# High Precision Distance Vector Hop Localization Algorithm for Wireless Sensor Networks

[1]Mohammed Aboud Kadhim, [2]Kamal Y. Kamal
Middle Technical University,
Baghdad, Iraq

*Abstract*—**Internet of Things (IoT) systems require localization for the nodes within Wireless Sensor Networks (WSNs) for many location-based services. Since thousands of sensor nodes would exist in some networks, having GPS on each node is impractical, not merely due to the hardware cost, but also because of the poor performance indoors. Localization is now recognized as a crucial area for study. The research presented in this paper puts forward a half-measure weighted centroid DV-Hop localization algorithm. The proposed algorithm adjusts the locations of unknown nodes using redundant information obtained from localization equations. Simulation of sensor networks yielded significant improvements in accuracy and reduced error rates in localization estimation, while maintaining low hardware and computational costs.**

***Index Terms*—DV-Hop, WSN, IoT, localization.**

## I. INTRODUCTION

Wireless Sensor Networks (WSNs) are utilized for various applications, and among the significant challenges for WSNs is the physical localization of nodes; sufficient localization is required for position-based WSN applications, such as target tracking, path planning, event detection, and data routing [1], [2]. Localization algorithms are usually implemented to determine the physical coordinates of sensor nodes within a WSN. One of the many localization solutions for wireless sensor networks is the Distance Vector Hop (DV-Hop) algorithm. Its cost-effectiveness and ease have made it extremely popular [3]. However, the DV-Hop algorithm suffers from inherent errors in distance estimation; thus, researchers have developed various enhancements to improve the DV-Hop algorithm's ability to determine node positions [4]. Some add gadgets that measure distance or antennas that point in specific directions [5]. Others use machine learning to make networks learn on their own [6], [7], [8]. Such approaches involve processing complex computations, which introduce overheads that may not always be compatible with wireless networks that have limited computational resources. The DV-Hop, as a decentralized range-free localization algorithm, was initially proposed by Niculescu et al. [9] and has garnered significant attention due to its ease of use and minimal hardware constraints for sensor nodes within a WSN. Conversely, the algorithm has the disadvantage of measuring hop distances rather than straight-line distances, which results in higher range error rates. Li et al. focused on adjusting the shortest distance of "zigzag" paths using geometric topographies, but did not take into account the density distribution of the nodes [10]. Tang et al. attempted to correct the average hop distance of the anchor nodes but did not address the error in estimating the positions of the unknown nodes [11]. Shahzad et al. eliminated some distant anchors that are away from unknown nodes to ease the computation process, but ignored some anchors' useful information [12]. Cai et al. tried to improve accuracy through a weight model but overlooked the weight-error relation [13].

## II. BACKGROUND

Localization in wireless networks represents an important area of research in which localization algorithms consider each sensor node in the affected environment as a physical landmark. Some position-based applications require location-related information from sensor nodes to determine an accurate position.

Sometimes, sensor nodes operate in remote and harsh environments, such as disaster relief areas and forests; in these cases, traditional location tracking methods and simple sensor node localization approaches may not yield a feasible solution. Therefore, efficient localization methods are adapted to provide accurate positional information for the sensor nodes. Data aggregation strategies also require the positional information of the sensor nodes, and the collected data from various sensors would be useless if accurate information on the sensor nodes is not available. Localization is a process by which sensor nodes can determine their locations. Cyber-physical systems, eHealth, environmental monitoring, indoor automation, automated path planning, and weather forecasting are examples of the vast range of applications that require location-based services [14], [15]. Global positioning systems (GPS) are widely used to determine the locations of nodes; however, they incur a high cost in terms of power consumption, and their performance is also known to be poor indoors. Over the past decade, the scientific community has recognized the importance of this topic, leading to a substantial amount of research in this field. Localization is the process of determining the position of a node within a network of sensors. When the node's positions are unknown, connectivity information is utilized to determine the localization of the unknown nodes. Localization techniques can be classified into two basic categories: "Target or Source Localization" and "Node or Self Localization". Target/Source localization is further localized into "Single-Target Localization" and "Multiple-Target Localization".

On the other hand, "Node/Self Localization" is further classified into "Range-Based Localization" and "Range-Free Localization". Furthermore, "Range-Based Localization" is subclassified into more specific algorithms, such as "Connectivity", "Centroid", "Energy", and "Region Overlap" localization algorithms. The "Range-Based" localization algorithms use distance measurement techniques to calculate the location of unknown nodes; alternatively, "Range-Free" localization algorithms use the contents of the messages rather than measuring the proximity in terms of hop count or estimated distance to landmarking sensor nodes with known locations [16], "cooperative" localization techniques which require the existence of communication among all nodes [17], and "Non-Cooperative" localization techniques, where the unknown nodes communicate only with the anchor nodes [18], "Centralized" localization technique, which is also known as "Network-Centric Positioning", and "Distributed" localization technique, which is also known as "self-positioning" where no central management for the determination of the nodes' position; instead, each node estimates its location based on its local information [19]. Recent studies have investigated the mobility effects on localization [20], [21], [22], "Anchor-Based" and "Anchor-Free" localization [23], [24]. The methods used to estimate the location of sensor nodes are: The "Range-Based Localization" algorithms usually adopt one of the following techniques to measure a distance: "Angle of Arrival (AOA)", "Time Difference of Arrival (TDOA)", or "Received Signal Strength Indicator (RSSI)". The "Range-Free Localization" algorithms do not measure distance or angle among nodes. These algorithms can be divided into "Local/Pattern-Match" and "Hop-Count" localization techniques.

The DV-Hop is a range-free algorithm that estimates the distance between sensor nodes based on the hop count; it goes through three steps: 1) counting the minimum number of hops between the unknown node and the anchor; 2) estimate the distance from anchors to the unknown node by multiplying the minimum number of hops and the average distance per hop. and 3) determining the unknown node's coordinates mostly based the trilateration method or probability evaluation [31], [32], [33], [34], [35]. Many researchers have conducted extensive studies on improving the DV-Hop algorithm [25], [26], [27], [28], [29], [30]. Some research has focused on optimizing steps (1 and 2) of the DV-Hop algorithm; for example, Li et al. optimized the minimum number of hops by setting two communication radii and increasing the communication radius (R) [36]. Gui et al. proposed a medium access control (MAC) method based on the "Chinese residual set (CRT)" protocol sequence to localize DV-Hops [37]. Kaur et al. developed the "Enhanced Weighted Centroid DV-Hop (EWCL)" algorithm to address the issues of poor accuracy and excessive power consumption. The transmission radius determined the EWCL algorithm's weighting factor, average hop distance, and hop count [38]. Other scientists have also optimized step (3) of DV-Hop. Recently, due to the impressive performance of intelligent computing on complex optimization problems, several nature-inspired schemes have been introduced [13], [39]. For example, Zhou et al. optimized DV-Hop based on bacterial foraging optimization (BFO) [40]. Kaur et al. optimized DV-Hop algorithm based on the "Gray-Wolf" optimization for 2D and 3D WSNs environments [41]. Song et al. applied "Firefly Swarm Optimization" algorithm [42] with a chaotic inertial weight update [43]. Cui et al. proposed the CS-DV-Hop algorithm [44], a hybrid DV-Hop algorithm that combines the “Cuckoo Search” optimization algorithm [45]. Wang et al. proposed an improved DV-Hop algorithm based on both BFO and glow-worm swarm optimization (GSO), thus it is called “BFO-GSO.” This approach has proven to have a convergence speed; however, its computing time has increased slightly compared to merely BFO [46].

## III. HD-DV-HOP ALGORITHM

The half-distance DV-Hop routing protocol is the decentralized range-free localization approach utilized by the DV-Hop localization method. The essential idea involves determining the distances between signal and obscure hubs by replicating the typical hop distance in WSNs, with the bounce considered a fundamental part of reference point hubs. Three mathematical techniques are typically used to determine a receiver’s position (en) from signals received from multiple transmitters: triangulation, trilateration, and multilateration. When employing the DV-Hop algorithm, localization errors may occur because paths between beacons and unknown nodes may not be direct in a network with randomly positioned wireless sensor nodes. Moreover, the more hops there are, the larger the accumulated errors are. In the initial step, each anchor hub conveys its coordinates as a reference point to the other nodes within the network, containing the anchor’s area with a hop count of one. Each hub tracks the base bounce count per anchor for all signals it receives. Reference points with higher hop count values associated with a specific anchor are considered outdated and will be disregarded. Then, those not flat guides overflowed outward, with bounce count values augmented at each middle-of-the-road hop. Through this component, all hubs in the organization get a negligible bounce build-up to each secure hub. In the next phase, when an anchor receives hop counts from various anchors, it calculates an average hop size and shares it with the entire network. Nodes without visual information use the hop size multiplied by the hop count to calculate the distance from the anchor. The average hop size to an anchor $i$ is estimated as:

$$H_{HopSize} = \frac{\sum_{j=1}^{J}(x_i - x_j)^2 + (y_i - y_j)^2}{\sum_{j=1}^{J} h_{ij}} \tag{1}$$

Where $(x_i, y_i)$, $(x_j, y_j)$ are coordinates of anchor $i$ and anchor $j$, $h_{ij}$ denotes the hops between beacons $i$ and $j$. Each anchor node transmits the information about its hop size to the network using “controlled flooding”. Unknown nodes receive hop size data and save the first one; they also transmit their hop size to neighboring nodes. This method enables nodes to receive the hop size from the beacon node with the minimum number of hops. Ultimately, the unknown nodes determine the distance to beacon nodes based on the hop lengths to those beacon nodes.

Each anchor node shares its hop size with the network during controlled flooding. When an unknown hub gets this hop-size data, it stores it first and then passes it to its adjoining hubs. This method ensures that most nodes get the hop size from the beacon node with the fewest hops to them. As a result, based on the hop lengths to the beacon nodes, unknown nodes ultimately establish the distances to the beacon nodes.

Let $(x, y)$ denote the unknown coordinates of node $D$, $(x_i, y_i)$ are the known coordinates of the $i^{th}$ receiver anchor node, and $i^{th}$ anchor node distance to node $D$ is $d_i$, then we can compute the unknown coordinates of node $D$ as:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ \qquad \vdots \\ (x - x_i)^2 + (y - y_i)^2 = d_i^2 \end{cases} \tag{2}$$

The coordinates are computed as:

$$A = -2 \times \begin{bmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \vdots & \vdots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{bmatrix} \tag{3}$$

$$B = \begin{bmatrix} d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_2^2 - d_n^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix} \tag{4}$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix} \tag{5}$$

Where: $P = (A^T A)^{-1} A^T B$

## HD-DV-Hop Algorithm

### Inputs:

```
- Set of anchor nodes A = {a1, a2, ..., an}
- Set of unknown nodes U = {u1, u2, ..., um}
- Location coordinates of anchor nodes (x_a, y_a) for a ∈ A
```

### Output:

```
- Estimated location coordinates (x_u, y_u) for u ∈ U
```

### Initialization:

```
For each anchor node a ∈ A:
  Broadcast the location coordinates (x_a, y_a)
for each unknown node u ∈ U:
  Calculate hop count to each anchor node h(u, a)
```

### Distance Estimation:

```
for each anchor node a ∈ A:
  Sum_dist = 0
  Sum_hops = 0
  for each anchor node b ∈ A, b ≠ a:
    Sum_dist += distance((x_a, y_a), (x_b, y_b))
    Sum_hops += h(a, b)
  HopSize(a) = Sum_dist / Sum_hops
```

**Coordinate Estimation:**

```
  for each unknown node u ∈ U:
    Distance_estimates = []
    for each anchor node a ∈ A:
      Estimated_distance = h(u, a) × HopSize(a)
      Distance_estimates.append(Estimated_distance)
    (x_u, y_u) = Multilateration(Distance_estimates, anchor_locations)
Refinement (Optional):
  while stopping criteria not met:
    for each unknown node u ∈ U with estimated coordinates (x_u, y_u):
      Treat u as a pseudo-anchor node
      Broadcast (x_u, y_u) to other nodes
    Repeat Distance Estimation and Coordinate Estimation steps
    for the remaining unknown nodes using pseudo-anchor nodes
Return estimated coordinates (x_u, y_u) for all u ∈ U
```

**Explanation:**

Initialization: Anchor nodes broadcast their location coordinates, and unknown nodes calculate their hop counts to each anchor node.

Distance Estimation: For each anchor node, the average hop distance (HopSize) is calculated by dividing the sum of distances between the anchor node and other anchor nodes by the sum of their respective hop counts.

Coordinate Estimation: For each unknown node, the estimated distance from each anchor node is calculated by multiplying the hop count to that anchor node with the corresponding average hop distance (HopSize). Using these estimated distances and the known locations of anchor nodes, the unknown node's coordinates are calculated using a multilateration technique, such as the least-squares method.

Refinement (Optional): In this optional step, the algorithm can be iteratively refined by treating the unknown nodes with estimated coordinates as pseudo-anchor nodes. The Distance Estimation and Coordinate Estimation steps are repeated for the remaining unknown nodes, using both the pseudo-anchor nodes and the original anchor nodes.

The algorithm continues to refine the coordinates until a stopping criterion is met, such as a maximum number of iterations or a desired accuracy threshold.

## IV. SIMULATION AND DATA ANALYSIS

In this section, we compare and analyze the outcomes at a system level; the simulation was carried out with MATLAB software to assess the sufficiency of the proposed algorithm, in terms of efficiency and suitability. We randomly distributed 100 wireless sensor nodes within a 100 m x 100 m square; 60 nodes are unknown, 40 of which are beacons.. The half-weighted centroid shapes the premise of the organization geography utilized by the DV-Hop bounce confinement calculation. The estimated distances between the unknown and the beacon nodes were calculated by multiplying the least hop distance by the average hop distance between the two nodes. The communication radius of the nodes plays a significant role in determining the minimum amount of hops between nodes. Node locations also vary depending on the hop distances between communication radii. The correspondence range is more modest, and the geography is nearer to the actual area, which has higher hub densities, as shown in Figures 1-10. By optimizing the network's topology, network longevity can also be extended, and energy consumption can decrease.

We compared the localization errors of the proposed algorithm over an identical communication radius (R = 100 m). The DV-Hop algorithm's node localization error (e) fluctuated between 15 and 60. In contrast, the proposed algorithm's error rate ranged between 2 and 15, so it is more precise, mainly attributed to the increasing radiuses of unknown nodes, which reduced the minimum communication radius, as illustrated in Figures (10-20).

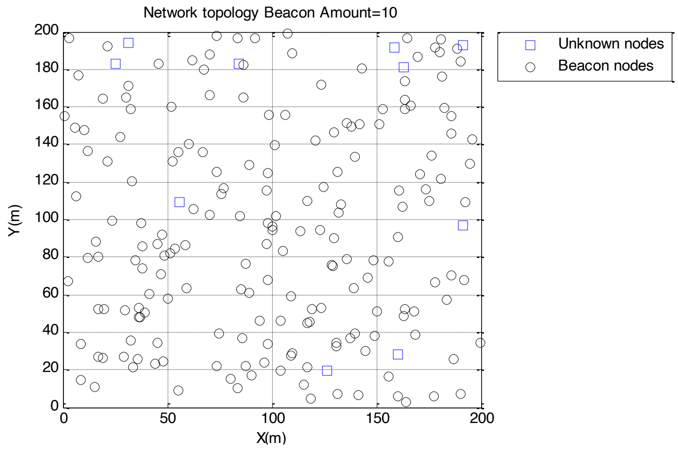


Figure (1): Network topology (10 Beacons)

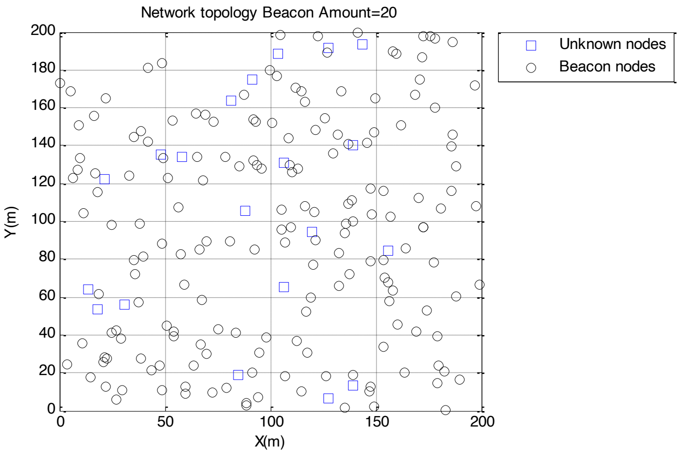


Figure (2): Network topology (20 Beacons)

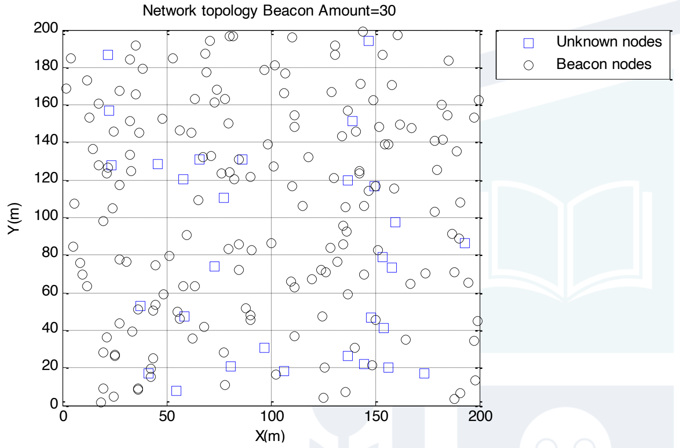


Figure (3): Network topology (30 Beacons)

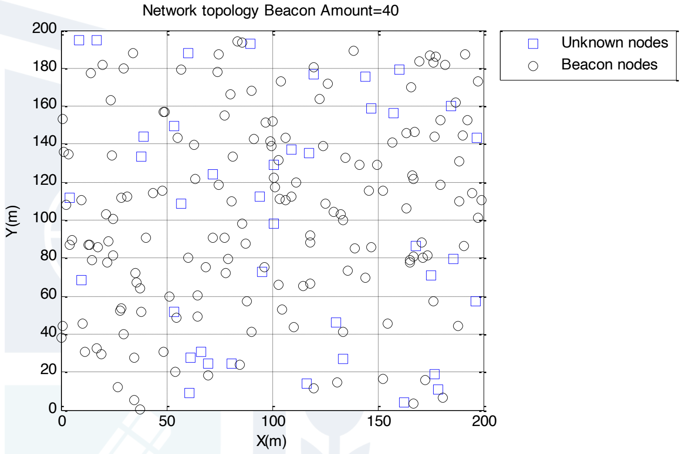


Figure (4): Network topology (40 Beacons)

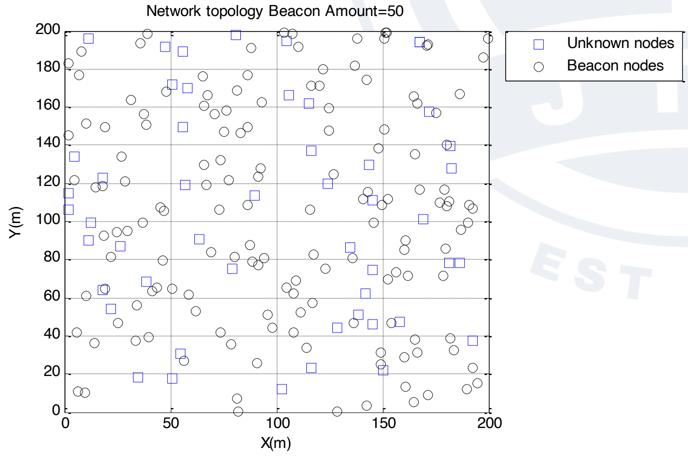


Figure (5): Network topology (50 Beacons)

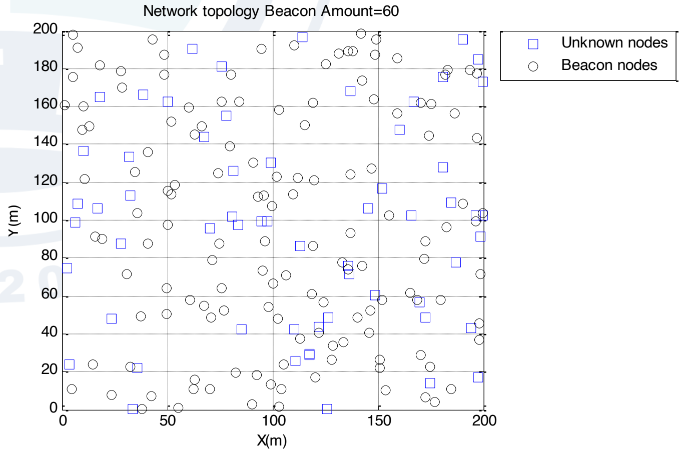


Figure (6): Network topology (60 Beacons)

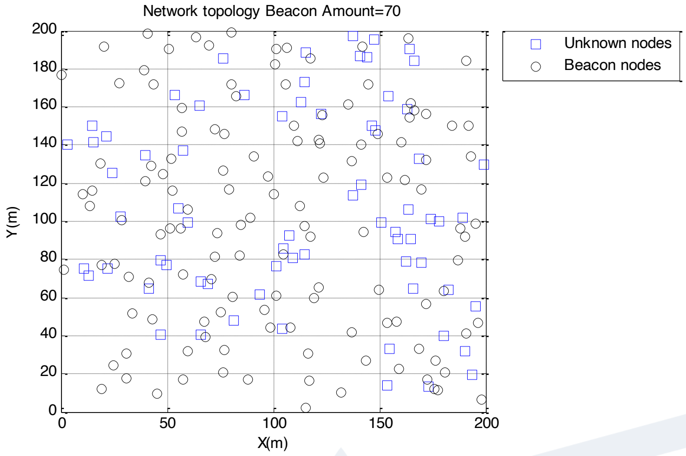


Figure (7): Network topology (70 Beacons)

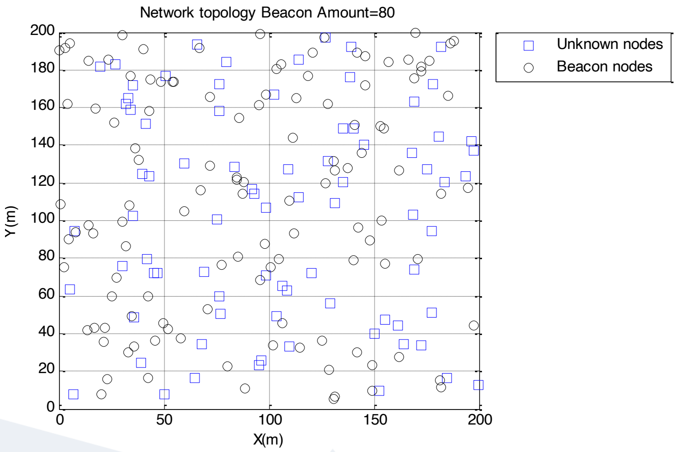


Figure (8) Network topology (80 Beacons)

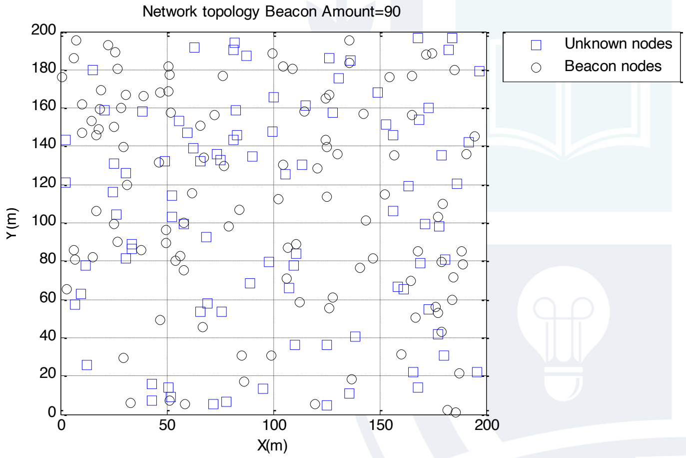


Figure (9): Network topology (90 Beacons)

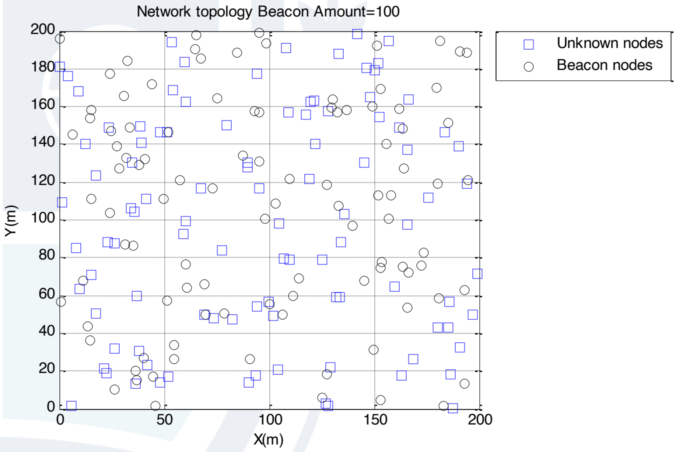
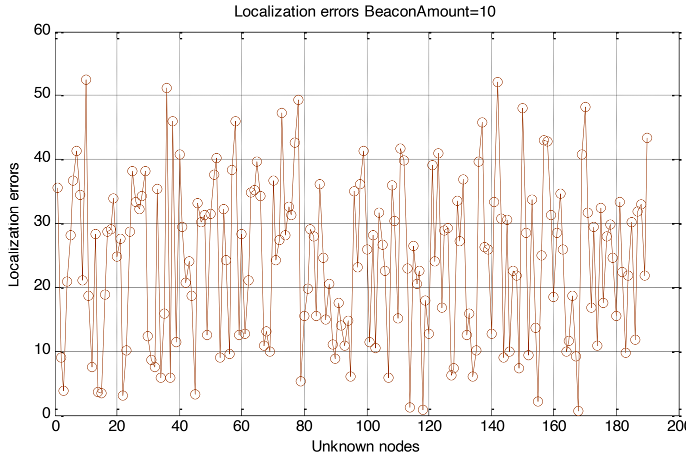


Figure (10): Network topology (100 Beacons)



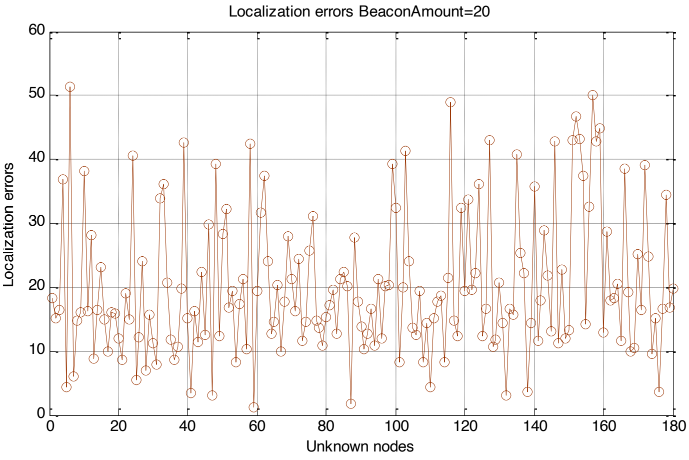
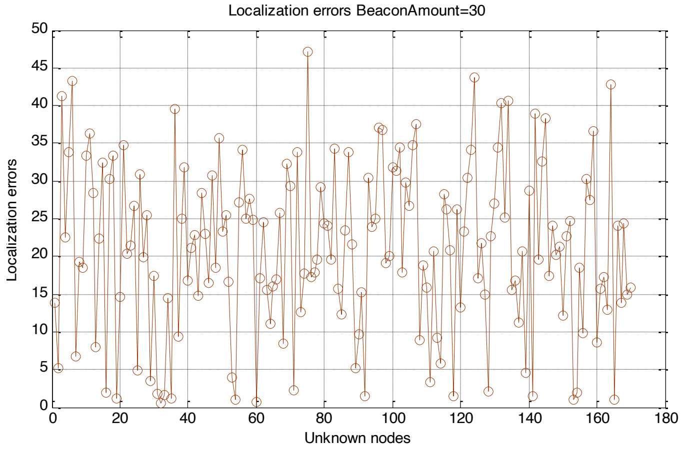


Figure (11): Localization errors (10 Beacons)

Figure (12): Localization errors over communication radius (20 Beacons)



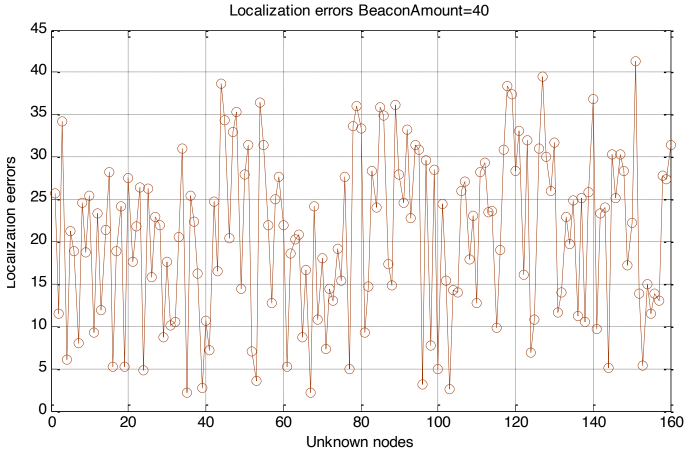
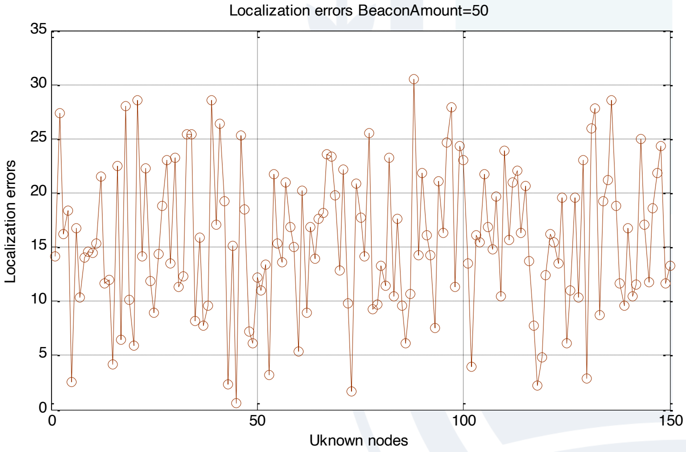


Figure (13): Localization errors over communication radius (30 Beacons)

Figure (14): Localization errors over communication radius (40 Beacons)



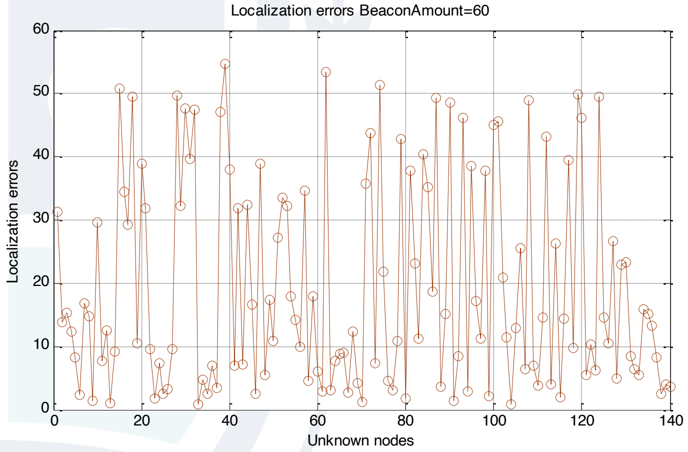
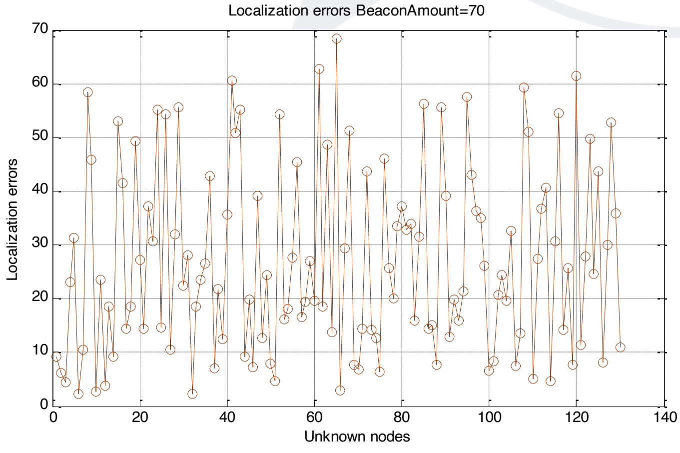


Figure (15): Localization errors over communication radius (50 Beacons)

Figure (16): Localization errors over communication radius (60 Beacons)



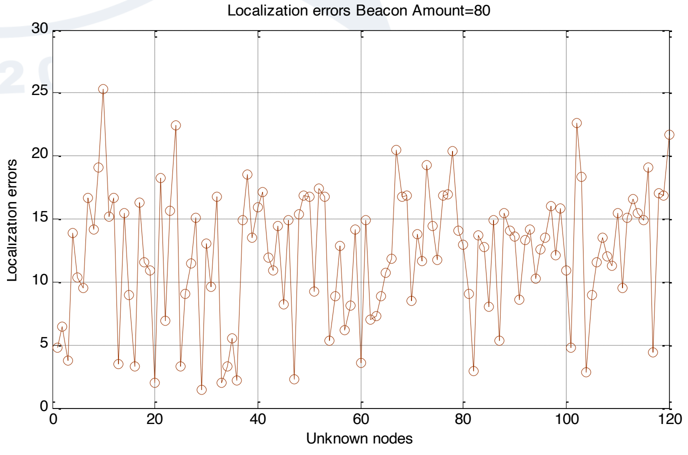


Figure (17): Localization errors over communication radius (70 Beacons)

Figure (18): Localization errors over communication radius (80 Beacons)

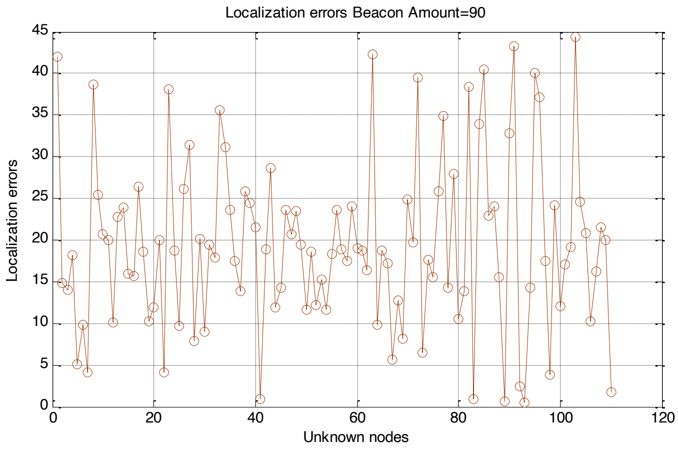


Figure (19): Localization errors over communication radius (90 Beacons)

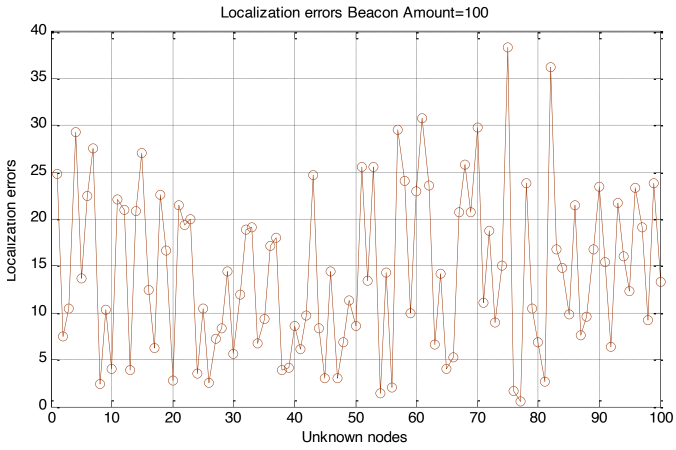


Figure (20): Localization errors over communication radius (100 Beacons)

## V. CONCLUSION

An enhancement algorithm for WSNs localization was proposed half-measure weighted centroid DV-Hop algorithm, where the uneven distribution of nodes, holes, and errors in the average hop distance was simulated. The new algorithm used a cutting-edge localization strategy by adding a half-measure weighted centroid with the DV-Hop algorithm. Beacon nodes use the centroid algorithm to determine their location, and the DV-Hop localized accuracy is then used as the weight for locating unknown nodes. The enhanced localization algorithm decreased localization error and increased the localization accuracy of unknown nodes. The simulation outcomes indicate that a higher location coverage and a lower error rate are associated with more frequently placed beacons. In the application scenario, anchor nodes can be manually placed, and location performance can be enhanced. Since the study was conducted in a perfect network simulation environment, future research will be required to examine how the enhanced algorithm might be used in less typical network environments.

## REFERENCES

[1] G. Pettorru, V. Pilloni, and M. Martalò, "Trustworthy Localization in IoT Networks: A Survey of Localization Techniques, Threats, and Mitigation," *Sensors*, vol. 24, no. 7, Art. no. 7, Jan. 2024, doi: 10.3390/s24072214.

[2] W. Liu, G. Wei, and M. Zhu, "A survey on multi-dimensional path planning method for mobile anchor node localization in wireless sensor networks," *Ad Hoc Networks*, vol. 156, p. 103416, Apr. 2024, doi: 10.1016/j.adhoc.2024.103416.

[3] J. Ren, P. Qi, C. Li, P. Zhu, and Z. Li, "Multisource sparse inversion localization with long-distance mobile sensors," *Electronics*, vol. 13, no. 6, Art. no. 6, Jan. 2024, doi: 10.3390/electronics13061024.

[4] V. C. S. R. Rayavarapu and A. Mahapatro, "MOANS DV-Hop: An anchor node subset based localization algorithm for wireless sensor networks," *Ad Hoc Networks*, vol. 152, p. 103323, 2024.

[5] Y. Hu and X. Li, "An improvement of DV-hop localization algorithm for wireless sensor networks," *Telecommun Syst*, vol. 53, no. 1, pp. 13–18, May 2013, doi: 10.1007/s11235-013-9671-8.

[6] H. Lee, C. Wu, and H. Aghajan, "Vision-based user-centric light control for smart environments," *Pervasive and Mobile Computing*, vol. 7, no. 2, pp. 223–240, Apr. 2011, doi: 10.1016/j.pmcj.2010.08.003.

[7] P. Yadav and S. C. Sharma, "An efficient optimal localization technique for WSN using hybrid machine learning algorithms," *Wireless Pers Commun*, vol. 133, no. 4, pp. 2601–2639, Dec. 2023, doi: 10.1007/s11277-024-10892-z.

[8] "Intelligent wireless sensing driven metaverse: A survey," *Computer Communications*, vol. 214, pp. 46–56, Jan. 2024, doi: 10.1016/j.comcom.2023.11.024.

[9] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, in MobiCom '07. New York, NY, USA: Association for Computing Machinery, Sep. 2007, pp. 51–62. doi: 10.1145/1287853.1287861.

[10] Q. Tang and J. Wang, "An improved DV-Hop localization algorithm for wireless sensor network based on TDOA quantization," in *2017 International Conference on Network and Information Systems for Computers (ICNISC)*, Apr. 2017, pp. 19–24. doi: 10.1109/ICNISC.2017.00013.

[11] F. Shahzad, T. R. Sheltami, and E. M. Shakshuki, "DV-maxHop: A Fast and accurate range-free localization algorithm for anisotropic wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2494–2505, Sep. 2017, doi: 10.1109/TMC.2016.2632715.

[12] X. Cai, H. Wang, Z. Cui, J. Cai, Y. Xue, and L. Wang, “Bat algorithm with triangle-flipping strategy for numerical optimization,” *Int. J. Mach. Learn. & Cyber.*, vol. 9, no. 2, pp. 199–215, Feb. 2018, doi: 10.1007/s13042-017-0739-8.

[13] C. Wu, Z. Yang, and Y. Liu, “Smartphones based crowdsourcing for indoor localization,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 444–457, Feb. 2015, doi: 10.1109/TMC.2014.2320254.

[14] M. Wang and C. Huang, “Mobile anchor node assisted node collaborative localization based on light reflection in WSN,” *Wireless Netw*, vol. 30, no. 4, pp. 2801–2818, May 2024, doi: 10.1007/s11276-024-03701-9.

[15] Y. V. Lakshmi, P. Singh, S. Mahajan, A. Nayyar, and M. Abouhawwash, “Accurate range-free localization with hybrid DV-hop algorithms based on PSO for UWB wireless sensor networks,” *Arab J Sci Eng*, vol. 49, no. 3, pp. 4157–4178, Mar. 2024, doi: 10.1007/s13369-023-08287-6.

[16] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative localization in wireless networks,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009, doi: 10.1109/JPROC.2008.2008853.

[17] S. Tomic, M. Beko, and R. Dinis, “RSS-based localization in wireless sensor networks using convex relaxation: noncooperative and cooperative schemes,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 2037–2050, May 2015, doi: 10.1109/TVT.2014.2334397.

[18] A. Kaushik *et al.*, “Toward integrated sensing and communications for 6G: Key enabling technologies, standardization, and challenges,” *IEEE Communications Standards Magazine*, vol. 8, no. 2, pp. 52–59, Jun. 2024, doi: 10.1109/MCOMSTD.0007.2300043.

[19] X. Wang, Y. Liu, Z. Yang, K. Lu, and J. Luo, “Robust component-based localizationin sparse networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1317–1327, May 2014, doi: 10.1109/TPDS.2013.85.

[20] S. Salari, S. Shahbazpanahi, and K. Ozdemir, “Mobility-aided wireless sensor network localization via semidefinite programming,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 12, pp. 5966–5978, Dec. 2013, doi: 10.1109/TWC.2013.110813.120379.

[21] B.-F. Wu and C.-L. Jen, “Particle-filter-based radio localization for mobile robots in the environments with low-density wlan aps,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 6860–6870, Dec. 2014, doi: 10.1109/TIE.2014.2327553.

[22] N. Zaarour, N. Hakem, and N. Kandil, “An accurate anchor-free contextual received signal strength approach localization in a wireless sensor network,” *Sensors*, vol. 24, no. 4, Art. no. 4, Jan. 2024, doi: 10.3390/s24041210.

[23] R. N. Biswas, A. Saha, S. K. Mitra, and M. K. Naskar, “Design and implementation of anchor coprocessor architecture for wireless node localization applications,” *Peer-to-Peer Netw. Appl.*, vol. 17, no. 2, pp. 961–984, Mar. 2024, doi: 10.1007/s12083-024-01640-y.

[24] F. Thomas and L. Ros, “Revisiting trilateration for robot localization,” *IEEE Trans. Robot.*, vol. 21, no. 1, pp. 93–101, Feb. 2005, doi: 10.1109/TRO.2004.833793.

[25] J. Li, X. Yue, J. Chen, and F. Deng, “A novel robust trilateration method applied to ultra-wide bandwidth location systems,” *Sensors*, vol. 17, no. 4, Art. no. 4, Apr. 2017, doi: 10.3390/s17040795.

[26] A. E. M. El Ashry and B. I. Sheta, “Wi-Fi based indoor localization using trilateration and fingerprinting methods,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 610, no. 1, p. 012072, Sep. 2019, doi: 10.1088/1757-899X/610/1/012072.

[27] T. Yang, A. Cabani, and H. Chafouk, “A survey of recent indoor localization scenarios and methodologies,” *Sensors (Basel)*, vol. 21, no. 23, p. 8086, Dec. 2021, doi: 10.3390/s21238086.

[28] Q. Luo, K. Yang, X. Yan, J. Li, C. Wang, and Z. Zhou, “An improved trilateration positioning algorithm with anchor node combination and K-Means clustering,” *Sensors (Basel)*, vol. 22, no. 16, p. 6085, Aug. 2022, doi: 10.3390/s22166085.

[29] Y. Cao and Z. Wang, “Improved DV-hop localization algorithm based on dynamic anchor node set for wireless sensor networks,” *IEEE Access*, vol. 7, pp. 124876–124890, 2019, doi: 10.1109/ACCESS.2019.2938558.

[30] G. Li, S. Zhao, J. Wu, C. Li, and Y. Liu, “DV-hop localization algorithm based on minimum mean square error in Internet of Things,” *Procedia Computer Science*, vol. 147, pp. 458–462, Jan. 2019, doi: 10.1016/j.procs.2019.01.272.

[31] G. Song and D. Tam, “Two novel DV-hop localization algorithms for randomly deployed wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 7, p. 187670, Jul. 2015, doi: 10.1155/2015/187670.

[32] X. Li, K. Wang, B. Liu, J. Xiao, and S. Han, “An improved range-free location algorithm for industrial wireless sensor networks,” *J Wireless Com Network*, vol. 2020, no. 1, p. 81, Apr. 2020, doi: 10.1186/s13638-020-01698-1.

[33] W. Yu and H. Li, “An improved DV-Hop localization method in Wireless Sensor Networks,” in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, May 2012, pp. 199–202. doi: 10.1109/CSAE.2012.6272938.

[34] D. Prashar, K. Jyoti, and D. Kumar, “Design and analysis of distance error correction–based localization algorithm for wireless sensor networks,” *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 12, p. e3547, 2018, doi: 10.1002/ett.3547.

[35] T. Li, C. Wang, and Q. Na, “Research on DV-Hop improved algorithm based on dual communication radius,” *J Wireless Com Network*, vol. 2020, no. 1, p. 113, Jun. 2020, doi: 10.1186/s13638-020-01711-7.

[36] L. Gui *et al.*, “DV-Hop Localization with protocol sequence based access,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9972–9982, Oct. 2018, doi: 10.1109/TVT.2018.2864270.

[37] A. Kaur, P. Kumar, and G. P. Gupta, “A weighted centroid localization algorithm for randomly deployed wireless sensor networks,” *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 1, pp. 82–91, Jan. 2019, doi: 10.1016/j.jksuci.2017.01.007.

[38] Z. Cui, F. Li, and W. Zhang, “Bat algorithm with principal component analysis,” *Int. J. Mach. Learn. & Cyber.*, vol. 10, no. 3, pp. 603–622, Mar. 2019, doi: 10.1007/s13042-018-0888-4.

[39] “DV-Hop localization algorithm based on bacterial foraging optimization for wireless multimedia sensor networks | Multimedia Tools and Applications.” Accessed: Jun. 11, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s11042-018-5674-5

[40] A. Kaur, P. Kumar, and G. P. Gupta, “Nature inspired algorithm-based improved variants of DV-Hop algorithm for randomly deployed 2D and 3D wireless sensor networks,” *Wireless Pers Commun*, vol. 101, no. 1, pp. 567–582, Jul. 2018, doi: 10.1007/s11277-018-5704-7.

[41] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd Ed. Frome, United Kingdom: Luniver Press, 2010.

[42] L. Song, L. Zhao, and J. Ye, “DV-Hop node location algorithm based on GSO in wireless sensor networks,” *Journal of Sensors*, vol. 2019, no. 1, p. 2986954, 2019, doi: 10.1155/2019/2986954.

[43] Z. Cui, B. Sun, G. Wang, Y. Xue, and J. Chen, “A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber–physical systems,” *Journal of Parallel and Distributed Computing*, vol. 103, pp. 42–52, May 2017, doi: 10.1016/j.jpdc.2016.10.011.

[44] X.-S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India: IEEE, Dec. 2009, pp. 210–214. doi: 10.1109/NABIC.2009.5393690.

[45] Y. Wang, P. Wang, J. Zhang, X. Cai, W. Li, and Y. Ma, “A novel DV-Hop method based on coupling algorithm used for wireless sensor network localisation,” *International Journal of Wireless and Mobile Computing*, vol. 16, no. 2, pp. 128–137, Jan. 2019, doi: 10.1504/IJWMC.2019.099027.