

Recovery from Network Faults in Underwater Wireless Sensor Network using Cat Swarm Optimization

Rinki Dev¹, Kuhoo²,

Madhuri Rao³ and Debahuti Mishra⁴

^{1 3 4}Dept. of Computer Science & Engineering,
Siksha 'O' Anusandhan(Deemed to be University),
Bhubaneswar, Odisha, India

rinkidev7002@gmail.com,

madhurirao@soa.ac.in, debahutimishra@soa.ac.in

²Department of Mechanical Engineering,

College of Engineering and Technology,

Bhubaneswar, Odisha, India

kuhoo.cet@gmail.com

Abstract

Underwater Wireless Sensor Network (UWSN) comprises of sensor nodes equipped with acoustic communication systems. UWSN are being extensively used in ocean habitat surveillance, marine explorations and in harbor surveillance. Deploying sensors inside the ocean is a very challenging job. UWSN nodes have unique characteristics such as small size, limited bandwidth and a very critical battery life. The battery of such nodes cannot be recharged by traditional techniques and hence routing of information needs to be minimized. It is not feasible to use wireless transmission techniques as salt water absorbs wireless signals, making acoustic communication the only choice of communication. The speed of transmission in acoustic

communication is only 1500 m/sec which is 2 Lakh times slower than available wireless sensor transmission techniques. Here in this work, we propose to carry our research work in recovery from network faults in underwater wireless sensor network using cat swarm optimization. Faulty transmissions occur in UWSN due to topology changes caused by mobility of nodes. Node mobility is a result of water current that cannot be eliminated. Hence, UWSN needs mechanisms to deal with faulty transmission. We aim to explore possible literature on UWSN and propose an innovative algorithm based on Cat Swarm Optimization (CSO) to recover from network faults.

Key Words : *Underwater Wireless Sensor Network, Network Faults, Cat Swarm Optimization (CSO)*

I INTRODUCTION

UWSNs consist of sensors deployed under the surface of water where communication is through acoustic medium. Since, the sensors are placed inside the water surface; the signals face long propagation delays, bandwidth failure and are prone to network issues due to node mobility. Underwater WSNs are being explored in harbor surveillance, aquatic habitat monitoring and also in ocean floor surveillance [1]. In future it could be used in oil pipelines or supervising hydroelectric reservoirs [2]. Also UWSNs can be used in the field of robotics where robots can make mutual decisions communicate effective information in a protected way and achieve network profit even inside water bodies [3-4]. They could be explored in detecting intrusion and pirate attacks in sailing ships [5-6]. UWSNs have various issues that need to be taken care of. Range of acoustic communication is narrowed by the energy suppression which causes adversity and provocation to localization of nodes in large scale [7-8]. So, node localization can be improved in large scale in underwater networks. Issues such as static node deployment, full sensing, communication and storing backups for future reference should be addressed [9-10]. Dynamic clustering needs to be

explored as nodes drift away from their initial clusters due to water current [11]. This network is prone to network faults due to its dynamic nature. Energy consumption of nodes is very challenging unless issues caused by node mobility are not well addressed. In underwater communication terrestrial routing protocols cannot be applied. So, self organized routing protocols are used to compute the propagation delay of acoustic waves in underwater communication [12]. *Depth and Energy Aware Cooperative* routing protocol is used in UWSN for effective and reliable data transmission [13-14]. Limited bandwidth, limited battery, less memory, network fading, etc issues in UWSN can be solved by location based routing protocol [15-16].

The layout of the paper is as follows; section 2 discusses the application of CSO for UWSN with the seeking and tracing phases of it. The experimental evaluation and result analysis has been given in Section 3 and finally, section 3 given the conclusion of the proposed work.

II CAT SWARM OPTIMIZATION FOR NETWORK RECOVERY IN UWSN

CSO is defined as one of the latest swarm-based algorithm. CSO comprises of two modes, seeking mode and tracing mode which simulates the resting and hunting behaviors of cats. CSO has been explored by Amara *et al.* [17] in optimizing the reliability of design and cost constraints in Hybrid Wind Gas using Cat Swarm intelligence. Bahrami *et al.* [18] reviewed the superiority of CSO over different algorithms along with their different variations developed by other researchers. Modified Cat Swarm Optimization (MCSO) was presented by Lin *et al.* [19] where they used CSO along with local search procedures which resulted in improved search efficiency in problem space. Mohamadeen *et al.* [20] presented a new version of CSO i.e. Binary Cat Swarm Optimization (BCSO) to design an optimization model based on Support Vector Machines (SVM). This model was used to find the best transformer tests based on various climatic, biological and chemical factors. CSO can be considered as an accurate clustering method. It is based on the individual behavior of cats. Energy of the system is conserved as each node

(Cat) performs seeking and tracing mode of operation alternatively. Tsai *et al.* (2016) [21] proposed a sink node placement method by utilizing a newly designed object function and employing the CSO algorithm to find out the near best location for placing the sink node.

UWSN are network characterized with dynamic topology. The dynamic topology is witnessed due to water current. Water current is inevitable. Water current causes nodes to drift away from their initial position of deployment. Routing techniques employed in UWSN, therefore need to address the issues that arise due to water current. The network should be adaptable and deal with faults that arise due to node mobility. Different types of faults and error arise in the network. Nodes are not aware of the new location of neighboring nodes and continue transmitting packets as per old position and schedule. This leads to transmissions errors, link failures, and packet loss. Network is likely to be congested with many packets being forwarding without success. Network lifetime is affected and reduced drastically. Network throughput is also adversely affected. Packet delay increases making the network becomes more unreliable and the network is prone to more disconnections and partitions. This affects traditional clustering approaches in UWSN as well. To address these issues, a fault detection and recovery mechanism is needed for UWSN. The technique that could help the network in adapting its routing information based on node mobility is needed. Such a mechanism should first detect faults in the network. Faults are areas of the network topology with possible network disconnections. If the network is disallowed from being partitioned it will help the network in avoiding many of the issues listed above. Here, we propose a solution to deal with network faults. Detection of network faults needs further exploration. Here, the network architecture as proposed by Rao *et al.* [22] is considered for the proposed algorithm. It includes an Aerial Mobile Sink (AMS) node that communicates with the base station. It is a mobile robotic device that traverses an aerial path visiting all cluster head nodes. Cluster head nodes are second type of nodes. They are devices equipped with both wireless and acoustic mode of communication. They communicate with the AMS using its wireless interface. Communication with the underwater wireless sensor nodes is achieved via acoustic means. Finally, the third category of

nodes is the UWSN Nodes. They report to a given cluster head at a given point of time. Further in this work, the UWSN nodes are assumed to be equipped with capabilities of being mobile. These nodes are considered to be CAT nodes. They are assumed to be in two modes. In seeking mode, each cat nodes observes its environment. In tracing mode, the node which has maximum number of neighboring nodes and requires minimal energy consumption is selected to move to regions in the network that are likely to be less covered. The proposed algorithm is inspired from CSO. Here, underwater sensor nodes are considered to depict the behavior of cats. A cat is said to be in two modes such as; in *seeking mode*, the cat observes every activity in its neighborhood and in the *tracing mode* the cat makes a calculated move towards a prey. Here, we consider all underwater nodes to be cat nodes. An empty or void space in the network is assumed to be a prey. Each cat node observes its neighborhood in the seeking mode in search of its prey. In the tracing mode the cat with the best fitness function moves to hunt the prey. The proposed algorithm thus detects network faults and by moving its nodes wisely avoids network disconnections. By preventing network disconnections, the system is fault tolerant. It helps in achieving better throughput and in enhancing network lifetime. The proposed CSO algorithm has two phases seeking phase and tracing phase.

A. Seeking phase of the proposed algorithm

Here, N number of cats is considered for simulation. Energy of each node is initially considered to be 5 Joules. For each data packet reception 0.1 Joules of energy is spent, while for each transmission 0.2 Joules of energy is utilized. In moving from current position to new position 0.4 Joules of energy is considered to be expended. In observing the environment 0.01 Joules of energy is considered to be spent every second. The current position of the cats is randomly generated using (1) and (2).

$$Cat_x = 1 + (10 - 1) \times rand \quad (1)$$

$$Cat_y = 1 + (10 - 1) \times rand \quad (2)$$

The next possible position of the cats are calculated using the (3).

$$X_{cn} = (1 \pm SRD \times R) \times X_C \quad (3)$$

Here, X_C is the *Current position* of a given cat, X_{cn} is the new position and R is a random number of range ($0 < R < 1$). The probability that cat_i is selected for next transmission depends on the maximization and minimization function as expressed in (4). At this point of time, the void region or the region with less network coverage is assumed to be known.

$$P_i = |FS_i - FS_b| / |FS_{max} - FS_{min}| \quad (4)$$

Where, P_i is Probability of selection of current candidate cat_i for next transmission, FS_i is the current fitness value of cat_i , FS_{max} is the maximum value of fitness function and FS_{min} is minimum value of fitness function for a possible action, while FS_b is the estimated fitness function.

Here, FS_{max} is a *minimization function* that aims to ensure that ensures nodes which requires least energy for next transmission is selected. FS_{min} is a *maximization function* that selects nodes which have maximum number of neighbors as the next node for transmission. By doing so, the algorithm ensures that a node with greater network coverage and with higher residual energy is selected for next transmission. This *objective function* which involves both minimization and maximization fitness functions therefore eliminates network faults as expressed in (5).

$$FS_b = 0.5 \times E_t + 0.5 \times n \quad (5)$$

Here, E_t is the minimum energy required for next transmission to cluster head and n here represents the number of current neighbors for a given node. The node closest to cluster head which is only one hop away will require least energy for data transmission. Also, for assuring reliable network coverage a node is expected to have at least 1 neighbor one hop away, while a maximum of 4 nodes is considered for maximum coverage. Hence, the FS_{max} is calculated as given in (6) and FS_{min} is calculated as shown in (7). Taking

these values, the cat with the best fitness value is selected to move in *tracing mode*.

$$FS_{max} = 0.50.21 + 0.54 = 2.1 \quad (6)$$

$$FS_{max} = 0.50.23 - 0.51 = 0.2 \quad (7)$$

B. Tracing phase of the proposed algorithm

The cat with the highest probable value of fitness function is selected as the cat with the best solution. The greatest value of minimization and maximization problem is compared and stored in the array *best*, X_{best} gives the greatest value of the best array. At this point of time, the void region or the region with less network coverage is assumed to be known using (8).

$$\begin{aligned} best &= \max(Prob_{max}, Prob_{min}) \\ X_{best} &= \max(best) \end{aligned} \quad (8)$$

Then, the velocity of the cats with respect to the best probable value is calculated as given in (9).

$$v_{k,d} = v_{k,d} + r_1 \times c_1 (X_{best,d} - X) \quad (9)$$

Where, $v_{k,d}$ = velocity of cat_k in dimension d , $X_{best,d}$ = position of the cat with the best solution, $X_{k,d}$ = position of the cat_k , c_1 = a constant and r_1 a random value in the range of $[0,1]$. Then accordingly the new position of the others cats is calculated in order to assure that network faults is reduced in an energy efficient manner. The proposed algorithm is presented below.

The proposed algorithm is assumed to be executed every T_p time. Here, T_p is the time duration of one iteration of the proposed algorithm which consists of two phases or modes. The seeking mode and tracing mode of the algorithm are performed linearly, one after the other. The location of the network fault is assumed to be known by the *Cluster Head* node. **Fig. 1** depicts the control flow of the proposed algorithm.

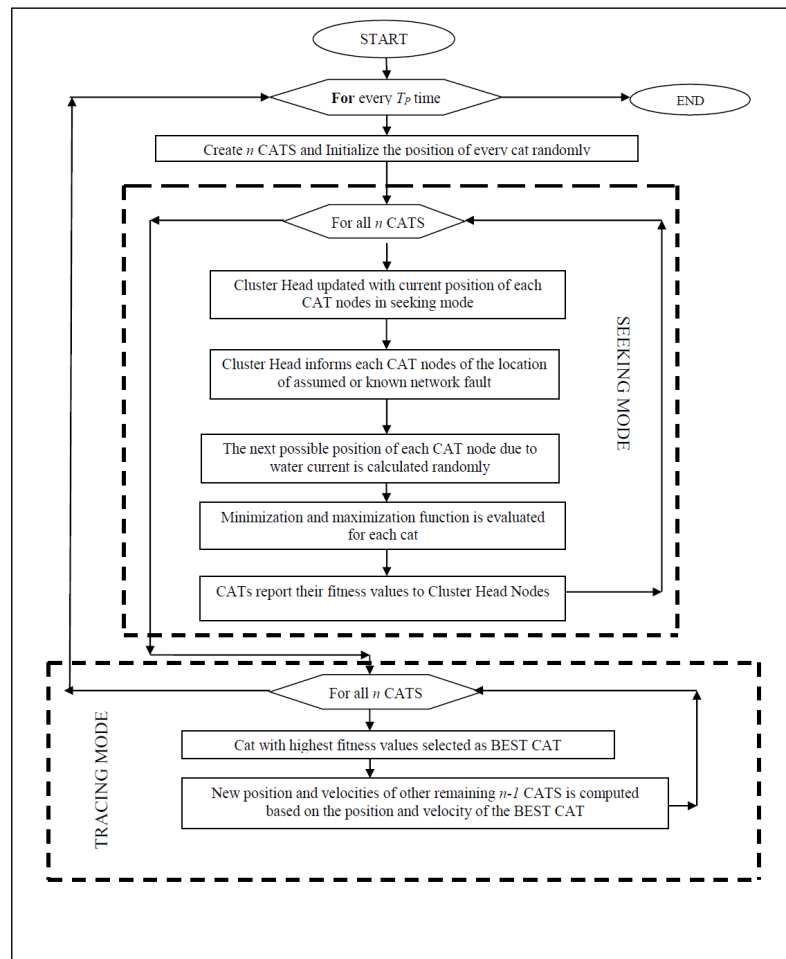


Fig. 1. Control flow of the proposed algorithm

Algorithm 1: Proposed algorithmStep 1: **for** every T_p time

// Begin of Seeking mode

Step 2: **for** every n // n is number of CAT nodes

Step 3: Cluster Head updated with current position of each CAT nodes in seeking mode

Step 4: Cluster Head informs each CAT nodes of the location of assumed or known network fault

Step 5: The next possible position of each CAT node due to water current is calculated randomly

Step 6: Minimization and maximization function is evaluated for each cat

Step 7: CATs report their fitness values to Cluster Head Nodes

```

Step 8: end for
// End of seeking mode.
// Begin of Tracing mode
Step 9: for every  $n$  //  $n$  is number of CAT nodes
Step 10: Cat with highest fitness values selected as BEST CAT
Step 11: New position and velocities of other remaining  $n-1$  CATs is computed based on the
position and velocity of the BEST CAT
Step 12: end for
// End of Tracing mode
Step 13: end for
// algorithm repeated after every  $T_p$  time

```

III EXPERIMENTAL EVALUATION AND RESULT ANALYSIS

Initially the simulation is carried out for 10 number of cat nodes. The cats are in seeking mode in the first part of the algorithm. Here cats are underwater sensor nodes considered with *5 Joules* of initial energy. The current position of 10 cats in X axis is obtained using (1). The values of $cat_x = \{1.7599 \ 3.3388 \ 4.8827 \ 2.6366 \ 2.3099 \ 8.8236 \ 5.9487 \ 8.6773 \ 4.1586 \ 4.6163\}$. Further the current position of 10 cats in Y axis using (2) gives us following values of $cat_y = \{4.5980 \ 8.2006 \ 9.1958 \ 3.3742 \ 2.2246 \ 6.2173 \ 2.3046 \ 6.5985 \ 5.6192 \ 1.6837\}$. The current position of cat nodes taken initially is plotted in **Fig. 2** below.

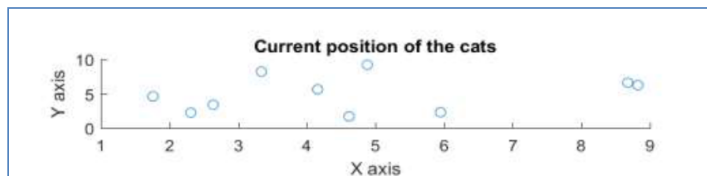


Fig. 2. Current position of cat nodes

The next possible position of underwater sensor nodes, assumed to be cat nodes here is calculated using (3). The values obtained for $X_{pos} = \{0.5489 \ 0.7982 \ 2.6486 \ 3.0942 \ 1.4740 \ 3.8739 \ 2.8555 \ 8.8016 \ 1.3066 \ 0.5788\}$, while the values of $Y_{pos} = \{0.7371 \ 2.5581 \ 0.5936 \ 4.1443 \ 1.4149 \ 7.2747 \ 0.3332 \ 3.3432 \ 2.9506 \ 0.2889\}$. Next, the fitness values of each cat using (3) and (4) is evaluated to find the probability of for selection. Here, $FS_{min} = -0.200$ and $FS_{max} = 2.1000$. The possible position of cat nodes

is depicted in **Fig. 3**.

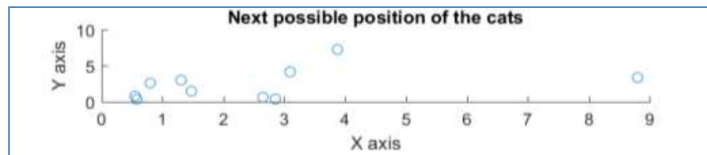


Fig. 3. Next possible position of cats

For each data packet reception $0.1J$ of energy is considered to be spent. For each transmission $0.2J$ of energy is utilized. Moving to a different point requires $0.4J$ of energy and observing the environment requires $0.01J$ of energy per second, for this simulation. FS_{max} is a minimization function that aims to ensure that nodes which requires least energy for next transmission is selected. Using (5) the fitness functions are evaluated. On running the simulation for 10 nodes the values obtained for minimization function are stored in array $Prob_{max}[]$ and its values obtained as : $Prob_{max} = \{0.4965 \ 0.5027 \ 0.3370 \ 0.1129 \ 0.1890 \ 0.2405 \ 0.4440 \ 0.0937 \ 0.1057 \ 0.1604\}$. FS_{min} is a maximization function that selects nodes which have maximum number of neighbors as the next node for transmission. The values obtained for 10 nodes are stored in array $Prob_{min}[]$. $Prob_{min} = \{0.6308 \ 0.5949 \ 0.6314 \ 0.7170 \ 0.6752 \ 0.7842 \ 0.5893 \ 0.8309 \ 0.6144 \ 0.8333\}$. Further all nodes are now progressed to tracing mode, where the Best node is selected based on (8). Hence, for this simulation $X_{best} = 0.8333$ is obtained. Using (9), further the velocities of all nodes is calculated in order to ensure that nodes move in such a manner that network faults are reduced. The velocities of the 10 nodes obtained are $vel = \{0.0310 \ 0.0365 \ 0.0309 \ 0.0178 \ 0.0242 \ 0.0075 \ 0.0374 \ 0.0004 \ 0.0335 \ 0\}$. Finally, the new position of the nodes is obtained in $X_{new} = \{1.7909 \ 3.3753 \ 4.9136 \ 2.6544 \ 2.3341 \ 8.8311 \ 5.9861 \ 8.6776 \ 4.1921 \ 4.6163\}$ and $Y_{new} = \{4.6291 \ 8.2371 \ 9.2267 \ 3.3920 \ 2.2488 \ 6.2249 \ 2.3420 \ 6.5989 \ 5.6528 \ 1.6837\}$. **Fig. 4** Depicts the new position of the cat nodes achieved with the help of the proposed algorithm.

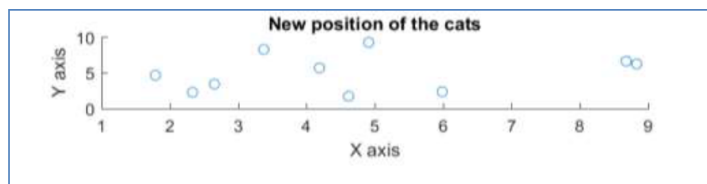


Fig. 4. New positions of cat nodes with proposed algorithm

It can be noted from **Fig. 4** that the proposed algorithm achieves a better node distribution than otherwise. This essentially helps in reducing network disconnections and network faults. This can be proved by calculating the mean distance of underwater sensor nodes from the cluster head as depicted in **Fig. 5**.

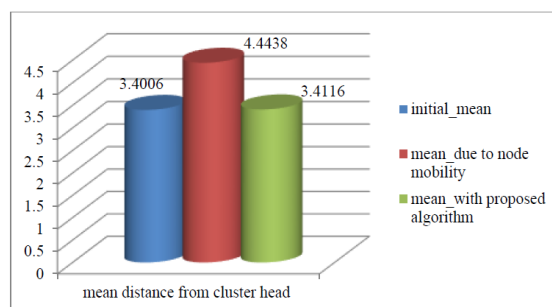


Fig. 5. Ratio of change in mean distance of UWSN nodes from CH

Fig. 5 depicts three possible mean distances from the cluster head node. *initial_mean* is the mean distance of the underwater nodes at the initial state of the algorithm. *mean_due to node mobility*, indicates the possible mean distance of nodes from the cluster head after the nodes have moved away due to water current. While, *mean_with proposed algorithm* is the mean distance of nodes from cluster head after executing the proposed algorithm for T_p time. It can be seen from **Fig. 5** that the proposed algorithm helps in reducing the mean distance of the nodes from the cluster head, which increases due to node mobility. If nodes move away from the cluster head, they are likely to affect the network performance due to enhanced delay. It also affects the routing table. Further, an adjacency matrix is constructed for the given topology that helps

in calculating the covariance of the considered topology. The adjacency matrix for the initial topology was found to be as depicted in $A[]$, while $B[]$ is the adjacency matrix that resulted due to mobility and water current. $C[]$ is the adjacency matrix achieved after running the proposed algorithm for T_p time. It is found that the proposed algorithm is effective in reducing the covariance of the topology.

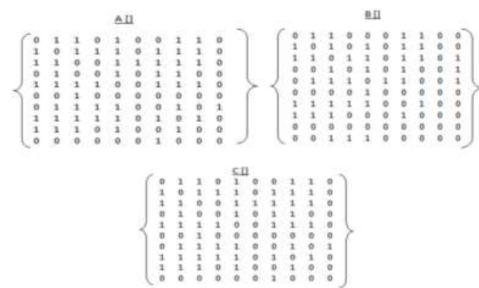


Fig. 6. Adjacency matrices with respect to CH

It is seen from the adjacency matrix $A[]$ that there were 51 edges between the nodes initially for the considered topology. However, due to node mobility, the number of edges between nodes is reduced to 40, as shown in $B[]$. This proves that node mobility affects network connectivity and creates network faults. This issue is addressed by our proposed algorithm, which effectively restores the number of edges to 49 edges, as shown in $C[]$.

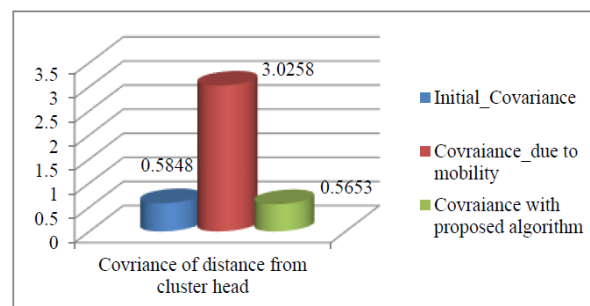


Fig. 7. Ratio of change in covariance of distance of UWSN nodes from CH

Further the covariance of the mean distance between the nodes and the cluster head is calculated to show the effectiveness of the proposed algorithm. **Fig. 7** depicts the initial covariance for the considered topology. It also shows the covariance that results due to node mobility and finally depicts the covariance achieved with the proposed CSO algorithm. The proposed algorithm is effective in restoring the network connectivity by 9%. It reduces the adverse effect of changes in network topology due to node mobility by 68.52%.

IV CONCLUSION

UWSN unlike traditional WSN is more prone to issues that arise due to node mobility. Often UWSN networks are dynamic in nature. Traditional WSN protocols are not suitable for UWSN in their current form. UWSN are prone to network failures, link loss and congestion due to node mobility. Node mobility is a result of water current which cannot be eliminated. This network therefore experiences unreliable data transmission owing to node mobility. Data transmission is impaired due to network faults and link loss. A mechanism to recover from network faults in UWSN is proposed in this research. The proposed algorithm is inspired by the nature of CSO. Here, each underwater sensor node is assumed to be in two modes; seeking mode and tracing mode. In seeking mode, each node receives information about an assumed network fault while it is observing its environment. In the tracing mode, each node predicts a possible path to deal with the network fault. The cat with the best solution moves towards the network fault and the other cats in the network align themselves accordingly. The proposed algorithm thus avoids network faults which are usually void spaces created in the network due to node mobility. The proposed algorithm was found to be effective in restoring network connectivity. It also reduced the adverse effect of changes in network topology caused due to node mobility by 68.52%.

References

- [1] Heidemann, J., Stojanovic, M. and Zorzi, M., 2012. Underwater sensor networks: applications, advances and challenges. *Phil. Trans. R. Soc. A*, 370(1958), pp.158-175.
- [2] Vieira, L.F.M., Vieira, M.A.M., Nacif, J.A.M. and Vieira, A.B., 2018. Autonomous Wireless Lake Monitoring. *Computing in Science & Engineering*, (1), pp.66-75.
- [3] Ferri, G., Munaf, A., Tesei, A., Braca, P., Meyer, F., Pelekanakis, K., Petroccia, R., Alves, J., Strode, C. and LePage, K., 2017. Cooperative robotic networks for underwater surveillance: an overview. *IET Radar, Sonar & Navigation*, 11(12), pp.1740-1761.
- [4] Rao, M., Kamila, N.K. and Kumar, K.V., 2016, October. Underwater wireless sensor network for tracking ships approaching harbor. In *Signal Processing, Communication, Power and Embedded System (SCOPEs), 2016 International Conference on* (pp. 1872-1876). IEEE.
- [5] Rao, M., Kamila, N.K. and Kumar, K.V., 2016, October. Underwater wireless sensor network for tracking ships approaching harbor. In *Signal Processing, Communication, Power and Embedded System (SCOPEs), 2016 International Conference on* (pp. 1872-1876). IEEE.
- [6] Rao, M. and Kamila, N.K., 2017. Tracking intruder ship in wireless environment. *Human-centric Computing and Information Sciences*, 7(1), p.14.
- [7] Zhang, B., Wang, H., Zheng, L., Wu, J. and Zhuang, Z., 2017. Joint Synchronization and Localization for Underwater Sensor Networks Considering Stratification Effect. *IEEE ACCESS*, 5, pp.26932-26943.