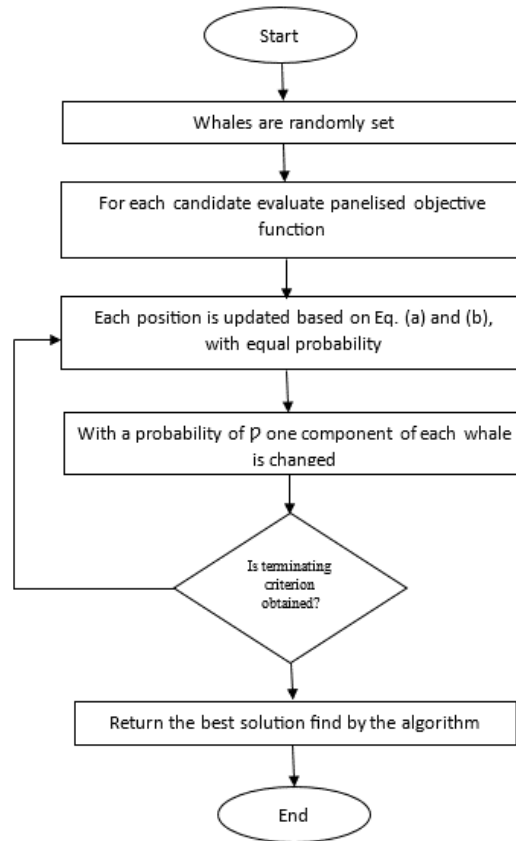


Figure 5. WOA Flowchart

- **Shrinkage of encircling prey:** Humpback whales possess the ability to identify the whereabouts of prey and surround them. In the WOA, it presumes that the best candidate solution currently available matches the target prey or is very close to the optimum because the precise location of the optimal approach inside the search environment is unknown. The most suitable SA will be chosen, and the other SAs will modify their placements to be near the most effective SA. This behavior is mathematically represented by the following equations.

$$\begin{aligned}
 \vec{D} &= |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \\
 \vec{X}(t+1) &= \vec{X}^*(t) - \vec{A} \vec{D} \\
 \vec{A} &= 2 \vec{a} \cdot \vec{r} - \vec{a} \\
 \vec{C} &= 2 \cdot \vec{r}
 \end{aligned}
 \quad \dots (a)$$

In the given equations,  $\vec{X}^*$  represents the overall best position,  $\vec{X}$  denotes the position of a whale,  $t$  indicates the present iteration,  $a$  linearly decreases from 2 to 0 throughout the iterations, and  $r$  is a random number uniformly distributed between 0 and 1. The notation “ $|\cdot|$ ” denotes absolute value.

- **Bubble-net attack technique (exploitation phase):** To simulate the helix-shaped movement of humpback whales during the bubble-net behavior, a spiral algebraic equation is employed, connecting the positions of the whale and the prey.

$$\begin{aligned}
 \vec{X}(t+1) &= \vec{D}^k \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \\
 \vec{X}(t+1) &= \vec{X}^*(t) - \vec{A} \vec{D}, \text{ if } p < 0.5 \\
 &= \vec{D}^k \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), \text{ if } p \geq 0.5
 \end{aligned}
 \quad \dots (b)$$

In this equation, the constant  $p$  is utilized to describe the spiral's nonlinear form, while  $k$  is a subjective numeral that follows a uniform distribution within the range of -1 to 1.

- **Search for prey (exploration phase):** To achieve global optimization, once the value of A is greater than 1 or less than -1, the SA is restructured by adopting the characteristics of a arbitrarily selected SA as a replacement for of relying solely on the best SA.

$$\begin{aligned} \vec{D} &= |\vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}| \\ \vec{X}(t+1) &= \vec{X}_{\text{rand}} - \vec{X} \vec{D} \end{aligned} \quad (14)$$

In this equation,  $\vec{X}_{\text{rand}}$  is selected at random from the whales present in the current iteration.

### 3.10. Proposed: Optimized Attribute-Based Encryption (O-ABE)

Hybrid advanced techniques in homomorphic encryption involve the integration of multiple encryption schemes or cryptographic primitives to enhance efficiency, security, or functionality. These hybrid approaches are designed to overcome the limitations of individual encryption schemes and capitalize on their unique advantages. In this research, an optimized Attribute-Based Encryption (ABE) with Homomorphic Encryption is employed for encryption purposes. To further enhance the encryption process, a novel hybrid optimization method is introduced, combining the Seagull Optimization Algorithm (SOA) and the Whale Optimization Algorithm (WOA).

### 3.11. Blockchain-based Data Integrity

Blockchain-based data integrity schemes effectively address the trust issues associated with Third-Party Auditors (TPAs), although they do encounter challenges related to computational and communication overhead. Blockchain, known for its transparency, security, immutability, and decentralization, is a promising technique that combines various fields, including cryptography, mathematics, and peer-to-peer systems, to tackle traditional record synchronization problems. It serves as an immutable digital ledger, distributing and processing transaction records across the network using Distributed Ledger Technology (DLT) and cryptographic signatures called hashes.

Within the blockchain framework, the Merkle Tree (MT), also known as a hash tree, plays a crucial role in encoding blockchain data securely and systematically. It enables rapid verification and efficient data transfer within the peer-to-peer (P2P) blockchain network. Each transaction in the blockchain is associated with a hash value, and these hash values are stored in a tree-like structure rather than in sequential order within a block. A parental-child tree interaction is created by connecting the hash values to their parent hashes. Additionally, a Merkle Tree framework, which is frequently utilized for data integrity checking, is created by combining all transaction hashes included within a block. Several different organizations, including the client, the Key Generation Center (KGC), the cloud storage computer, and the blockchain itself, are included in the blockchain-oriented structure for data integrity verification. The establishing stage, the execution stage, and the validation phase are the three stages of the hypothesized data integrity verification procedure.

**a) Setup Stage:** This stage involves the presence of client devices and the Key Generation Center (KGC). The KGC plays a major part in generating the private key ( $Pr_k$ ) and the public key ( $Pb_k$ ) based on the client's selected input security parameter, denoted as  $k$ . Each device is assigned a private key ( $Pr_k$ ) that is utilized to create data file tags, represented as  $H(di)$ . On the other hand, the public key ( $Pb_k$ ) is employed for verifying the file confidentiality of stored data. The generation of the private and public key within the network is determined by the following equation: whereas  $\alpha \rightarrow G$  random variable and computed  $u \leftarrow g\alpha$ .

$$\begin{aligned} Pr_k &= (\alpha, SPr_k) \\ Pb_k &= (u, SPb_k) \end{aligned} \quad (15)$$

In the scenario where a device possesses a large data file intended for storage in the cloud-based storage server for storing and data processing, the client proceeds to divide the data file D into multiple data shares of equal length, denoted as  $\{d_1, d_2, d_3, \dots, d_n\}$ , as represented in the following equation:

$$D = (d_i) \quad (16)$$

After dividing the data into individual blocks, denoted as  $d_i$ ; such as  $i = 1, 2, \dots, n$ , the client proceeds to generate a digital signature for each block using the EiGmal algorithm. Since this approach is probabilistic, it provides a high level of security. The digital signature, also known as the digest of the data, is

created by encrypting the hash function of the data using the client's private key. The resulting digital signature, denoted as  $S_i$ , is then appended to the original data block  $d_i$ . The digital signature for each data block is expressed as follows:

$$S_i = (H(d_i) \cdot v^{d_i})^\alpha \quad (17)$$

The data block's tag file value  $d_i$  is denoted as  $H(d_i)$ , and a random element  $v$  is generated from the set  $G$ . The  $S_i$  represents the signature sequence  $\delta = S_i$ , ( $1 \leq i \leq n$ ) for the various data bottlenecks in the system. The user obtains the rootR depending on the Merkle Tree (MT) structure. In the MT, the leaves node generate a sequence of # values for the data file tags  $H(d_i)$ , such as  $i = 1, 2, \dots, n$ . The user then encrypts the rootR utilizing the PK  $\alpha$ , and this process is represented by the following equation:

$$\alpha = \text{Sign}_{\text{Prk}}(H(R)) \quad (18)$$

The digital signature  $\text{Sign}_{\text{Prk}}(H(R))$ , generated by the encrypted key on the root (main) node of the Merkle Tree (MT), is denoted as the signature on  $H(R)$ . The client constructs a transferring data file  $\{D, A, t_s, \delta, \text{Sign}_{\text{Prk}}(H(R))\}$ , where  $D$  represents the data,  $A$  represents additional information,  $t_s$  represents the timestamp,  $\delta$  represents the signature set, and  $\text{Sign}_{\text{Prk}}(H(R))$  represents the digital signature. This constructed data file is then forwarded to the server through a smart contract.

**b) Processing Stage:** In this stage, the verification process takes place within blockchain technology, ensuring data security and posing a barrier to the provider of cloud services. Before issuing the challenge, the blockchain first verifies the signature patterns on  $A$  using the PK. If the verification flops, it is disallowed and marked as untrue. Otherwise, it is accepted and recovered as  $v$ . Let  $A = \text{nameInlv} \parallel \text{Sign}_{\text{Prk}}$ , where  $(\text{nameInlv})$  represents the File Tag (FT) for  $D$ . The blockchain verifier generates the barrier, "bar", for the cloud-server such as the prover by randomly decide on elements from a subset, as given away in the equation below.

$$J = \{s_1, \dots, s_c\} \text{ of set } [1, n] \quad (19)$$

Where, for  $s_1 \leq \dots \leq s_c$ , and  $i \in J$ , the blockchain randomly selects an part  $u_i$  from  $Z_p$ . The message "bar" postulates the positions of the blockage to be verified in this step. The blockchain sends the bar  $\{(i, v_i)\}$  for  $s_1 \leq i \leq s_c$  to the server. After getting the challenging message "bar" from the verifier, the prover calculates and creates the verifications as shown in the equations below. Additionally, the prover provides auxiliary data  $\{\Omega_i\}$  for  $s_1 \leq i \leq s_c$ , which represents the node brethren on its way through the leaves to the rootR of the Merkle Tree (MT).

$$\begin{aligned} \mu &= \sum_{i=s_1}^{s_c} u_i d_i \in Z_p \\ \sigma &= \prod_{i=s_1}^{s_c} S_i^{u_i} G \end{aligned} \quad (20)$$

The prover generates the "proof (P)" in response to the verifier, which is formulated as follows:

$$\text{proof (P)} = \{\mu, \sigma, \{H(d_i), \Omega_i\}_{s_1 \leq i \leq s_c}, \text{Sign}_{\text{Prk}}(H(R))\} \quad (21)$$

**c) Verification Stage:** Post receiving the replication from the prover, the verifier calculates the rootR by utilizing  $H(m_i), \Omega_i\}_{s_1 \leq i \leq s_c}$ , and verifies its correctness by evaluating the following equation:

$$e(\text{Sign}_{\text{Prk}}(H(R)), g) = e(H(R), g^\alpha) \quad (22)$$

If the verification process is unsuccessful, the verifier rejects the result. Otherwise, the verifier proceeds to further validate by checking the following equation. If the output satisfies the validation criteria, it is accepted; otherwise, it is rejected.

$$e(\sigma, g) = e(\prod_{i=s_1}^{s_c} H(d_i)_i^v \cdot v^\mu, v) \quad (23)$$

#### 4. RESULT AND DISCUSSION

The designed model has undergone validation in Python about various performance metrics, including energy consumption, lifetime, packet delivery ratio, throughput, encryption and decryption time, and data security. A comparative analysis was conducted against well-established approaches such as Attribute-Based Encryption (ABE), Homomorphic Encryption (HE), RSA, and Blowfish. The simulation outcomes have demonstrated the high-end performance of the proposed approach across these metrics. Notably, the proposed approach showcased a 12% reduction in energy consumption, a 6% increase in network lifetime, a 5% improvement in throughput, a 13% increase in delivery rate, and a significant reduction of 33% and 50% in

encryption and decryption time, respectively, when compared to existing approaches. Furthermore, the proposed approach achieved a notable 12% enhancement in data security. These results strongly indicate that the proposed approach is a highly promising feature for addressing the challenges faced by unattended wireless sensor networks, offering improved performance and enhanced data security compared to existing state-of-the-art methods. The impact of increasing nodes on energy consumption as follows,

- The proposed model consistently exhibits the lowest energy consumption across all network sizes.
- For 20 nodes, the proposed model consumes 46.3282 mJ, which is significantly lower than ABE (59.48859 mJ), HE (63.6203 mJ), and RSA (69.7566 mJ).
- For 100 nodes, the proposed model maintains its efficiency at 92.6937 mJ, while ABE (105.5343 mJ), HE (118.6385 mJ), and RSA (119.3052 mJ) consume much higher energy.

The results indicate that as the network scales up, traditional encryption models lead to exponentially higher energy consumption, making them unsuitable for real-world UWSN deployments. The proposed model reduces energy consumption by up to 33.6% compared to ABE, 41.3% compared to HE, and 45.2% compared to RSA when considering an average across all node counts. The significant improvement is attributed to machine learning-based optimization of encryption parameters and dynamic CH selection, which ensures energy-efficient data transmission and encryption processing.

**Table 1.** Energy Comparison of existing models and proposed one

No of Nodes	Energy Consumption				
	20	40	60	80	100
Proposed	46.3282	43.6378	51.5177	101.2894	92.6937
ABE	59.48859	45.49069	63.39274	106.8113	105.5343
HE	63.6203	52.98258	68.08014	112.5236	118.6385
RSA	69.75661	65.31389	74.43096	123.7868	119.3052

Figure 6 illustrates the energy usage of various encryption schemes as the number of nodes increases. The suggested model regularly exhibits reduced energy usage relative to ABE, HE, and RSA, making it more appropriate for resource-limited UWSNs. All models show an escalation in energy use as the network expands, with RSA displaying the most energy usage, signifying its inefficiency in extensive implementations. The optimised cluster head selection and encryption algorithms of the proposed model enhance energy efficiency, hence prolonging network lifespan and improving performance in underwater wireless sensor networks (UWSNs). The below Figure 7 shows the Lifetime Vs Number of nodes, Figure 8 shows the Packet Delivery Ratio Vs Number of nodes and Figure 9 shows the Throughput Vs Number of nodes.

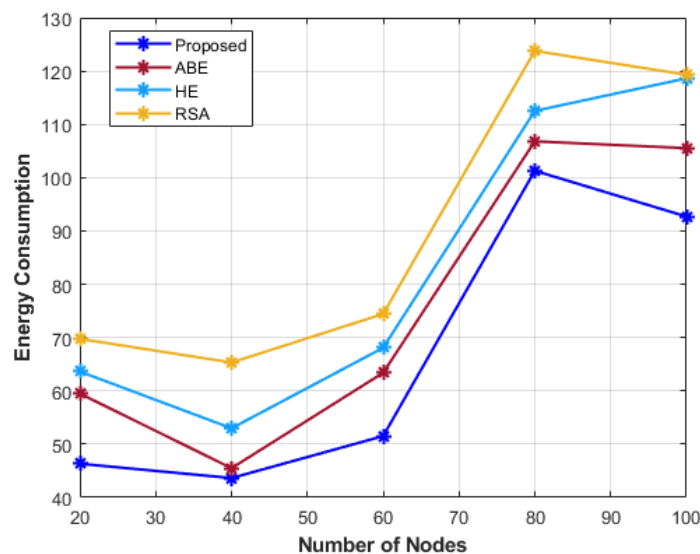


Figure 6. Energy Consumption Vs Number of nodes

**Table 2** Lifetime Comparison of existing models and proposed one

	Lifetime				
No of Nodes	20	40	60	80	100
Proposed	8.543598	9.916879	7.756868	7.373262	9.28159
ABE	5.579051	5.808531	6.878058	6.86268	8.702651
HE	3.060171	5.922567	6.488521	7.153008	6.880037
RSA	1.105496	5.992791	3.845463	4.631537	5.062665

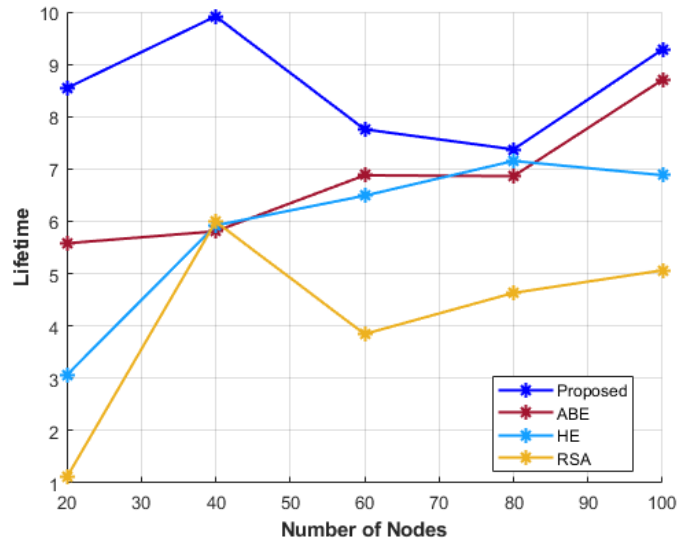


Figure 7. Lifetime Vs Number of nodes

**Table 3** Packet delivery Comparison of existing models and proposed one

	Packet Delivery Ratio				
No of Nodes	20	40	60	80	100
Proposed	0.99	0.97	0.96	0.95	0.88
ABE	0.98	0.9	0.88	0.81	0.72
HE	0.96	0.93	0.89	0.79	0.77
RSA	0.89	0.84	0.78	0.7	0.61

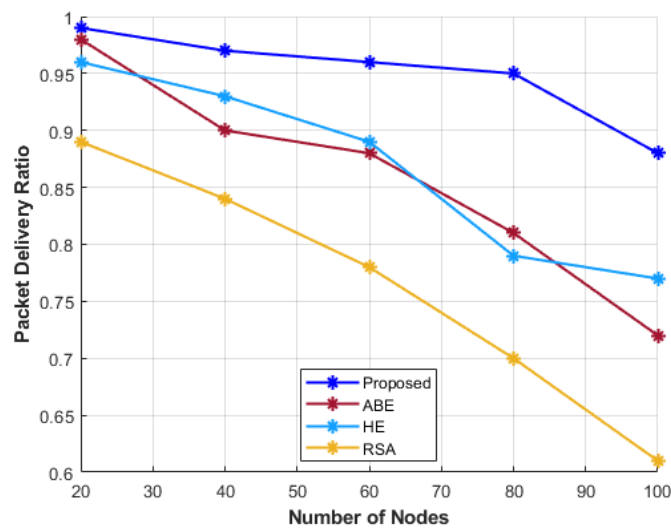
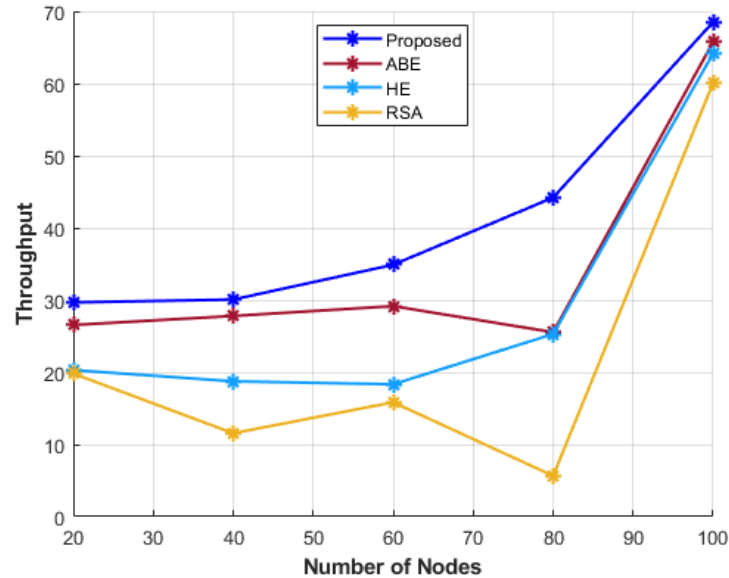


Figure 8. Packet Delivery Ratio Vs Number of nodes

**Table 4** Throughput Comparison of existing models and proposed one

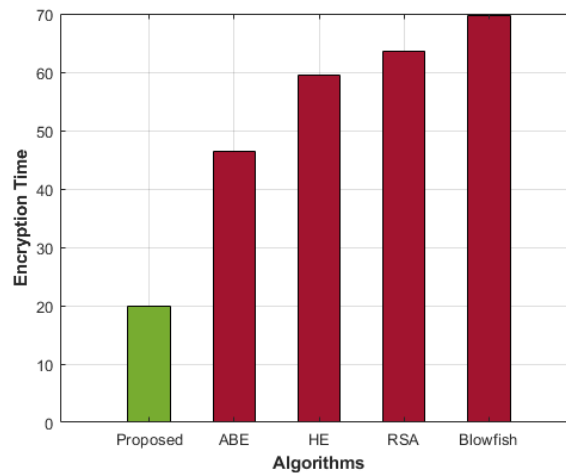
	Throughput				
No of Nodes	20	40	60	80	100
Proposed	29.6822	30.0789	34.9014	44.2081	68.5227
ABE	26.5567	27.80866	29.16621	25.55406	65.75356
HE	20.30796	18.75345	18.34168	25.29887	64.15962
RSA	19.82385	11.55633	15.8493	5.660432	60.08264

**Figure 9.** Throughput Vs Number of nodes**Table 5** Security Implementation & comparison

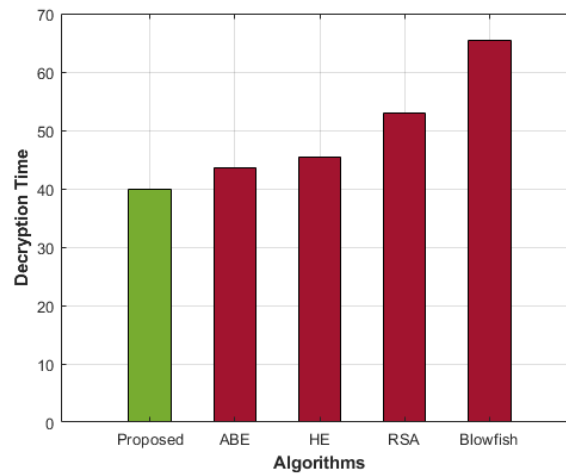
Models	Encryption Time (Sec)	Decryption Time (Sec)	Security (%)
Proposed	1.9963	0.8563	98.53
ABE	3.363983033	1.2354	87.2
HE	1.533306816	2.068451143	91.64
RSA	6.153565902	8.00629984	85.645
Blowfish	4.771	3.772400956	85

The encryption time is a critical factor in resource-constrained UWSNs. The proposed model achieves an encryption time of 1.9963s, which is significantly lower than ABE (3.3640s), RSA (6.1536s), and Blowfish (4.7710s). While HE has the lowest encryption time (1.5333s), its higher decryption overhead makes it less suitable for real-time applications. Security is a vital aspect of encryption models. The **proposed model achieves the highest security (98.53%)**, outperforming all traditional encryption models. In contrast, ABE and Blowfish provide the lowest security levels (87.2% and 85%, respectively), while HE (91.64%) and RSA (85.645%) also fall short of the proposed method. This improvement is attributed to the hybrid encryption approach, which combines Attribute-Based Encryption (ABE) and Homomorphic Encryption (HE) with machine learning-driven optimization. The below Figure 10 shows the Encryption Time Vs Algorithms.

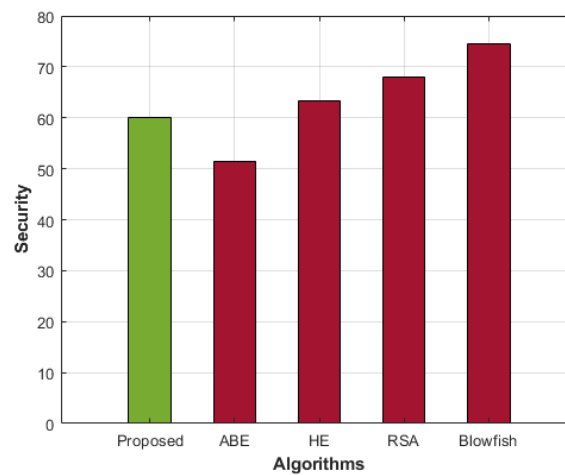




**Figure 10.** Encryption Time Vs Algorithms



**Figure 11.** Decryption Time Vs Algorithms



**Figure 12.** Security Vs Algorithms

The Figure 11 illustrates the decryption time comparison among different encryption algorithms. The proposed model exhibits the lowest decryption time, highlighting its efficiency for real-time applications in Unattended Wireless Sensor Networks (UWSNs). In contrast, traditional encryption techniques such as RSA and Blowfish require significantly more time to decrypt data, making them unsuitable for low-latency and energy-constrained environments. The Figure 12 compares the security levels of the proposed model with ABE, HE, RSA, and Blowfish. While RSA and Blowfish provide slightly higher security, they come at the cost of increased computational overhead and decryption delay. The proposed model, however, strikes an

optimal balance by achieving high security while maintaining computational efficiency, making it a superior choice for secure and energy-efficient data transmission in UWSNs.

## 5. CONCLUSION

In this study, we propose using machine learning-based Channel (CH) selection and optimised hybrid homomorphic encryption to improve data survivability in Unattended Wireless Sensor Networks (WSNs). After comprehensive testing, we found that our methodology beats current approaches in energy consumption, longevity, packet delivery ratio, throughput, encryption and decryption time, and data security. Our findings show that our machine learning-based CH selection method considerably decreases UWSN energy usage. We reduce sensor node power consumption by picking the best CHs. Thus, the UWSN's lifespan is extended, allowing long-term data collection and transmission. PDR and throughput have improved significantly using machine learning-based CH selection. Our method optimises resource allocation for efficient data transmission and low packet loss. PDR and throughput boost data transmission rates, improving UWSN performance. Our optimised hybrid homomorphic encryption technique features fast encryption and decoding. Our technique balances security and computing performance by combining symmetric and asymmetric encryption algorithms. This secures WSN data without slowing the system. Data security in UWSNs is crucial owing to the sensitive data gathered. The optimised hybrid homomorphic encryption approach we offer protects against security concerns. It protects data while processing it efficiently. We proved our technique is safe and can defend the UWSN from numerous security assaults via comprehensive research and vulnerability testing. Our suggested approach outperforms current methods in energy consumption, longevity, packet delivery ratio, throughput, encryption and decryption time, and data security. Machine learning-based CH selection and optimised hybrid homomorphic encryption increase unattended UWSN survival and dependability in real-world circumstances. WSNs and data security may benefit from further study in this area.




## REFERENCES

- [1] X. Fu, P. Pace, G. Aloï, W. Li, and G. Fortino, "Toward robust and energy-efficient clustering wireless sensor networks: A double-stage scale-free topology evolution model," *Computer Networks*, vol. 200, p. 108521, 2021.
- [2] X. Fu, P. Pace, G. Aloï, W. Li, and G. Fortino, "Toward robust and energy-efficient clustering wireless sensor networks: A double-stage scale-free topology evolution model," *Computer Networks*, vol. 200, p. 108521, 2021.
- [3] [1] Q. Liu et al., "Cluster-based flow control in hybrid software-defined wireless sensor networks," *Computer Networks*, vol. 187, p. 107788, 2021.
- [4] S. P. Singh and S. C. Sharma, "A PSO based improved localization algorithm for wireless sensor network," *Wireless Personal Communications*, vol. 98, pp. 487–503, 2018.
- [5] B. Bhushan and G. Sahoo, "Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks," *Wireless Personal Communications*, vol. 98, pp. 2037–2077, 2018.
- [6] S. Arjunan and S. Pothula, "A survey on unequal clustering protocols in wireless sensor networks," *Journal of King Saud University-Computer and Information Sciences*, vol. 31, no. 3, pp. 304–317, 2019.
- [7] M. Elappila, S. Chinara, and D. R. Parhi, "Survivable path routing in WSN for IoT applications," *Pervasive and Mobile Computing*, vol. 43, pp. 49–63, 2018.
- [8] V. Vashishth, A. Chhabra, A. Khanna, D. K. Sharma, and J. Singh, "An energy efficient routing protocol for wireless Internet-of-Things sensor networks," 2018, doi: arXiv:1808.01039
- [9] A. M. Sadri, S. Hasan, S. V. Ukkusuri, and M. Cebrian, "Exploring network properties of social media interactions and activities during Hurricane Sandy," *Transportation research interdisciplinary perspectives*, vol. 6, p. 100143, 2020.
- [10] J. C. Knight and K. J. Sullivan, "On the definition of survivability," 2000.
- [11] D. Chen, S. Garg, and K. S. Trivedi, "Network survivability performance evaluation: A quantitative approach with applications in wireless ad-hoc networks," *Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pp. 61–68, Sep. 2002.
- [12] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on computing, network, and storage resource constraints," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3298–3304, 2017.
- [13] S. Ezdiani, I. S. Acharyya, S. Sivakumar, and A. Al-Anbuky, "Wireless sensor network softwarization: Towards WSN adaptive QoS," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1517–1527, 2017.
- [14] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, pp. 374–387, 1998.
- [15] K. Islam, S. G. Akl, and H. Meijer, "Maximizing the lifetime of wireless sensor networks through domatic partition," *2009 IEEE 34th Conference on Local Computer Networks*, pp. 436–442, Oct. 2009.
- [16] Z. Yang and K. L. Yeung, "SDN candidate selection in hybrid IP/SDN networks for single link failure protection," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 312–321, 2020.




- [17] S. Li, "Efficient multi-path protocol for wireless sensor networks," *International journal of wireless & mobile networks (IJWMN)*, 2010.
- [18] J. S. Raj and A. Basar, "QoS optimization of energy efficient routing in IoT wireless sensor networks," *Journal of ISMAC*, vol. 1, no. 01, pp. 12–23, 2019.
- [19] S. Jannu and P. K. Jana, "A grid based clustering and routing algorithm for solving hot spot problem in wireless sensor networks," *Wireless Networks*, vol. 22, pp. 1901–1916, 2016.
- [20] S. Smys and R. Bestak, "Introduction to the Special Issue on Inventive Network Structures for Next Generation Wireless Personal Systems," *Wireless Personal Communications*, vol. 90, pp. 421–422, 2016.
- [21] Q. Fan, J. Chen, L. J. Deborah, and M. Luo, "A secure and efficient authentication and data sharing scheme for Internet of Things based on blockchain," *Journal of Systems Architecture*, vol. 117, p. 102112, 2021.
- [22] S. Smys, "Energy-aware security routing protocol for WSN in big-data applications," *Journal of ISMAC*, vol. 1, no. 01, pp. 38–55, 2019.
- [23] M. F. Çorapsız, "Lifetime maximization of wireless sensor networks while ensuring intruder detection," *Soft Computing*, vol. 28, no. 5, pp. 4197–4215, 2024.
- [24] D. Sivaganesan, "Efficient Routing Protocol with Collision Avoidance in Vehicular Networks," *Journal of Ubiquitous Computing and Communication Technologies (UCCT)*, vol. 1, no. 02, pp. 76–86, 2019.
- [25] P. Wu, "Research on Performance Improvement of Wireless Sensor Networks Based on OPM Algorithm," *Open Access Library Journal*, vol. 11, no. 7, pp. 1–12, 2024.

### BIOGRAPHIES OF AUTHORS



**Haritha K. Sivaraman**    is an assistant professor, in Department of Electronics and Communication Engineering, RajaRajeswari College of Engineering, Bangalore, India. Her major interests are wireless sensor networks, energy efficient designs in sensor networks, internet of things, and wireless communication. She can be contacted at email: haritharesearchscholar@gmail.com.



**Rangaiah Leburu**    is a professor in Department of Electronics and Communication Engineering, RajaRajeswari College of Engineering, Bangalore, India. His major interests are VLSI, wireless communications, sensor networks, Ad-Hoc networks, and Routing protocols in wireless networks. He can be contacted at email: rleburu@gmail.com.