

CHAOTIC BOBCAT OPTIMIZATION ALGORITHM (CBOA) BASED ENERGY EFFICIENT CLUSTERING ROUTING PROTOCOL FOR UNDERWATER WIRELESS SENSOR NETWORKS

¹Mrs.Shunmugavadivoo.T,

Research Scholar, Department of Computer Science Hindusthan College of Arts & Science
Coimbatore

²Dr.M.Selvapriya,

Assistant professor Department of Computer Science Hindusthan College of Arts & Science
Coimbatore

Corresponding author name: Mrs.Shunmugavadivoo.T,

Corresponding author mail id: sunmusankar@gmail.com

ABSTRACT

The Underwater Wireless Sensor Network (UWSN) was created since there is a rising need for real-time data collection, remote monitoring, and autonomous operation in underwater environments. So, the main things to think about in a UWSN are how long the network will last and how much energy it would need. The routing protocol for a UWSN is in-charge of finding the optimal method to transport data packets to the surface destination in a way that is both fast and effective. Clustering and routing methods can help UWSNs use less energy. A new protocol for UWSNs introduces CBOA-MHDT - Chaotic Bobcat Optimization Algorithm-Multi-Hop Data Transmission. The CBOA-MHDT technique chooses cluster heads (CHs) and groups them into clusters based on things like how much energy is left, how far apart the clusters are, and how far apart the clusters are from each other. The CBOA method uses the Bobcat Optimization method (BOA) and chaos to make optimization work better. The hunting techniques of bobcats inspired BOAs, and the addition of chaos is meant to make the optimization process better at finding and using resources. After that, each cluster gets a CH to gather data packets from sensor nodes that are distributed around at random. Things like residual energy, distance, and node degree, were considered by the suggested method, which finds a fitness function. MHDT does not need sensor nodes and the sink to talk to each other directly. It passes the data across several intermediate nodes instead. Simulations have been executed in Matrix Laboratory R2021b (MATLABR2021b) to assess and contrast the proposed algorithm's efficacy against established clustering techniques, utilizing metrics such as PLR - Packet Loss Ratio, network lifetime, throughput, Residual Energy, and PDR - Packet Delivery Ratio.

Keywords: Underwater wireless sensor networks, Weight Clustering Approach (WCA), Multi-Hop Data Transmission, Network topology, Chaotic Bobcat Optimization Algorithm, chaotic operator, Energy Efficient clustering routing.

1. INTRODUCTION

The oceans cover over 140 million square miles of the world, which is an oceanic planet. But just 10% of the oceans have been surveyed so far [1]. Wireless Sensor Networks could be highly beneficial for keeping an eye on water environments because they can wirelessly sense, gather, and send data to users in real time, which in turn leads to a new type of wireless sensor technology known as the Underwater Wireless Sensor Network (UWSN). Deep underwater, where sensors employ sound waves to transmit and receive messages, UWSN technologies are deployed and put into operation.

The undersea network of nodes that has been set up is used for several things, including exploration, supervision, and military objectives. Sensors can detect and focus on changes in environmental factors like temperature and pressure. With the data from the sensors, it is feasible to

look at the environment and keep track of how it changes [2, 3]. It is not too hard to use these devices. Using an acoustic modem, it communicates with each other and to the sink node by an RF modem [2]. Modern UWSNs use acoustic modems to talk to each other because deepwater conditions make it harder for channels to work and cause a lot of loss of transmission. Some of these reasons include finding out the pollution levels, bacteria, oxygen, salinity and water's temperature; to find mines; and to predict how the weather will change studying how people affect marine ecosystems; and finding underwater oil fields [4, 5].

UWSN is a network that lets you keep an eye on tasks in a certain area. There are smart sensors and cars that can talk to each other wirelessly [6]. The sink receives the information from sensor nodes on the surface. The sink node has a transmitter that can change the sound waves it receives from nodes that are below the surface. The transceiver can broadcast and take in radio frequency signals over long distances, which allow it talk to the station on land. The data that is collected is either used in the same place or sent to another network for a certain reason [6]. New communication technologies let underwater sensor nodes and apps talk to each other and give clients services that can alter based on what they want and need. The UWSNs sector is growing quickly, and modern sensor node technologies are now widely used in many fields. This means that awareness needs to grow every year [6]. Figure 1 shows a general picture of the UWSN environment.

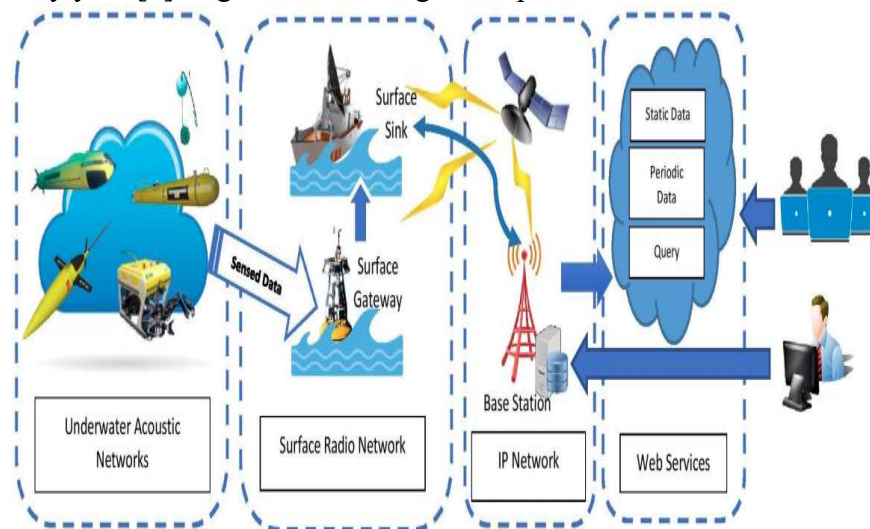


Figure 1: Overview of UWSN [1]

Loss of Path, attenuation, geometric spreading, multipath fading, delay, and Doppler spread were challenges faced by UWSN [7–8]. The development of routing protocols improves the way nodes talk to each other [9–10]. However, routing algorithms face unique challenges when navigating underwater areas due to the aquatic circumstances. The restrictions are sluggish data transfer speeds, not enough channel capacity, high bit error rates, greater energy use, long delays, weak signals, and changes in the network structure over time [10–11].

Depth-based routing and other current UWSN routing approaches strive to cut down on transmission and attenuation losses, but they typically do not work well in real time because they are not flexible or efficient. Compared to radio signals, acoustic transmissions in UWSNs use a lot less energy. This makes the delays longer and the faults in the data packets that are transmitted more often. To effectively address issues arising in underwater settings, it is essential to develop routing protocols specifically tailored for underwater applications that facilitate energy-efficient communication while adhering to the stringent performance criteria established by UWSNs [11]. To

help UWSNs consume less energy, a lot of routing solutions have been made, however the recurring hotspot problem still impacts these networks, which makes sensors fail too fast.

Multi-hop data transfers are far more efficient than single-hop data transfers for long-distance transmissions for UWSN. Also, a consistent network topology is critical for balancing traffic and avoiding data collisions [12]. Cluster-based routing approaches put sensor nodes into groups called clusters to make the network run better [13–14]. Cluster-based routing works for ordinary wireless networks, but not for a UWSN because of the limited bandwidth, spread stay, underwater current, and probability of error [15]. The multi-hop method utilizes less power compared to the single-hop method when delivering data over long distances.

Many studies have demonstrated that cluster routing algorithms effectively prevent collisions. They balance the amount of traffic, and they transport data between clusters using multi-hop methods [13]. A cluster routing approach has a lot of groups of nodes. Cluster head and a lot of "cluster member nodes" are there in each group. The CH gives out channels for sending data as soon as clusters are established. The cluster member provides data based on the distribution that might keep things from crashing into each other [14]. Then, CHs handle aggregation, which can cut down on data duplication and the data packet's count that need to be delivered to the Sensor Node (SN). This helps save energy [15]. CH stands for "full cluster." SNGather information and transmit it to the surface sinks through other CHs. The clustering strategy cuts down on the overall travel and communication expenses for the other nodes because only the CH transmits the data to the next-hop CH, which then transmits it to the surface sinks.

It is observed that UWSN can conserve energy by using clustering and multi-hop routing. Due to the underwater current, high water pressure, poor bandwidth, propagation latency, and high error rate, UWSN wouldn't function with cluster-based protocols for ordinary wireless networks. Nondeterministic polynomial-time (NP) hard optimization comprises of both clustering and routing that can be solved with metaheuristics issues. This work introduces an innovative protocol for UWSN, designated CBOA-MHDT and this protocol uses a weighted clustering method (WCA) to find the best cluster heads and make clusters. To determine the best group of cluster heads and attain the goal, the CBOA method is utilized. After then, each group gets a CH that gathers data packets from sensor nodes that are spread out randomly. Matrix Laboratory R2021b (MATLABR2021b) is used to see how well the suggested strategy performs compared to alternative ways of grouping things.

2. LITERATURE REVIEW

A novel CSRO-MHDT was developed by Anuradha et al. [16], denoting "chaotic search-and-rescue-optimization-based multi-hop data transmission protocol for UWSNs," referring to UWSNs. The CSRO-MHDT method picks the CHs and organizes them. This informs us a few things, like how much energy is left, how far apart the clusters are, and how far apart they are from each other. The CSRO algorithm combines the SRO approach with chaotic principles. The CSRO-MHDT approach uses a fitness function that considers the distance, node's degree, and remaining energy, among other factors. The CSRO-MHDT approach works because it sends 88% of packets. EECRP (energy-efficient clustering routing protocol), FCMMFO (fuzzy C-means and moth-flame optimization), FBCPSO (fuzzy scheme and particle swarm optimization), EGRC (energy-efficient grid routing based on 3D cubes) and LEACH-ERE (low-energy adaptive clustering hierarchy based on expected residual energy methods), were the samples of these methods.

A new EECRP- the Energy-Efficient Clustering Routing Protocol was suggested by Xiao et al. [17], for UWSMs, which uses data fusion and a genetic algorithm. From the CMN - Cluster

Member Node, the CHN - Cluster Head Node of the clustering routing protocol collects data and it combines it with an upgraded BPNN (Back Propagation Neural Network), and transmits the combined data to a SN - Sink Node in several hops. The better GA finds the best method for data to move between the CHN and the SN in several hops. This makes the transmission faster and uses less energy. For the best CHN selection in the clustering model, GA employs a specific method to encode data, a certain way to cross over data, and a better way to mutate data. An improved momentum technique can make BPNN better for data fusion by cutting down on redundant data and the amount of data sent, which saves energy. Also, an effective way to choose CHNs is suggested by looking at where the nodes are and how much energy is left over. The results reveal that the suggested method uses less energy, lasts longer, and has a lower PLR than other methods.

Fei et al. [18] introduced an innovative Fuzzy C Means (FCM) clustering method for clustering and choosing CH, utilizing Moth-Flame Optimization (MFO), referred to as FCMMFO. FCMMFO is added to make the network work better. FCM makes clusters consume less energy, while MFO is utilized to determine the most effective CH in each cluster. In MFO, the moths fly around, and the flames correspond to pins that moths drop when they are hunting for something. So, each moth looks about the flame and shifts it if it sees a better position for it. With MFO, the moth will always discover the best response. This strategy can efficiently lower the amount of energy that networks use and even out the load on the sensor nodes. The experimental results show that FCMMFO uses less energy than other algorithms that are presently in use.

Krishnaswamy and Manvi [19] suggested a novel methodology for cluster formation utilizing fuzzy-based clustering (FBC) and Particle Swarm Optimization (PSO), referred to as FBCPSO. The fuzzy clustering model first separates the nodes into several fuzzy subsets based on where they are located. Using fuzzy clustering to generate clusters by considering things like geographic locations and the property of belonging to the cluster. When clusters are made, the nodes that are closest to the location and need less energy to talk to each other are put in the same cluster. The fitness function chooses the CH depending on energy requirement, how far it is from the CH and node, and the distance of node from the Base Station. This helps underwater wireless networks last longer. PSO is used to find out which nodes are CH. While considering the variables like the node's death rate, live nodes count, the quantity of messages received, and the amount of energy used, the suggested method works better.

SOSNET - A technique for the Smart Ocean underwater Sensor Network is suggested by Durrani et al. [20], to dynamically cluster nodes. SOSNET, which is based on the MFO (Moth Flame Optimizer), was created to identify the count of clusters for routing. MFO is a way to optimize things that is based on how moths travel toward light. The MFO method for VANETs (CAMONET) is based on MFO and puts automobiles into ad hoc networks (VANET). It thinks that the identical answer can be used for UWSN also. CAMONET basically finds out how many groups a network requires to work well. The metrics used to compare these are the range of transmission for nodes, the density of nodes, and the grid's size. The results reveal that SOSNET worked well when associated to other algorithms.

RBCMIC - The Region-Based Courier-Nodes Mobility with Incremental Cooperative Routing, an energy-efficient approach to cooperative routing was introduced by Yahya et al. [21]. A new depth-based incremental cooperative routing method is suggested; to make routing better. To make the network last longer, an incremental relaying method is added. The suggested approach takes advantage of the fact that wireless nodes make uses of incremental cooperative routing, which can transmit signals to many people at once. Courier nodes and RBCMIC go across the whole area and get information from the destination node. After that, the data goes to its place and each sensor

node uses less power to do it. Energy efficiency is improved by carefully reviewing and confirming the suggested plan using tried-and-true methods. This also makes the network last longer. The suggested method works better for network longevity, residual energy, throughput, and end-to-end latency.

The Distance- and Energy-constrained K-means Clustering Scheme (DEKCS) was suggested by Omeke et al. [22], to choose a cluster head and where to put the CH in the cluster. They also used how much battery power it had left to make the choice. For possible CHs, change the residual energy thresholds straight away to make sure the network runs out of power completely before it disconnects. A new clustering strategy that uses Autonomous Underwater Vehicles (AUVs) to collect data is introduced to make underwater networks last longer. This relies on how close the sensor nodes are to each other and how much energy they have left. Depending on how big the network is, the elbow approach enables you figure out the best number of clusters. This will allow the network to grow. Tests show that the proposed method is superior than the standard protocol with respect to the counts of dead nodes, alive nodes, received messages and utilized energy.

The IMCMR-UWSN technique was given by Mohan et al. [23], which is an improved version of the Met heuristics-based Clustering using Multichip Routing protocol for UWSNs, this method is used to find the most effective technique to get to a spot and pick CHs. The IMCMR-UWSN approach has two parts: CKHA (Chaotic Krill Head Algorithm) is used for grouping things together, and the SA-GSO (Self-Adaptive Glow Worm Swarm Optimization) is used, for routing things over multiple hops. The CKHA approach picks CHs and puts them into groups based on criteria like how much energy is left, how far apart the groups are, and how far apart the groups are from each other. The SA-GSO method makes fitness function that checks four things: distance, trust, delay, and energy left. The IMCMR-UWSN method is shown to greatly improve the UWSN's energy efficiency and lifespan. We did simulations to make sure the proposed method worked better. The acquired results were observed with respect to lifetime of the network, expenditure of overall energy, and the perfectly received packet's count.

The development of met heuristics-based clustering and a protocol for routing for UWSN (MCR-UWSN) was suggested by Subramanian et al. [24]. This technique is designed to identify the optimal CHs and the fastest routes to the BS. This method forms group and find the optimal CHs, the author utilizes the CEPOC (culture emperor penguin optimizer-based clustering). The EPOC method has produced a function of fitness that picks CHs and builds clusters based on diverse input criteria. The MCR-UWSN technique seeks to pick a decent collection of CHs and deliver data to the right place. The grasshopper optimization method is a multi-hop routing approach used for routing in UWSN. GOA is a new metaheuristic method that is based on SI. It is since mature grasshoppers in a group have a wider range and move quickly. The simulation results guarantee that the MCR-UWSN technique will work better than the current ones when it comes to saving energy.

Tian et al. [25] employed the principle of central control to enhance global network energy efficiency. CCCS, or Centralized Control-Based Clustering Scheme, is made just for UASNs- Underwater Acoustic Sensor Networks. This approach employs an adaptive clustering method that considers node density, sets up interconnected controllers within clusters, and improves the options between cluster and relay nodes to discover the optimal routes while using the least amount of energy. When compared to similar systems, these simulations show that this scheme successfully balances node energy utilization and outspreads the life of a UASN.

An adaptive energy-efficient clustering strategy for UWSNs, utilizing Multi-Dimensional Game Theory-based Clustering (MDGTC) was given by Xie et al. [26]. MDGTC makes a multidimensional clustering game model that makes it more likely that the best CH node will behave

as C-CH again when choosing Candidate Cluster Head (C-CH) nodes. Subsequently, to make the CH selection process even better, an adaptive CH competition mechanism is added by considering the energy utilization of both global and local nodes. For UWSNs, MDGTC is a good technique to control distributed energy. MDGTC can use energy more efficiently and help the network last a lot longer.

Shyamsundar and Harshavarthan [27] set out to study and learn more about seas and underwater ecosystems. The UWSN network links together several sensory devices to make a short-term communication network that transmits the data. The Gravitational Search Algorithm (CBR-GSA) is recommended for routing devices based on clustering. GSA gravitationally influenced intelligence makes it possible to transmit data in a way that uses minimal energy, which improves the stability and durability of networks. This makes it a good alternative to traditional routing methods. The recommended method has four steps: exploration, clustering, routing, and transmission. The suggested method also works effectively for both high and low node densities. The clustering algorithm presumes the equivalence of inertial and gravitational mass, hence reducing the need for exploration and mandating a minimal iterative exploited output. The underwater world is made up of several equal-sized cubic blocks. The average block density is used to select CHs. Residual and block energy are the two items that are looked at. Routing protocols have transformed the cluster-based strategy since they are faster and more efficient. For the next several broadcasts, the sensor nodes merely send a packet with extra power. Using GSA with other metaheuristic algorithms in a hybrid fashion can speed up convergence and make it work better.

3. PROPOSED METHODOLOGY

This study introduces an innovative Chaotic Bobcat Optimization Algorithm-Multi-Hop Data Transmission protocol for Underwater WSNs. The Weight Clustering Approach is a new way to choose CHs and build clusters that is used by the CBOA-MHDT protocol. CH is chosen depending on how much energy the member nodes use while sending data, how much energy the cluster head and its closest node have left, and how far away the CH is from the base station where the data transmission starts. The hunting tactic of bobcats when they assault their prey and chase them is where BOA comes from. The chaotic operator is very good at exploring and exploiting, and they balance these two things while searching for a good CH solution for routing. After choosing CH, multi-hop data transmission is used to make UWSNs more energy-efficient and consistent. The suggested model is a centralized routing system that consumes less energy and determines the optimum path in UWSN. The suggested model's overall process is shown in Figure 2.

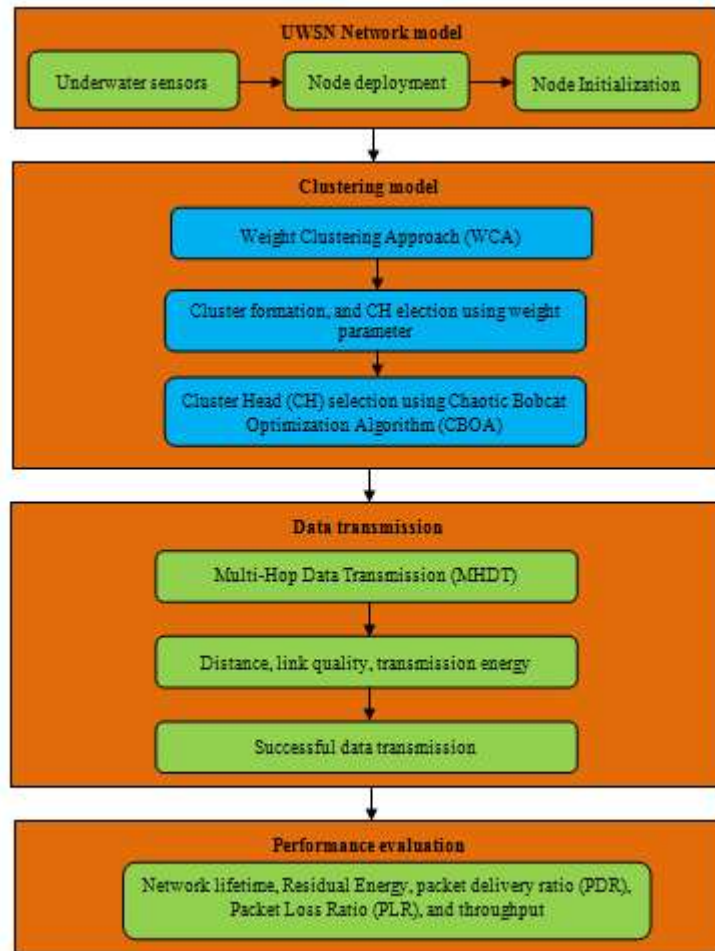


Figure 2: Overall Process of Proposed Model

3.1 System Model

Clustering based UWSN model is illustrated in Figure 3. The UWSN has a base station, sensors that work underwater, and a sink node. The nodes sent data to the sink, which are close to SN. A different node made the clusters. There are sensors on the surface that provide data to the SN [28]. Then the data is transmitted to the nearby BS [29]. There are two types of transceivers in the SN nodes: one that lets the sensors talk to each other and one that can send and receive radio signals. The other kind is a radio transceiver that can send and receive radio waves to talk to the BSS. Underwater nodes feature built-in audio transceivers that let them send messages to neighbouring nodes. The current moves the UWSN at a speed of around 1 to 3 meters per second. The radio and acoustic channels each have their own separate models.

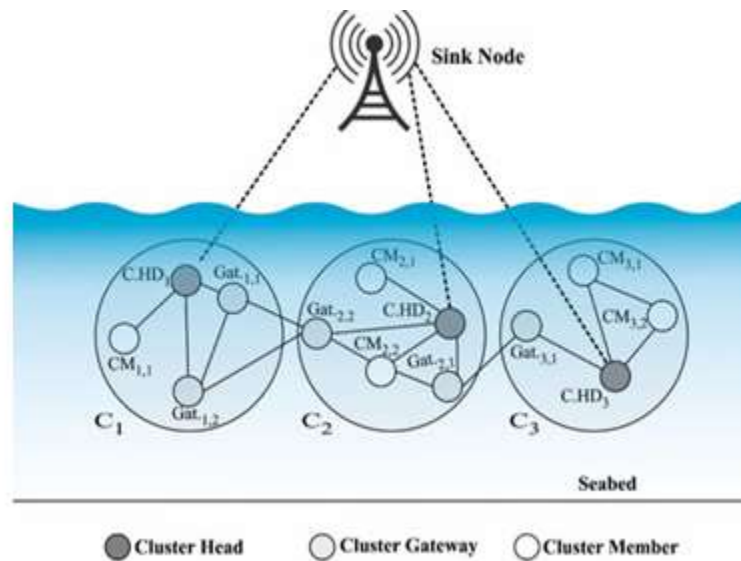


Figure 3: Overview of Cluster Based UWSN

There are N dynamic nodes in the system, and during an $L \times L \times L$ process, they are spread out randomly and not too densely. The data source is data that can feel water. An underwater sensor was used to collect the data. The parameters that have been found include the flow, pressure, and temperature of the current. An acoustic modem was added to the underwater sensor so that it could talk to other nodes in the water [23-24]. To talk to the BS on the surface landmass, SN uses both Radio Frequency modem and an acoustic modem. The RF modem transmits data through the base station, while the SN's acoustic modem collects data from sensors that are underwater. The BS can move through a sensing field and get information from sensor nodes in the field, but it can't transfer data very far. Each sensor node uses less electricity because it doesn't need as many relays to transmit the sensor's message to the BS. It is considered that the state of the networks is linked to the networks. The topology varies quickly since you may move the underwater sensors around according on how fast the water currents are, and it is actually among 1 and 3 m/s [24-25]. The following shows how the network condition looklike:

- In a primary situation, the nodes know where they are and where the SN is.
- The CH and the CM/relay are made by the node.
- The CH moves around to save energy among the sensors.

Radio waves and acoustic waves are significantly different from each other. Because of this, a Wireless Sensor Network is not an excellent option for broadcasting networks that work underwater. The power consumption method is used as an underwater acoustic channel, for this investigation. It takes a lot of energy to send k bits of data through a reservoir d at a speed of R:

$$E_{nTxk,d} = k \times E_{nElec} + kR E_{nTx} \quad (1)$$

The energy used to route it is denoted as E_{nTx} and the energy employed to deliver one bit of information is represented by E_{nElec} . The receiver radio consumes energy as defined by equation (2) to get k bits of information.

$$E_{nRxk} = k E_{nR} \quad (2)$$

The amount of energy used by the receiver circuitry for each bit is denoted by E_{nR} . The power use is calculated below to combine the data with k bits,

$$E_{DAk} = k \times E_{DA} \quad (3)$$

E_{DA} , is the amount of energy used to break one bit of data, which is 5 nJ/bit.

A common and effective way to get rid of duplicate data, make data smaller, and use less energy, is nothing but Fusing data. Sensor nodes in UWSNs can gather data without much care. When excess information is sent to the SN, it wastes energy that isn't needed, which makes the node die early and shortens the network's life. CH, conversely, sends data to the SN, by combining the data. This can save a lot of energy. The node moves because of the water stream, but you may find it by looking at how it moves randomly over time.

3.2 WCA process

In UWSNs, clustering is a way to route data that saves energy by putting sensor nodes into categories (clusters) and assigning each group a CH, who manages the gathering and transferring data to the sink. The weight clustering technique uses three measures—node degree is denoted by NDE_i , residual energy is given by RE_i and D_i shows the distance to define the cluster architecture and the CH that was used. The weight W was calculated for each node as follows:

$$W_i = w_1 * RE_i + w_2 * D_i + w_3 * NDE_i \quad (4)$$

The coefficients of the model are denoted as w_1 , w_2 , and w_3 and it is stated by the equation (5)

$$w_1 + w_2 + w_3 = 1 \quad (5)$$

Based on the total energy E_n , the transmission energy and reception energy are used to figure out the overall residual energy.

$$RE = E_n - E_{nTxk,d} + E_{nRxk} \quad (6)$$

Also, you may find the average distance between neighbouring nodes that are single-hop neighbours like this:

$$D = \frac{1}{N} \sum_{i=1}^N B_{i,d_i, n_bj} \quad (7)$$

The distance from the SN to the j^{th} SN is denoted by d_i, n_{bj} , NDE is the number of nearby nodes that have a transmitting radius according to equation (8).

$$NDE = N_x \quad (8)$$

The node's transmission range is given by TR , further $N_x = n_{y,dx,ny} < TR$, $x \neq y$, & $d_{x,y}$ gives the distance midst the two nodes and is denoted as n_x and n_y .

3.3 Chaotic Bobcat Optimization Algorithm (CBOA) Based Cluster Head Selection

Chaotic Bobcat Optimization Algorithm that mimics how bobcats act in the wild to find the best CH in the UWSN. The basic idea behind BOA stems from the way bobcats pursue their prey and then chase it, which is meant to help them choose the best CH for grouping. This hunting method comprises two parts: (i) tracking the prey to become close to it, and (ii) catching the prey by chasing. The CBOA method is a population-based optimizer that can identify the optimal CH by doing the procedure repeatedly. In the process, each bobcat is both the CH and the CM. The CH solution space sets the values for the decision variables, such as node (NDE_i), residual energy (RES_i), and distance (D_i) so, for the clustering, each bobcat's position could be a possible CH solution. This is a vector in arithmetic, and each part of the vector is a variable that may be changed. The algorithm's population includes bobcats, which may be shown mathematically using a matrix as shown in Equation (9). Equation (10) [30] randomly choose where bobcats start in the space where they are trying to solve an issue.

$$X = X_1 : X_i : X_{NN} \times m \quad (9)$$

$$x_{i,d} = l_{bd} + r \cdot (u_{bd} - l_{bd}) \quad (10)$$

The BOA population matrix is called X , and the i^{th} bobcat is called X_i (CH solution). The d^{th} dimension of the CH search space (decision variable) is represented by $x_{i,d}$. M is the quantity of option variable's, and the quantity of bobcats that are CHs in the clustering model is N . Some of these

include node degree (NDE_i) residual energy (RES_i) and distance (D_i) the number is a random number between 0 and 1. The lower lbd and upper ubd bounds of the d^{th} decision variable are designated. Equation (11) shows the calculated values for the objective function, which can be shown as a vector [30].

$$F = F_1 : F_i : F_{NN} \times 1 = F_{X1} : F_{Xi} : F_{XNN} \times 1 \quad (11)$$

F is the vector of the objective function that was tested, and F_i is the objective function that was employed to test the i^{th} bobcat. These assessed factors for the CH solution's objective function are beneficial for determining the efficacy of each CBOA member in generating potential solutions. The greatest and worst values for the objective function are likewise the best and worst values for the CBOA. The placements of the bobcats in the problem-solving environment shift each time the CBOA design process is conducted. This means that the candidate solutions' parameters and the target function have also changed. So, every time the process starts afresh, the best CBOA member should change depending on how the objective function's outcomes compare [30].

Phase 1: Tracking and moving towards prey (exploration phase): The first phase of CBOA is to change the positions of the humans in the CH selection space based on a simulation regarding the way bobcats hunt by following their prey and moving toward it. When a bobcat approaches toward its prey, it disturbs the CM position of the people in the CH selection space a lot. This makes it easy for CBOA to work together on the worldwide search. In CBOA, the target position for each bobcat is the location of the rest of the population that has greater scores for the objective function [30]. To find the candidate prey set for each bobcat, Eq -14 is used,

$$C_{Pi} = X_k : F_k < F_i \ \& \ k \neq i, \text{ Where } i=1,2,\dots,N \text{ and } k \in 1,2,\dots,N \quad (12)$$

The set of possible places for the i^{th} bobcat to find prey is called C_{Pi} , and population member with an improved goal function value compared to the i^{th} bobcat is expressed by X_k , and based on equation (4), objective function is represented as F_k . The CBOA design proposes that, one of these preys were picked randomly by each bobcat and attacks it. Using Equation (13) [30], for each CBOA member, a new CH position is determined based on how the bobcat's place changes as it moves toward its prey in this technique. It takes the place of the old place of the associated member, if this new CH position makes the goal function's value better.

$$x_{i,j}P1 = x_{i,j} + 1 - 2r_{i,j} \cdot SP_{i,j} - l_{i,j} \cdot X_{i,j} \quad (13)$$

$$X_i = X_iP1, F_iP1 \leq F_i X_i, \text{ else} \quad (14)$$

In this case, $SP_{i,j}$ is the prey that the i^{th} bobcat chose, and $SP_{i,j}$ displays its j^{th} dimension? The new CH location for the i^{th} bobcat is X_iP1 , and the j^{th} dimension is $x_{i,j}P1$. For the i^{th} bobcat, the value of the goal function is indicted by F_iP1 . For the i^{th} bobcat, random values from the interval [0, 1] are given by $r_{i,j}$, and $l_{i,j}$ is either 1 or 2.

Phase 2: Chasing to catch prey (exploitation phase): In the next stage of CBOA, the positions of the people in the CH selection space alter dependent on how the bobcat and prey pursue each other while hunting. This is how the bobcat catches its prey: it chases it. While modelling how a bobcat moves while seeking and catching prey, it creates small changes to where people are in the CH seeking space. This gives CBOA additional control over local search. In the structure of CBOA, Equation (15) assists to find a new spot for each member to be close to the hunting area [30]. This is because the bobcat's position changes while it is chasing. If this new place makes the objective function (Weight (W)), more valuable, the related members will move there as shown in Equation (16).

$$x_{i,j}P2 = x_{i,j} + 1 - 2r_{i,j} + t \cdot x_{i,j} \quad (15)$$

$$X_i = X_iP2, F_iP2 \leq F_i X_i, \text{ else} \quad (16)$$

In the BOA exploitation phase, X_i^{P2} , shows the latest location of the i th bobcat. The j th dimension of the i th bobcat is shown by x_{ij}^{P2} , and the value of the related objective function is shown by F_i^{P2} . The variables r_{ij} are random numbers between 0 and 1, and t is the number of times the process is run. In equations (10), (13), and (15), the chaotic map makes a random number r . A chaotic state is one that is very sensitive to its initial settings and is hard to predict. To makes the CH solution better, this characteristic protects from becoming stuck in local optima. During the individual phase, the chaotic map was used to get the best random number that was close to where they were at the time. To explore this, the concept of connecting different suggestions uses the objective function in the community stage. The most popular type of chaotic behavioural map utilized to create random numbers in the CH selection process is the Chebyshev map [30]. Equations (17), (19), and (20) delineate the novel process of $x_{i,j}^{P1}$, $x_{i,j}^{P2}$.

$$x_{i,j}^{P1} = x_{i,j} + 1 - 2c_{mi,j} \cdot S_{Pi,j} - l_{i,j} \cdot X_{i,j} \quad (18)$$

$$X_i = X_i^{P1}, F_i^{P1} \leq F_i X_i, \text{ else} \quad (19)$$

$$x_{i,j}^{P2} = x_{i,j} + 1 - 2c_{mi,j} \cdot l_{i,j} + t \cdot x_{i,j} \quad (19)$$

In the Chebyshev chaotic map [31], $c_{mi,j}$ represents the random number generated for the i th bobcat in its j th dimension. The CBOA system works by changing the CH positions of all the bobcats in the search area according to the exploration and exploitation phases. Then, the algorithm runs again, this time with fresh values for the target function and the CH position of the bobcats. Updating bobcats keeps going until the algorithm's last iteration, as indicated in Equations (12)–(20). Every time the iteration is changed and saved, the best CH solution is found. Once the technique is fully implemented, the finest choice The CH solution that is found in each iteration of the algorithm is used as the CBOA solution to choose the CH in the clustering model. Figure 4 displays the flowchart, and Algorithm 1 demonstrates the pseudo code.

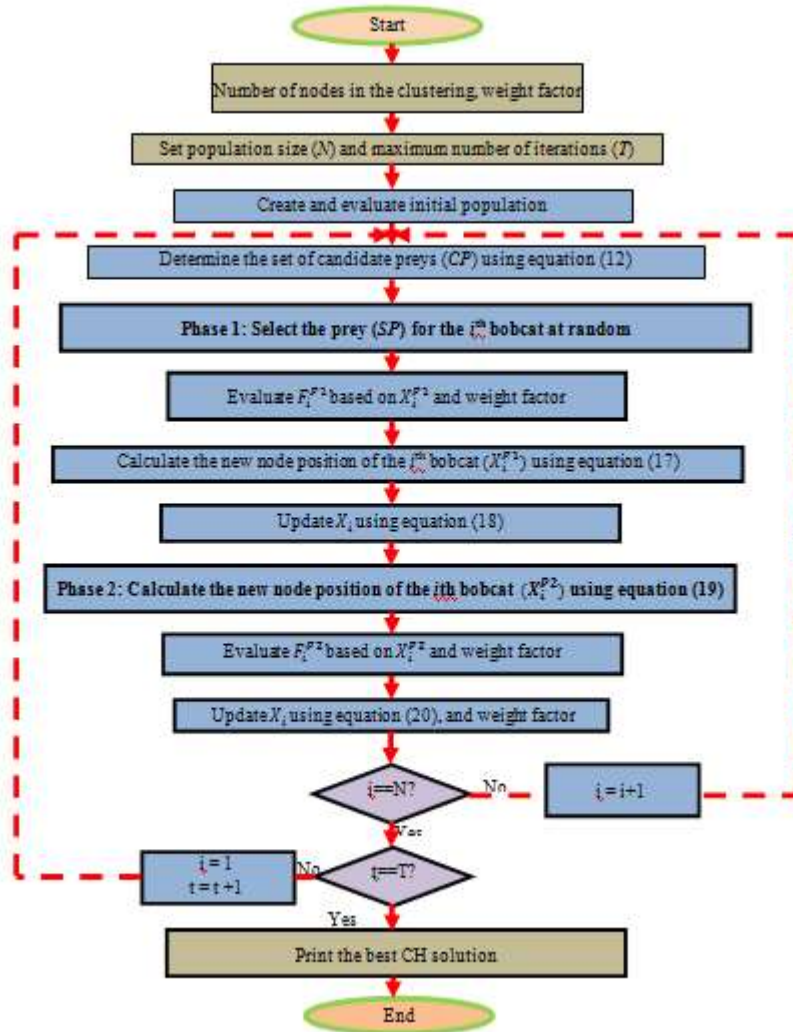


Figure 4: Flowchart of CBOA

ALGORITHM 1. CHAOTIC BOBCAT OPTIMIZATION ALGORITHM (CBOA)

Start

1. Input CH selection problem formulation based on the clusters, weight factor, and constraints
2. Set population size(N) based on number of nodes, CH, CM, and iterations(T)
3. Generate the initial population matrix at random using equation (10)
4. Evaluate the weight based on the equation (4)
5. For t=1 to T
6. For i = 1 to N

Phase 1: Tracking and moving towards prey (exploration phase)

7. Determine the preys set (optimal CH solution) for the ithCBOA member using equation (12)
8. Select the termite mounds for the ithCBOA member at random.

9. Calculate new node position of i^{th} member as optimal CH using equation (17)
10. Update i^{th} BOA member using equation (18)
- Phase 2: Chasing to catch prey (exploitation phase)***
11. Calculate new node position of i^{th} BOA member of CH using equation (19)
12. Update i^{th} member as selected CH using equation (20)
13. End for i
14. Save the best candidate solution as selected CH so far
15. End for t
16. Output the best quasi-CH solution obtained with the CBOA

End

Every UWSN delivers a "Hello Message" to all its neighbouring nodes together with its fitness. When the SNs get a "Hello" message, they check to see how fit it is compared to their own. When the UWSN gets a "Hello" message, it adds its own entries to a neighbour table and updates it. The UWSN then puts the neighbour table in order of fitness values that are lower. The CH node that is nearest to the base station, hence it is the best choice for this job. To form a cluster, the UWSN with the highest fitness value is chosen as the CH and sends a message to nearby nodes. This process occurs regardless of whether there are single nodes or multiple nodes within the base station's range. Nodes with lower fitness values are instructed to join the cluster, which helps them identify their CH. To deliver data to the destination, nodes use an intermediate UWSN, resulting in multiple hops. Following the CH is chosen; the other nodes of UWSN will either become shallow nodes or members of the cluster. This decision depends on a fitness function. People in the cluster need to watch the CH's actions and move to different spots when they need to. The CH takes location data from all UWSNs and utilizes it to maintain track of the shape of the cluster.

After CH selection, it sends a message to everyone in its communication range about how to establish a cluster. Within this phase, each CM node listens to what is going on around it to collect information from other nodes about how to set up a cluster. These notifications come with the ID and extra energy information for the CH. The sensor node will select clusters according to the following, based on the cluster formation messages it gets. The sensor node will only join the CH that goes with the message it gets about forming a cluster. If a sensor node receives a multiple cluster formation notification, it checks them out and joins the cluster with the CH that has the most outstanding energy. The sensor node is unable to communicate to any other node if it does not get the message to form a cluster. In that instance, the node falls to sleep and waits for the message to be sent. After the cluster is set up, the CH transmits a control message which puts the CMs to sleep, allows them to experience their environment, and collects information. The CH transmits a message to wake up the CMs during the routing. The CMs then send the information they have gathered to the CH, which executes fusion.

3.4 Multi-Hop Data Transmission

The suggested solution uses multi-hop data transmission to make UWSNs more reliable and use less energy. The sensed data is routed through numerous intermediate nodes instead of going straight to the sink from the sensor nodes. This approach minimizes transmission energy, extends the communication range, and reduces packet loss caused by underwater attenuation and noise. Each underwater sensor node monitors its surrounding environment and collects the required data. Nodes identify nearby neighbors using beacon messages or predefined communication ranges. Then, the information is transmitted to the nearest node depending on routing metrics that have already been configured. The chosen neighbour node obtains the data that was sensed and transmits it on to the node that sinks through a various hops. Intermediate nodes act as relays, ensuring reliable packet

forwarding while minimizing energy consumption. Finally, the transmitted data reaches the sink node after multiple hops, where it is aggregated and processed for further applications. To prevent rapid energy reduction, forwarding tasks are distributed among multiple nodes, thereby extending the overall network lifetime. Nodes send short beacons (ID, position, residual energy) and build a neighbor list egi . For each neighbor $j \in Negi$, compute distance dij , estimated link quality η_{ij} , and a rough transmit energy estimate $Entx(L, dij)$, and node depth. Calculate a simple score for each neighbor by equation (21),

$$score_{ij} = w_1 \cdot d_{ij} + w_2 \cdot \eta_{ij} + w_3 \cdot E_{nj} + w_4 \cdot (1 - \frac{Dep_j}{MDep}) \quad (21)$$

Weights are represented as w_1, w_2, w_3, w_4 that add up to one, the maximum energy is given as E_{nmax} , the depth of node j is given by Dep_j , the ocean's maximum depth is denoted as $MDep$. Neighbors are chosen with the highest score that also satisfies minimal energy and link thresholds. Transmit the packet to the selected next hop. If an ACK arrives, continue; on failure retry a few times or pick the next best neighbor. Subtract transmission/receive energy from nodes' residual energy and penalize low-energy nodes in future scoring to balance load.

4. RESULTS AND DISCUSSION

In this section, proposed model, and existing methods have been implemented using the MATLAB R2021b. The outcomes are noticed by comparing them to other approaches like EECRP, FCMMFO, FBCPSO, and CSRO-MHDT. Table 1 explains the parameters that help in the localization simulation. The proposed algorithm's performance is compared to that of current clustering algorithms by looking at things like network lifetime, residual energy, packet delivery ratio, packet loss ratio, and throughput.

Table 1: Simulation Parameters of UWSN model

| Parameter | Value |
|---------------------------------------------|-------------------|
| Localization Domain Area | 1,000 m × 1,000 m |
| Maximum Depth | 500 m |
| Beacon Percentage | 3%–5%–8%–12% |
| Number of nodes | 500 |
| Speed of Sound | 1500 m/s |
| Error in Speed of Sound | 0.07 m/s |
| Error in Depth | 0.1 m |
| Standard Derivation of Time stamping | 15 μs |
| Carrier frequency | 15 kHz |
| Bandwidth | 10 kHz |
| Packet Size | 512 bytes |
| Data Rate | 10 kbps |
| Transmit Energy | 35 W |
| Receive Energy | 0.3 W |
| Number of rounds | 1000 |
| Node deployment | Clustering model |
| Node Mobility model | Random Waypoint |

4.1 Performance Metrics

The performance analysis in clustering methods is measured by Lifetime of the Network, throughput, PLR, PDR, Residual Energy.

Network lifetime: The network's lifetime is the time from as it begins and when the final node in it

dies. It is measure by how many nodes are alive and how many are dead.

Alive nodes: The network nodes that still have enough power to send packets.

Dead nodes: The nodes that do not have enough energy to send data.

FND (First Node Dies): The time when the initial node with sensors runs out of power (i.e., number of alive nodes drops from N to N-1). Often used to indicate when the network first becomes partially degraded.

HND (Half Nodes Die): Time when 50% of nodes are dead (i.e., when the count of dead nodes $\geq N/2$).

LND (Last Node Dies): Time when the last node dies (entire network disconnected / no nodes can sense/transmit).

Residual Energy: It is the energy's quantity that is still in the network at a certain round. It is measured in Joules.

PDR: This ratio indicates how many of the packets were sent to their destination. To figure out PDR, do the following:

$$PDR = \frac{\text{Number of packets Received}}{\text{Total number of packets sent}} * 100 \quad (22)$$

PLR: This Ratio indicates what proportion of sent packets doesn't reach their destination. The PLR is figured out as follows:

$$PLR = \frac{\text{Number of pacets Lost}}{\text{Total number of packets sent}} * 100 \quad (23)$$

PDR and PLR are inversely proportional: a higher PDR means a lower PLR, indicating better network reliability and performance.

Throughput: It is the data's quantity that a network can send and receive in a certain length of time. The throughput is figured out as follows:

$$\text{Throughput} = \frac{\text{Total Data Received(bits)}}{\text{Total simulation time(Seconds)}} * 100 \quad (24)$$

Important measures were used here that involvethroughput, PLR, PDR, and residual energyto compare the proposed framework to existing approaches. Table 2 makes it quite clear what is going on. Proposed system has increased RE, PDR, reduced PLR, and increased throughput of 595 J, 88.23%, 11.77%, and40.55 Kbps.

Table 2: Results Comparison of Clustering Protocols

| Number of rounds/ Methods | Residual Energy (RE) (Joules(J)) | | | | | |
|---------------------------|----------------------------------|-------|-------|-------|-------|---------|
| | 200 | 400 | 600 | 8000 | 1000 | Average |
| FCMMFO | 1088 | 782 | 469 | 204 | 96 | 527.80 |
| FBCPSO | 1454 | 1078 | 685 | 421 | 209 | 769.40 |
| EECRP | 1771 | 1254 | 864 | 547 | 304 | 948.00 |
| CBR-GSA | 1885 | 1387 | 952 | 615 | 351 | 1038.00 |
| CSRO-MHDT | 2089 | 1576 | 1058 | 682 | 418 | 1164.60 |
| CBOA-MHDT | 2594 | 1845 | 1236 | 839 | 595 | 1421.80 |
| Number of rounds/ Methods | PDR (%) | | | | | |
| | 200 | 400 | 600 | 8000 | 1000 | Average |
| FCMMFO | 83.44 | 81.85 | 80.64 | 79.59 | 78.68 | 80.84 |
| FBCPSO | 85.18 | 83.49 | 82.23 | 81.18 | 80.06 | 82.43 |
| EECRP | 87.72 | 85.64 | 84.41 | 83.37 | 81.44 | 84.52 |
| CBR-GSA | 88.65 | 86.72 | 85.69 | 84.06 | 82.58 | 85.54 |
| CSRO-MHDT | 89.56 | 87.81 | 86.74 | 85.45 | 84.19 | 86.75 |
| CBOA-MHDT | 91.37 | 89.59 | 88.26 | 86.58 | 85.33 | 88.23 |
| Number of rounds/ Methods | PLR (%) | | | | | |

| | 200 | 400 | 600 | 8000 | 1000 | Average |
|----------------------------------|-------------------------|------------|------------|-------------|-------------|----------------|
| FCMMFO | 16.56 | 18.15 | 19.36 | 20.41 | 21.32 | 19.16 |
| FBCPSO | 14.82 | 16.51 | 17.77 | 18.82 | 19.94 | 17.57 |
| EECRP | 12.28 | 14.36 | 15.59 | 16.63 | 18.56 | 15.48 |
| CBR-GSA | 11.35 | 13.28 | 14.31 | 15.94 | 17.42 | 14.46 |
| CSRO-MHDT | 10.44 | 12.19 | 13.26 | 14.55 | 15.81 | 13.25 |
| CBOA-MHDT | 8.63 | 10.41 | 11.74 | 13.42 | 14.67 | 11.77 |
| Number of rounds/ Methods | Throughput(Kbps) | | | | | |
| | 200 | 400 | 600 | 8000 | 1000 | Average |
| FCMMFO | 22.54 | 25.32 | 28.44 | 32.05 | 35.26 | 28.72 |
| FBCPSO | 25.39 | 29.28 | 32.49 | 35.82 | 38.69 | 32.34 |
| EECRP | 27.76 | 32.47 | 35.31 | 38.16 | 42.33 | 35.21 |
| CBR-GSA | 28.79 | 33.58 | 36.37 | 39.49 | 43.72 | 36.39 |
| CSRO-MHDT | 29.82 | 34.73 | 37.58 | 41.74 | 45.87 | 37.95 |
| CBOA-MHDT | 32.76 | 37.69 | 40.83 | 43.68 | 47.79 | 40.55 |

Table 3, proposed system has increased network lifetime of 498 rounds, 864 rounds, and 983 rounds for FND, HND, and LND.

Table 3: Network Lifetime Comparison of Clustering Protocols

| METHODS | FND | HND | LND |
|------------------|------------|------------|------------|
| FCMMFO | 358 | 727 | 901 |
| FBCPSO | 401 | 745 | 919 |
| EECRP | 437 | 751 | 937 |
| CBR-GSA | 458 | 789 | 951 |
| CSRO-MHDT | 473 | 836 | 972 |
| CBOA-MHDT | 498 | 864 | 983 |

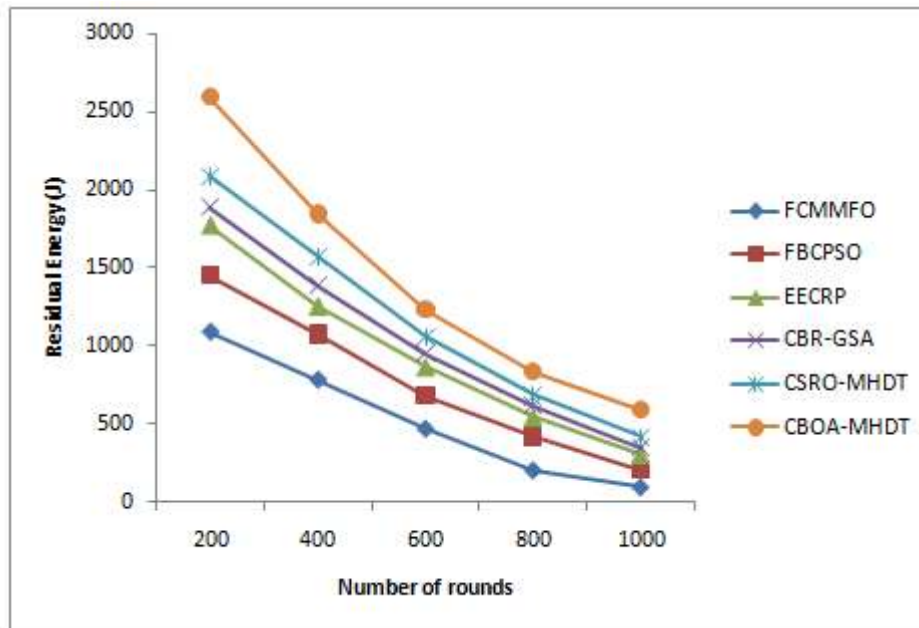


Figure 5: Residual Energy Comparison of Clustering Methods

Figure 5 shows how the residual energy of different clustering algorithms changes as the number of rounds goes from 200 to 1000. The more rounds indicates that the proposed system consistently achieves the highest average RE, reaching 1421.8 J, whereas other methods such as FCMMFO (527.80 J), FBCPSO (769.40 J), EECRP (948 J), CBR-GSA (1038 J), and CSRO-MHDT (1164.6J) demonstrate lower RE values. Furthermore 200 rounds, the proposed system also achieves the highest RE of 595 J, while the RE values of FCMMFO, FBCPSO, EECRP, CBR-GSA, and CSRO-MHDT are 96 J, 209 J, 304 J, 351 J, and 418 J, respectively (refer Table 2).

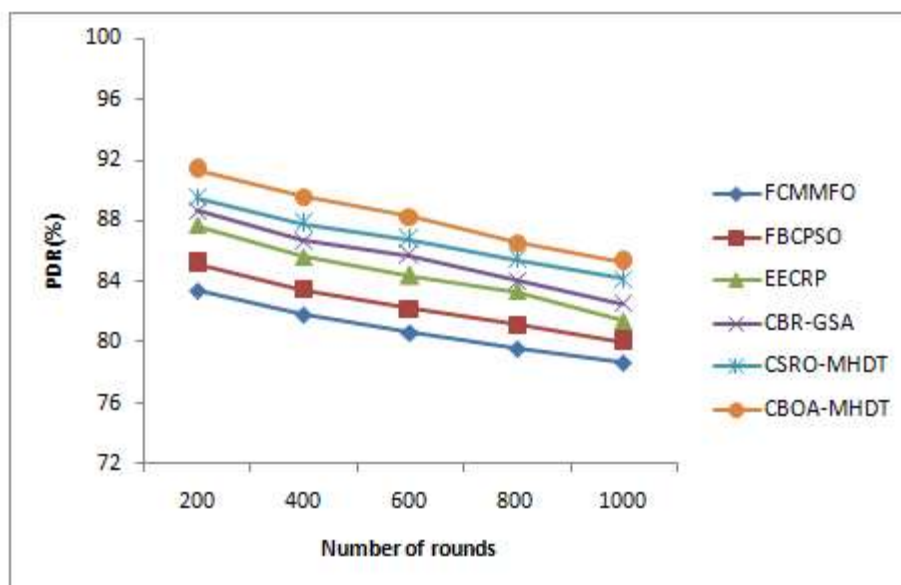


Figure 6: PDR Comparison of Clustering Methods

Figure 6 shows how the PDR of different clustering algorithms changes as the total amount of rounds goes from 200 to 1000. The proposed system constantly gets the best average PDR as the

overall number of rounds goes up, reaching 88.23%, whereas other methods such as FCMMFO (80.84%), FBCPSO (82.43%), EECRP (84.52%), CBR-GSA (85.54%), and CSRO-MHDT (86.75%) demonstrate lower PDR values. Furthermore 200 rounds, the proposed system also achieves the highest PDR of 91.37%, while the PDR values of FCMMFO, FBCPSO, EECRP, CBR-GSA, and CSRO-MHDT are 83.44%, 85.18%, 87.72%, 88.65%, and 89.56%, respectively(refer Table 2).

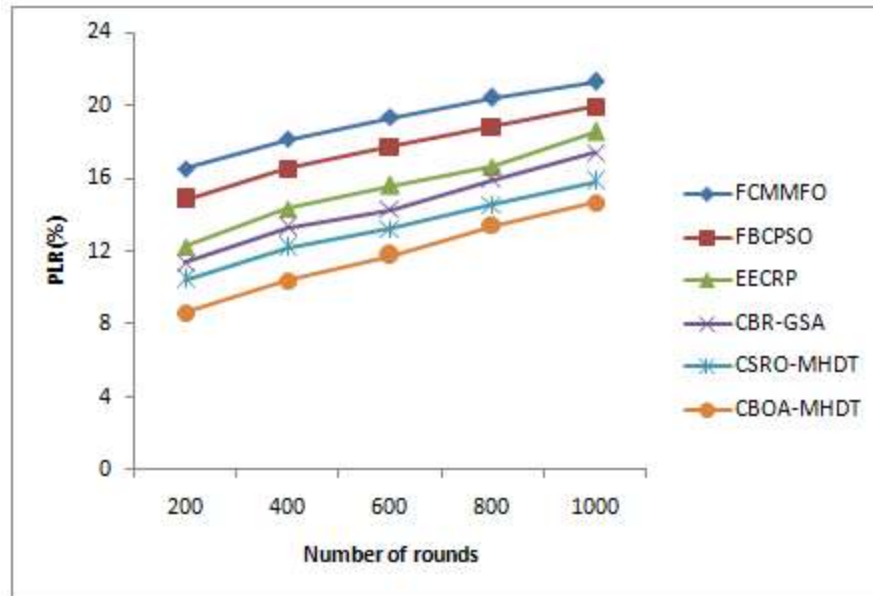


Figure 7: PLR Comparison of Clustering Methods

Figure 7 shows a PLR comparison between clustering algorithms based on the number of rounds, which might be between 200 and 1000. As the number of turns goes up, the proposed system consistently achieves the lowest average PLR, reaching 11.77%, whereas other methods such as FCMMFO (19.16%), FBCPSO (17.57%), EECRP (15.48%), CBR-GSA (14.46%), and CSRO-MHDT (13.25%) demonstrate increased PLR values. Furthermore 200 rounds, the proposed system also achieves the lowest PLR of 8.63%, while the PLR values of FCMMFO, FBCPSO, EECRP, CBR-GSA, and CSRO-MHDT are 16.56%, 14.82%, 12.28%, 11.35%, and 10.44%, respectively(refer Table 2).

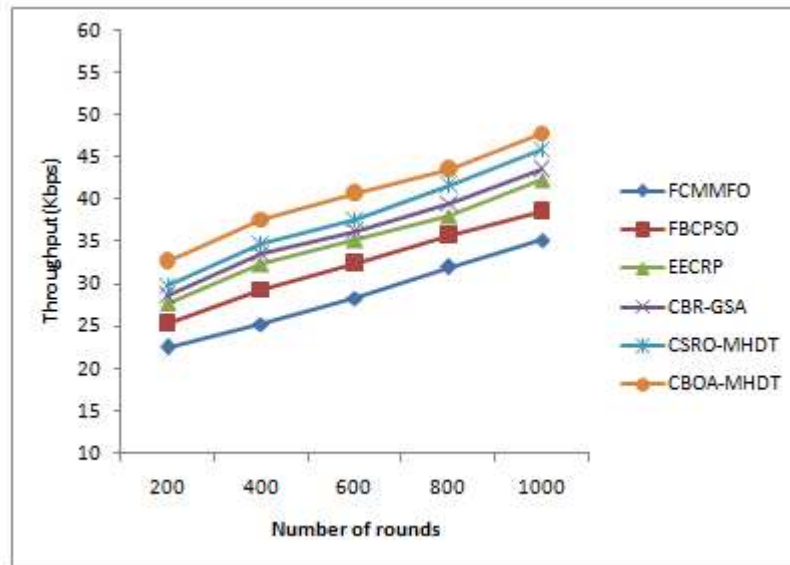


Figure 8: Throughput Comparison of Clustering Methods

Figure 8 shows how the throughput of clustering techniques changes when the number of rounds goes from 200 to 1000. As the count of number of turns goes up, the proposed system consistently achieves the increased throughput of 47.79Kbps, while other methods such as FCMMFO(35.26 Kbps), FBCPSO(38.69 Kbps), EECRP (42.33 Kbps), CBR-GSA (43.72 Kbps), and CSRO-MHDT (45.87 Kbps) exhibit lowest throughput. In terms of the average performance, the proposed system gives the highest throughput of 40.55 Kbps, whereas the average throughput of FCMMFO, FBCPSO, EECRP, CBR-GSA, and CSRO-MHDT are 28.72Kbps, 32.34Kbps, 35.21Kbps, 36.39 Kbps, and 37.95Kbps, respectively (refer Table 2).

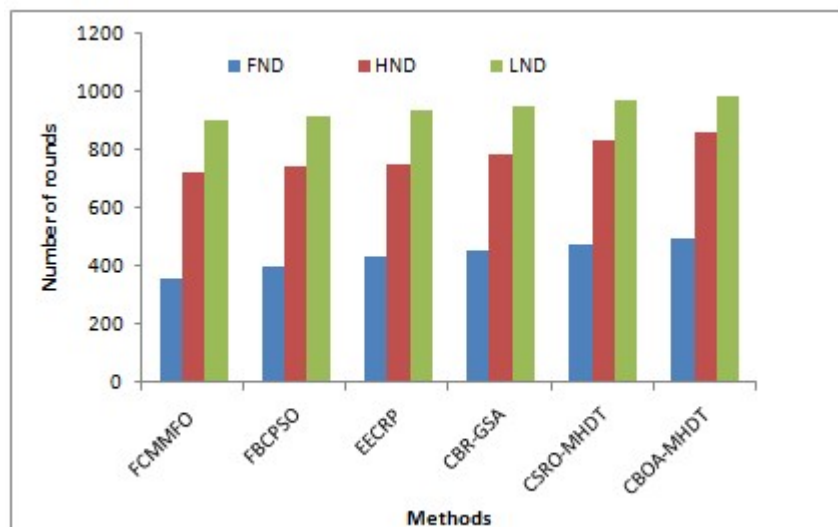


Figure 9: Network Lifetime Comparison of Clustering Methods

Figure 9 shows how long the network lasts for different clustering techniques, such as LND, HND and FND. The proposed technique improves the network’s lifespan result to 498 rounds and 864 rounds and 983 rounds for FND, HND, and LND. FCMMFO, FBCPSO, EECRP, CBR-GSA, and CSRO-MHDT have lowest FND results of 358 rounds, 401 rounds, 437 rounds, 458 rounds, and 473 rounds. FCMMFO, FBCPSO, EECRP, CBR-GSA, and CSRO-MHDT have lowest HND results

of 727 rounds, 745 rounds, 751 rounds, 789 rounds, and 836 rounds. FCMMFO, FBCPSO, EECRP, CBR-GSA, and CSRO-MHDT have lowest LND results of 901 rounds, 919 rounds, 937 rounds, 951 rounds, and 972 rounds (refer table 3).

5. CONCLUSION AND FUTURE WORK

The work introduces a new Chaotic Bobcat Optimization Algorithm for Multi-Hop Data Transmission for the UWSN. The suggested approach comprises four steps: clustering, choosing a CH, routing, and sending data. Clustering splits the network into groups based on how they are arranged horizontally. In each iteration, according to the distance, node degree, and the nodes' remaining energy level, CH is determined. CH, CBOA is introduced, which mimics how bobcats act in the wild to help choose the best option in the UWSN. CBOA comes from the way bobcats pursue their prey and how they chase each other to have the best CH selection. The CH sends the cluster formation details to everyone in its communication range, after CH selection. In routing, the CH transmits a wake-up message to the Cluster Member, who then sends the data it has collected to the CH. MHDT is used to make UWSNs more reliable and use less energy. The data is sent through several intermediate nodes, rather than having sensor nodes send data straight to the sink. MHDT uses less energy to send messages, enhances the range of communication, speeds up packet transmission, and lowers the number of packets that are lost. We evaluated metrics like throughput, PLR, PDR, Residual Energy and Network Lifetime to see how well the approaches worked across 1000 cycles. It has been proven that adding data aggregation methods to the design process can make systems even more energy efficient in the future.

Acknowledgement: Nil

Conflicts of Interest: The authors report there are no competing interests to declare.

Funding Sources: No funding was received to assist with the preparation of this manuscript.

Data Availability Statement: Nil

REFERENCES

- [1] Fattah, S., Gani, A., Ahmedy, I., Idris, M.Y.I. and Targio Hashem, I.A., 2020. A survey on underwater wireless sensor networks: Requirements, taxonomy, recent advances, and open research challenges. *Sensors*, 20(18), pp.1-30.
- [2] Khasawneh, A., Latiff, M.S.B.A., Kaiwartya, O. and Chizari, H., 2018. A reliable energy-efficient pressure-based routing protocol for underwater wireless sensor network. *Wireless Networks*, 24(6), pp.2061-2075.
- [3] Ahmad, A., Ahmed, S., Imran, M., Alam, M., Niaz, I.A. and Javaid, N., 2017. On energy efficiency in underwater wireless sensor networks with cooperative routing. *Annals of Telecommunications*, 72(3), pp.173-188.
- [4] Watt, A.J., Phillips, M.R., Campbell, C.A., Wells, I. and Hole, S., 2019. Wireless Sensor Networks for monitoring underwater sediment transport. *Science of the Total Environment*, 667, pp.160-165.
- [5] Khisa, S. and Moh, S., 2021. Survey on recent advancements in energy-efficient routing protocols for underwater wireless sensor networks. *IEEE Access*, 9, pp.55045-55062.
- [6] Lee, S., Jeong, Y., Moon, E. and Kim, D., 2017. An efficient MOP decision method using hop interval for RPL-based underwater sensor networks. *Wireless Personal Communications*, 93(4), pp.1027-1041.
- [7] Guo, W., Zhu, M., Yang, B., Wu, Y. and Li, X., 2023. Design of a self-organizing routing protocol for underwater wireless sensor networks based on location and energy information.

- Journal of Marine Science and Engineering, 11(8), pp.1-18.
- [8] Toor, A.S. and Jain, A.K., 2019. Energy aware cluster based multi-hop energy efficient routing protocol using multiple mobile nodes (MEACBM) in wireless sensor networks. *AEU-International Journal of Electronics and Communications*, 102, pp.41-53.
- [9] Singh, M.K., Amin, S.I. and Choudhary, A., 2021. Genetic algorithm based sink mobility for energy efficient data routing in wireless sensor networks. *AEU-International Journal of Electronics and Communications*, 131, pp.1-10.
- [10] Yadav S., V. Kumar, Hybrid compressive sensing enabled energy efficient transmission of multi-hop clustered UWSNs. *AEU-Int. J. Electron. Common.* 110, 152836 (2019).
- [11] Yu, W.; Chen, Y.; Wan, L.; Zhang, X.; Zhu, P.; Xu, X. An energy optimization clustering scheme for multi-hop underwater acoustic cooperative sensor networks. *IEEE Access* 2020, 8, 89171–89184.
- [12] Haider, S.K., Jiang, A., Almogren, A., Rehman, A.U., Ahmed, A., Khan, W.U. and Hamam, H., 2021. Energy efficient UAV flight path model for cluster head selection in next-generation wireless sensor networks. *Sensors*, 21(24), pp.1-22.
- [13] Bhattacharjya, K.; Alam, S.; De, D. CUWSN: Energy efficient routing protocol selection for cluster based underwater wireless sensor network. *Microsystems. Technol.* 2019, 1–17.
- [14] Zhang, J.; Cai, M.; Han, G.; Qian, Y.; Shu, L. Cellular clustering-based interference-aware data transmission protocol for underwater acoustic sensor networks. *IEEE Trans. Veh. Technol.* 2020, 69, 3217–3230.
- [15] Hou, R.; He, L.; Hu, S.; Luo, J. Energy-balanced unequal layering clustering in underwater acoustic sensor networks. *IEEE Access* 2018, 6, 39685–39691.
- [16] Anuradha, D., Subramani, N., Khalaf, O.I., Alotaibi, Y., Alghamdi, S. and Rajagopal, M., 2022. Chaotic search-and-rescue-optimization-based multi-hop data transmission protocol for underwater wireless sensor networks. *Sensors*, 22(8), pp.1-20.
- [17] Xiao, X., Huang, H. and Wang, W., 2020. Underwater wireless sensor networks: An energy-efficient clustering routing protocol based on data fusion and genetic algorithms. *Applied Sciences*, 11(1), p.312.
- [18] Fei, W.; Hexiang, B.; Deyu, L.; Jianjun, W. Energy-efficient clustering algorithm in underwater sensor networks based on fuzzy C means and Moth-flame optimization method. *IEEE Access* 2020, 8, 97474–97484
- [19] Krishnaswamy, V. and Manvi, S.S., 2019. Fuzzy and PSO based clustering scheme in underwater acoustic sensor networks using energy and distance parameters. *Wireless Personal Communications*, 108(3), pp.1529-1546.
- [20] Durrani, M.Y., Tariq, R., Aadil, F., Maqsood, M., Nam, Y. and Muhammad, K., 2019. Adaptive node clustering technique for Smart Ocean under water sensor network (SOSNET). *Sensors*, 19(5), p.1145.
- [21] Yahya, A., Islam, S.U., Zahid, M., Ahmed, G., Raza, M., Pervaiz, H. and Yang, F., 2019. Cooperative routing for energy efficient underwater wireless sensor networks. *IEEE Access*, 7, pp.141888-141899.
- [22] Omeke, K.G., Mollel, M.S., Ozturk, M., Ansari, S., Zhang, L., Abbasi, Q.H. and Imran, M.A., 2021. DEKCS: A dynamic clustering protocol to prolong underwater sensor networks. *IEEE Sensors Journal*, 21(7), pp.9457-9464.
- [23] Mohan, P., Subramani, N., Alotaibi, Y., Alghamdi, S., Khalaf, O.I. and Ulaganathan, S., 2022. Improved met heuristics-based clustering with multichip routing protocol for underwater wireless sensor networks. *Sensors*, 22(4), pp.1-16.

- [24] Tian, W., Zhao, Y., Hou, R., Dong, M., Ota, K., Zeng, D. and Zhang, J., 2023. A centralized control-based clustering scheme for energy efficiency in underwater acoustic sensor networks. *IEEE Transactions on Green Communications and Networking*, 7(2), pp.668-679.
- [25] Subramani, N., Mohan, P., Alotaibi, Y., Alghamdi, S. and Khalaf, O.I., 2022. An efficient metaheuristic-based clustering with routing protocol for underwater wireless sensor networks. *Sensors*, 22(2), pp.1-16.
- [26] 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*, vol. 24, no. 16, pp.1- 26616 – 26629.
- [27] Shyamsundar, R. and Harshavarthan, M., 2025. A cluster based routing for maximizing the lifetime of underwater wireless sensor network using gravitational search algorithm. *Results in Engineering*, 25, pp.1-14.
- [28] Arjunan, S., Pothula, S. and Ponnurangam, D., 2018. F5N-based unequal clustering protocol (F5NUCP) for wireless sensor networks. *International Journal of Communication Systems*, 31(17), pp.1-14.
- [29] Xiao, X. and Huang, H., 2020. A clustering routing algorithm based on improved ant colony optimization algorithms for underwater wireless sensor networks. *Algorithms*, 13(10), pp.1-17.
- [30] Benmamoun, Z., Khlie, K., Bektemyssova, G., Dehghani, M. and Gherabi, Y., 2024. Bobcat Optimization Algorithm: an effective bio-inspired metaheuristic algorithm for solving supply chain optimization problems. *Scientific Reports*, 14(1), pp.1-62.
- [31] Zhang, L., Zhu, Y., Ren, W., Wang, Y., Choo, K.K.R. and Xiong, N.N., 2021. An energy-efficient authentication scheme based on Chebyshev chaotic map for smart grid environments. *IEEE Internet of Things Journal*, 8(23), pp.17120-17130.