International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5948
***Article Received**: 07 October 2022 **Revised**: 16 November 2022 **Accepted**: 21 December 2022*

Minimizing the Localization Error in Wireless Sensor Networks Using Multi-Objective Optimization Techniques

[1]M. Sri Lakshmi, [2]Dr. K. Seshadri Ramana,[3] M. Jahir Pasha,[4]K. Lakshmi, [5]N. Parashuram, [6]M. Bhavsingh
[1,4,5] Assistant professor, Department of computer science and Engineering,
G.Pullaiah College of Engineering and technology, Kurnool
[2]Professor, Department of Computer Science &Engineering
Ravindra College of Engineering for Women Kurnool
[3]Associate professor, Department of computer science and Engineering,
G.Pullaiah College of Engineering and technology, Kurnool
[6] Assistant Professor, Ashoka Women's Engineering College, Kurnool
Email ID: srilakshmicse@gpcet.ac.in ,ramana.kothapalli @gmail.com , jahir4444@gmail.comlakshmicse@gpcet.ac.in ,parashuramcse@gpcet.ac.in,bhavsinghit@gmail.com

Abstract: When it comes to remote sensing applications, wireless sensor networks (WSN) are crucial. Because of their small size, low cost, and ability to communicate with one another, sensors are finding more and more applications in a wide range of wireless technologies. The sensor network is the result of the fusion of microelectronic and electromechanical technologies. Through the localization procedure, the precise location of every network node can be determined. When trying to pinpoint the precise location of a node, a mobility anchor can be used in a helpful method known as mobility-assisted localization. In addition to improving route optimization for location-aware mobile nodes, the mobile anchor can do the same for stationary ones. This system proposes a multi-objective approach to minimizing the distance between the source and target nodes by employing the Dijkstra algorithm while avoiding obstacles. Both the Improved Grasshopper Optimization Algorithm (IGOA) and the Butterfly Optimization Algorithm (BOA) have been incorporated into multi-objective models for obstacle avoidance and route planning. Accuracy in localization is enhanced by the proposed system. Further, it decreases both localization errors and computation time when compared to the existing systems.

Keywords: wireless sensor networks, localization error, localization models; Grasshopper Optimization Algorithm (GOA) , Butterfly Optimization Algorithm (BOA), Dijkstra; path planning.

I. Introduction

There have been many proposals for localization strategies in recent years. Nearly all of them [1, 2] are designed for permanently installed sensor networks. On the other hand, some people believe that the sensor can detect the location of a mobile device. The nodes that are aware of their position, called anchor nodes, are distinguished from the other nodes, which are called regular nodes. Positional awareness among sensor nodes is crucial in many situations [3]. For instance, the sensed data combined with location information allows a server to instantly determine where an event has occurred. The sensor nodes themselves could be equipped with Global Positioning System (GPS) receivers to facilitate an easy and precise method of pinpointing sensor locations. But the cost of production makes it impractical for use indoors. In addition, most uses necessitate only a rough localization [4]. Some of the sensor nodes could have a global positioning system (GPS), and the rest could rely on an automatic localization scheme, but the former is the more practical solution. In addition, sensor nodes can move around, greatly increasing the area that can be monitored. Consequently, mobile sensor networks require a planned localization scheme [5].

Most nodes in WSNs are deployed at uncontrollable, unpredictable locations. To solve the localization problem in WSNs, one must determine where each sensor node is located. The localization issue can be solved by employing the global positioning system (GPS) to pinpoint the precise location of sensor nodes. However, this method is infeasible in WSNs due to the fact that these networks are made up of relatively large sensor nodes, and installing a GPS receiver on each of these nodes would significantly increase the network's overall cost, complexity, and energy consumption. Researchers have developed a number of localization strategies to address these issues. Only a subset of sensor nodes, called anchor nodes, needs to be equipped with a global positioning system. The localization methods can be broken down into two distinct groups: those that require a fixed range and those that don't. In range-based localization, in addition to the angle of arrival (AOA), received signal strength (RSS), and other similar metrics, connectivity data are also used.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5948
***Article Received**: 07 October 2022 **Revised**: 16 November 2022 **Accepted**: 21 December 2022*

The proposed system uses the Improved Grasshopper Optimization Algorithm (IGOA) and the Butterfly Optimization Algorithm (BOA) for multi-objective optimization. To determine the shortest route, we use Dijkstra's algorithm and path planning software like Grasshopper's Integrated Grasshopper and Butterfly's BOPP. This work makes the following contributions to the field of study:

1. We use the IGOA and the BOA to create a localization system for wireless sensor network communication, inspired by the efficacy of multi-objective optimization methods.
2. To compare how well the improved localization model proposed works with the current model.

Precisely, this paper is organized into the following sections: In the second part, we provide a deep summary of the relevant prior studies. Section 3 will provide an overview of WSNs that use multi-objective optimization as well as the two different optimization approaches that are commonly used. Section 4 presents the findings and analysis, and Section 5 conclude the paper.

II. Related Work

Several strategies for boosting node localization precision have been discussed here. There have been many attempts to find ways to enhance localization precision by employing AI methods. A Bayesian algorithm for WSN node localizations was first presented by Morelande et al. [6]. The proposed algorithm is an improved version of "progressive correction" [7], which was published previously. The two approaches are contrasted using the Cramér-Rao bound (CRB) as a standard across a range of circumstances. There was a noticeable improvement in accuracy with the new algorithm. In addition, Ghargan et al. [8] proposed a method in which one ANN is hybridized independently with the three optimization algorithms of particle swarm optimization (PSO), backtracking search algorithm (BSA), and gravitational search algorithm (GSO) (GSA). The GSA-ANN hybrid achieved the best results, with a mean absolute distance estimation error of 0.02 m for outdoor scenarios and 0.2 m for indoor scenarios. A recent study compiled various cutting-edge machine learning techniques for WSN node localization [29].Methods like artificial neural networks (ANN), support vector machines (SVM), decision trees (DT), and the naive Bayes (NB) approach had their error distribution curves for localization compared. According to the cumulative localization error distributions reported in this study, NB performed better than all of the other machine learning methods. Using a number of supervised, unsupervised, and ensemble machine learning techniques, the authors of [10] created an outlier detection algorithm for an indoor localization setting called iF Ensemble. K-nearest-neighbor, random forest, and support vector machines are the supervised methods, while isolation forest is the unsupervised method (iForest). As part of stacking, an ensemble learning strategy, these methods are implemented. The model's results are compared to those of the individual machine learning algorithms used to create it, including the results of any stacking that may have been done. When using the suggested techniques for detecting outliers, the stacking model achieves a remarkable degree of localization accuracy—97.8 percent. Recently, [11] presented a Hop-Count quantization-based node localization algorithm called Kernel Extreme Learning Machines (KELM-HQ). Unknown node locations are calculated by the trained KELM. Results show that the proposed algorithm reduces localization error by 34.6% compared to fast-SVM, 19.2% compared to GADV-Hop, and 11.9% compared to DV-Hop-ELM. If nodes don't know where they are, localization can't tell them where they should be looking, and that could be a waste of time. GPS is used to easily pinpoint the location of the node, but this becomes more expensive when a large number of sensor nodes are deployed in a single area. Many algorithms have been proposed [12] to address localization difficulties. Overall, this research hopes to improve upon the localization accuracy of prior studies by employing a machine learning technique based on regression.

If nodes don't know where they are, localization can't tell them where they should be looking, and that could be a waste of time. Even though the Global Positioning System (GPS) can be used to easily localise the node, doing so for a large number of sensor nodes deployed in a single area can become quite expensive [13]. In order to address the localization problems, numerous algorithms have been proposed [14].

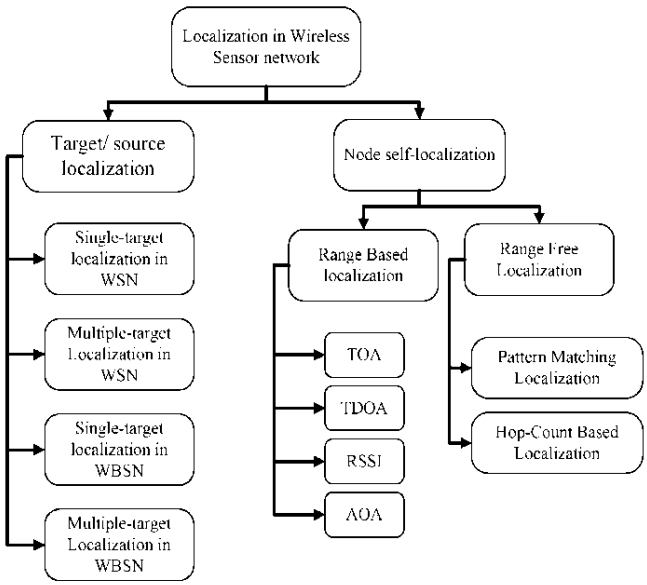

Figure 1. Localization in WSN

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5948
Article Received: 07 October 2022 Revised: 16 November 2022 Accepted: 21 December 2022

III. Methodology

3.1 Multi-objective localization using swarm intelligence

The localization algorithm, which is based on a distance routing protocol, is, in essence, a range-free localization algorithm. In WSNs, hop count and average hop distance are calculated by estimating the distance between beacon nodes and dumb nodes. Given the random nature of the connections between wireless sensor nodes, the network topology asserts various paths, some of which are not straight, between beacon nodes and dumb nodes. In this way, the algorithm's execution time during the node localization process can be used to visually represent certain errors. Wireless sensor networks (WSN) rely heavily on location data, and many of these networks' applications would be severely hindered without it. One example of a path-dependency algorithm, distance vector-hop (DV-Hop) [15], relies on connectivity and multi-hop transmission to derive an approximate position. Its simple and rough estimation method, however, leads to a significant positioning error.

3.2 Improved grasshopper optimization algorithm (IGOA)

The theoretical underpinnings of GOA are straightforward, and the approach is straightforward to implement. However, there are drawbacks that prevent the algorithm from generating optimal results. Unfortunately, the original GOA was unable to make the most of each iteration due to its linearly diminishing comfort zone. Due to the lack of random factors, the original algorithm has low variability. The algorithm quickly settles on a suboptimal solution. A nonlinear comfort zone parameter, a local search mechanism based on Levy's flight, and a random jumping strategy were introduced to address the drawbacks. The specifics of the aforementioned three enhancements are laid out here.

Nonlinear comfort zone parameter:

In the original GOA, changing the radius of the safety zone causes the search agents to iteratively approach the global optimum solution. When conducting a GOA [16], the search space is narrowed by means of the convenience zone parameter. The search agents need time to find an approximate optimum, so it's best if the comfort zone parameter is large enough so they have some breathing room during the exploration phase. To prevent the search agents from speeding past the local optimum during exploitation, the restrictive factor should be kept low. However, the search ability could not be made to coincide with the exploration and exploitation phases during the search iterations due to the linearly decreasing factor. The sigmoid function was implemented to improve the algorithm's search performance by harmonizing its two separate phases of investigation. A nonlinear threshold function that sees frequent use is the sigmoid function. The field of information science makes extensive use of it. The sigmoid function formula is as follows:

$$f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

Using a modification of the sigmoid function, the following approach is made for a nonlinear safety-zone parameter:

$$m = \frac{-0.5}{1+e^{-x}} + u \quad (2)$$

where u is the adjustment parameter and its value should be in the interval [0, 1].

Local search mechanism based on Levy flight

A sigmoid-based nonlinear comfort zone parameter is proposed below. When it comes to GOA, you can always count on a set of predetermined parameters. Every search agent could only look in one place, and a lack of randomness could dampen their inventiveness during the search iterations. It is common practise to add some randomness to a deterministic system in order to boost its efficiency. In order to improve the search agents' efficiency in locating optimal solutions, a local search mechanism based on L'evy flight is proposed. Upon completion of the location updating process, each search agent's position should be updated via Levy flight in accordance with a predetermined probability. The formula for the correction is as described below:

$$X_i = X_i + 10c \times s_{threshold} \times Levy(dim) \times X_i \quad (3)$$

where $s_{threshold}$ is the threshold parameter which controls the direction and the probability of the variation. $s_{threshold}$ is calculated as follows:

$$s_{threshold} = sign(x_{trans} - 1) + sign(x_{trans} + 1) \quad (4)$$

Random jumping strategy.

The theory behind GOA is simple. The algorithm gives no consideration to how to escape the local optimum and instead only considers how to converge to the global one. This meant that the GOA search process got stuck in a cycle of reaching a local optimum and never moving forward. It's possible that GOA's simplicity of implementation is actually a drawback, as it may prevent it from helping to eliminate the local optimum. In order to facilitate the ability to escape the local optimum, a random jumping strategy is introduced. When a search agent locates a better one, it can take the place of the previous target location. As soon as it stops, the equation for random jumping becomes effective. As explained below:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5948
***Article Received**: 07 October 2022 **Revised**: 16 November 2022 **Accepted**: 21 December 2022*

$$X_i^{new} = \left(\left(0.5 - rand\right) \times 2 + 1\right) X_i \quad (5)$$

where X_i is the position of the i-th search agent, and X_i^{new} is the new position after random jumping. If X_i^{new} has better fitness, it will replace X_i . Thus, action of jumping out occurs successfully.

Procedure of IGOA.

An Enhanced Grasshopper Optimization Algorithm (IGOA) is outlined in this paper. The IGOA process consists of the following steps: (1) initialization; (2) evolution (3) fitness updating; (4) jumping. presented as Algorithm 1, the IGOA pseudocode. Figure 2 presents the overarching structure of the IGOA process.

Flowchart: Proposed IGOA

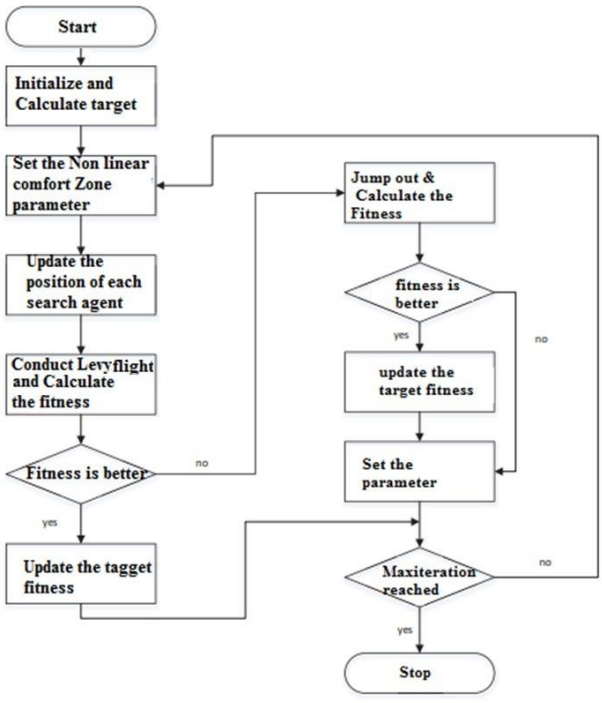

Figure 2. Proposed Flow Chart

Algorithm IGOA

1: *initialize the parameters*
2: *initialize the swarm position with random matrix*
3: *calculate the original target fitness and mark the target position*
4: *while (iter < MaxIteration* and Target fitness>*destination fitness) do*
5: *set the nonlinear comfort zone parameter m by Equation* (2)
6: *update* x_i *by Equation* (3)
7: x_i *conducts L'evy flight by Equation* (3)
8: *calculate the fitness*
9: *if current fitness is better than the target fitness then*
10: *update the target fitness and the target position*
11: *else*
12: x_i *jumps out by Equation* (3)
13: *calculate the fitness*
14: *if current fitness is better than the personal fitness then*
15: *update the personal position*
16: *end if*
17: *set parameters p*
18: *end if*
19: *end while*
20: *Return target fitness and target position*

3.3 Butterfly optimization algorithm

The intellectual Behaviour of butterflies while searching for a food source inspires the Butterfly Optimization Algorithm (BOA). Butterflies can navigate to their food source by analyzing scents [2]. Local searches and global searches are the two main types of searches. The butterfly employs a "global search" strategy when it senses the aroma of food and a "local search" strategy when it has exhausted that strategy and is forced to randomly choose its position in an attempt to locate its prey.

BOA has three phases: the initial, the iterative, and the final. First, the initialization phase is carried out, followed by iterative searching, and finally, the termination phase, at which point the algorithm is terminated once the best solution has been determined. This sequence is processed by BOA [17] during each run. It has been decided upon which constraints will be used in BOA. As soon as the values are assigned, an initial population of butterflies is generated for optimization. There is a set amount of memory available to hold this data, and both the memory size and the butterfly shape are unchanging. The positions are generated on a completely arbitrary grid. In the following section, we will enter the iteration stage. Each iteration involves re-evaluating the fitness value and the positions and then relocating all of the solutions in the butterfly space to the new positions. Formulas are used by butterflies to create scents at specific locations (9). In BOA, there are two distinct types of searches: a global search and a local search. In the former, the butterflies flutter toward the global search space's best solution vector, which is shown as

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5948
***Article Received**: 07 October 2022 **Revised**: 16 November 2022 **Accepted**: 21 December 2022*

$$y_i^{t+1} = y_i^t + (r^2 \times h^* - y_i^t) \times fi$$

(6)

Where y_i^t Solution vector of $i-th$ butterfly in iteration. fi is referred to the fragrance of i-th butterfly and random number r is taken as [0, 1]. The local search algorithm is shown as

$$y_i^{t+1} = y_i^t + (r^2 \times yj^t - yk^t) \times fi$$

(7)

Where yj^t and yk^t are butterflies from j and k in the solution space. If yj^t and yk^t are from the same swarm and random number r is [0, 1] then equation (7) converts to a local random search.

3.4 Implementation of multi-objective localization

It is a two-dimensional network with a square shape. The nodes, denoted by N, are all assumed to be mobile and movable. Since initial nodes have no information about their surroundings, they are all treated as if they were unknown nodes. This root uses nodes with local knowledge to inform their neighbours about their surroundings. There are several obstacles in the network area. The network's provided barrier is denoted by the letter "O" and has a circular outline. A mobile anchor is free to roam the network space anywhere except in close proximity to the obstruction. The mobile anchor is used to reduce the number of communication nodes. In this context, "M" represents the total number of mobile anchors (MA) [18] in the immediate vicinity of the network. Through its obstacle detection method, the MA can identify and avoid potential dangers. If MA exceeds the value, it stops providing the current position to the nodes. If the MA and UNs are not within range of one another, they will not be able to exchange data with one another. The UN contacts a reference node after learning where its members currently are. If QNs broadcast their current position to the other nodes, we can get a rough idea of where the UN is.

The localization error rate is minimized by the fitness function, it is denoted as

$$error_{total} = \sum_{k=1}^{N} error_{(k)}$$

(8)

The localization error of node i is error (*k*) and it is given as

$$error_k = \sqrt{(x_k - u_k)^2 + (y_k - v_k)^2}$$

(9)

Where (x_k, y_k) denote the location of node k and $(u_k, v_k$)denote the value of the same node k. Fitness values are calculated for each candidate node using IGOA and BOA techniques, and the final node is determined by maximum a posteriori (MA) estimation within the region. For the former, IGOA suggests the Grasshopper Optimization Path Planning (GOPP), while BOA suggests the Butterfly Optimization Path Planning (BOPP).

IV. Result analysis

Models are tested and analysed to determine how effective they will be. As a result, the constraints that were generated have been tried and true. Two algorithms, IGOPP and BOPP, are compared for their localization accuracy, localization ratio, and computational time. Path planning using IGOPP and BOPP for avoiding obstacles is shown in Figure 3.

3.2 Localization accuracy

The future model's Behaviour can be predicted by comparing it to existing models. The efficiency comes from utilizing 100 UN nodes. The implementation parameters and their values are listed in Table 1. Having precise localization is essential for effective route planning. Accuracy is enhanced by comparing it to other algorithms. To determine this, the localization error is used. Improved precision is the result of reduced localization errors. The localization accuracy is measured by calculating the average localization error across all nodes in a specific part of the network.

Table 1. Parameters with value

Parameter	Value
Network Size(A)	100 X100sq.m
No of Nodes (N)	100
Distance(d_{max})	35 m, 70 m, 105 m, 140 m, 175 m`
Max Iteration (t)	60-300
Resolution ®	1

3.3 Computation time

Computation time is measured in terms of how long it takes to run the algorithms. A computer with an X64-based processor and Windows 8.1 is used for this simulation. In both cases, 50 iterations are the maximum allowed. The values used come from Table 1, with the exception of R, which is set to 1. The parameters used are 100 UNs, dmax = 140, and a default value of 1. Therefore, BOPP has a faster computational speed than IGOPP. The computation time grows proportionally with the value of tmax. Both the IGOPP and BOPP models' respective computation times are displayed in Figure 4.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5948
***Article Received**: 07 October 2022 **Revised**: 16 November 2022 **Accepted**: 21 December 2022*

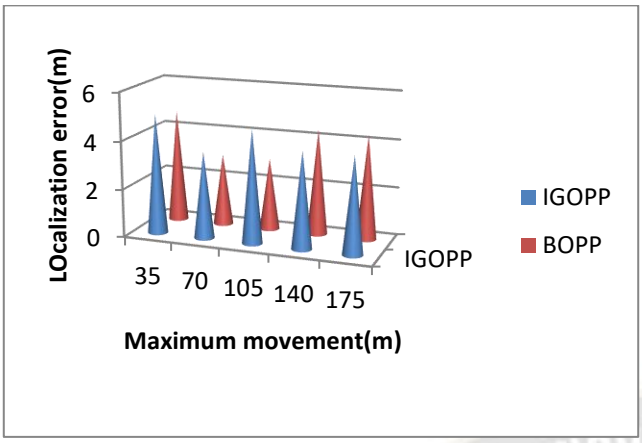

Figure3 : Mean localization error versus maximum movement for IGOPP and BOPP

3.4 Average Computation Time

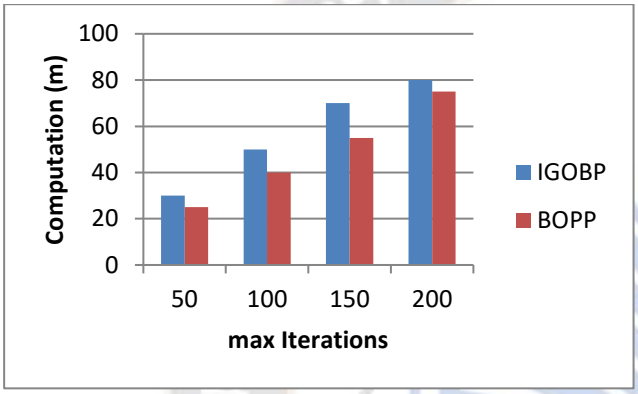

Figure 4. IGOPP and BOPP models' respective computation times

V. Concussion

The purpose of this proposed method is to enhance localization within a wireless sensor network. The localization error can be decreased by using the multi-objective strategy presented here. Both the IGOPP and BOPP dynamic obstacle avoidance route planning models have been designed for use in mobile anchor-assisted localization. Dijkstra's algorithm is used for path planning to find the shortest route, and the suggested model optimizes the path based on the most up-to-date information from the area. Both optimization methods steer clear of the obstacle in MA's way while also plotting the most direct route possible between origin and destination. Our final results show that the IGOPP and BOPP models we proposed are superior to the state-of-the-art in terms of accuracy, computation time, and localization ratio. On the other hand, the proposed model can only optimize localization with static barriers. In the future, we plan to investigate the system with movable obstructions, which will provide the MA with the ability to direct its movements and create complex patterns of motion across space.

References

[1]. Xiao, Shuo& Li, Tianxu& Tang, Chaogang& Cao, Yuan. (2019). Coverage Adaptive Optimization Algorithm of Static-Sensor Networks for Target Discovery. Chinese Journal of Electronics. 28. 398-403. 10.1049/cje.2018.02.009.

[2]. Xiao, Shuo& Li, Tianxu& Yan, Yan & Zhuang, Jiayu. (2019). Compressed sensing in wireless sensor networks under complex conditions of Internet of things. Cluster Computing. 22. 10.1007/s10586-018-2259-z.

[3]. Yedukondalu, Gangolu& K., Samunnisa &Bhavsingh, M. & Raghuram, I & Lavanya, Addepalli. (2022). MOCF: A Multi-Objective Clustering Framework using an Improved Particle Swarm Optimization Algorithm. International Journal on Recent and Innovation Trends in Computing and Communication. 10. 143-154. 10.17762/ijritcc.v10i10.5743.

[4]. Pasha, M. &Pingili, Madhavi & Sreenivasulu, K. & MALOTH, BHAV SINGH & Saheb, Shaik & Saleh, Alaa. (2022). Bug2 algorithm-based data fusion using mobile element for IoT-enabled wireless sensor networks. Measurement: Sensors. 24. 100548. 10.1016/j.measen.2022.100548.

[5]. Panwar, Anita & Kumar, Sh. (2012). Localization Schemes in Wireless Sensor Networks. Proceedings - 2012 2nd International Conference on Advanced Computing and Communication Technologies, ACCT 2012. 443-449. 10.1109/ACCT.2012.67.

[6]. M. R. Morelande, B. Moran and M. Brazil, "Bayesian node localisation in wireless sensor networks", Proc. IEEE Int. Conf. Acoust. Speech Signal Process., pp. 2545-2548, Mar. 2008.

[7]. C. Musso, N. Oudjane and F. Le Gland, "Improving regularised particle filters" in Sequential Monte Carlo Methods in Practice, Cham, Switzerland:Springer, pp. 247-271, 2001.

[8]. S. Gharghan, R. Nordin and M. Ismail, "A wireless sensor network with soft computing localization techniques for track cycling applications", Sensors, vol. 16, no. 8, pp. 1043, Aug. 2016.

[9]. H. Ahmadi and R. Bouallegue, "Exploiting machine learning strategies and RSSI for localization in wireless sensor networks: A survey", Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC), pp. 1150-1154, Jun. 2017.

[10]. M. A. Bhatti, R. Riaz, S. S. Rizvi, S. Shokat, F. Riaz and S. J. Kwon, "Outlier detection in indoor localization and Internet of Things (IoT) using machine learning", J. Commun. Netw., vol. 22, no. 3, pp. 236-243, Jun. 2020.

[11]. L. Wang, M. J. Er and S. Zhang, "A kernel extreme learning machines algorithm for node localization in wireless sensor networks", IEEE Commun. Lett., vol. 24, no. 7, pp. 1433-1436, Jul. 2020.

[12]. Alrajeh, N.A., Bashir, M. and Shams, B. (2013) 'Localization techniques in wireless sensor networks', International Journal of

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5948
***Article Received**: 07 October 2022 **Revised**: 16 November 2022 **Accepted**: 21 December 2022*

Distributed Sensor Networks, Vol. 9, No. 6, pp.1–9.

[13]. Prasanna, CH.Harika (2015). Automatic and Systematic Methodology for Test Packet Generation. International Journal of Computer Engineering In Research Trends. 2(12),pp 1173-1177.

[14]. Coluccia, Angelo & Fascista, Alessio & Schumann, Arne & Sommer, Lars &Dimou, Anastasios &Zarpalas, Dimitrios& Méndez, Miguel &Iglesia, David & González, Iago & Mercier, Jean-Philippe &Gagné, Guillaume & Mitra, Arka&Rajashekar, Shobha. (2021). Drone vs. Bird Detection: Deep Learning Algorithms and Results from a Grand Challenge. Sensors. 21. 2824. 10.3390/s21082824.

[15]. L.Sravanthi,B.RanjithKumar(2015). A Survey on Fault Identification Using Automatic Test Packet Generation .International Journal of Computer Engineering In Research Trends. 2(10),pp 871-874.

[16]. SHAHID, Muhammad & MALIK, Tahir & SAID, Ahsan. (2021). Heuristic based binary grasshopper optimization algorithm to solve unit commitment problem. TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES. 29. 944-961. 10.3906/elk-2004-144.

[17]. Arora, Sankalap & Singh, Satvir. (2016). An Improved Butterfly Optimization Algorithm for Global Optimization. Advanced Science, Engineering and Medicine. 8. 711-717. 10.1166/asem.2016.1904.

[18]. Kaushik, Abhinesh & Lobiyal, Daya. (2021). Localization in Wireless Sensor Networks Using a Mobile Anchor and Subordinate Nodes. 10.1007/978-981-33-6173-7_12.