information

MDPI

Article

Range-Free Localization Approaches Based on Intelligent Swarm Optimization for Internet of Things

Abdelali Hadir [1,*], Naima Kaabouch [2], Mohammed-Alamine El Houssaini [3] and Jamal El Kafi [4]

[1] National School of Commerce and Management, Hassan II University, Casablanca 20250, Morocco
[2] School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks, ND 58202, USA; naima.kaabouch@und.edu
[3] Higher School of Education and Training, Chouaib Doukkali University, El Jadida 24000, Morocco; elhoussaini.m@ucd.ac.ma
[4] Faculty of Sciences, Chouaib Doukkali University, El Jadida 24000, Morocco; elkafi.j@ucd.ac.ma
* Correspondence: a.hadir@encgcasa.ma

Abstract: Recently, the precise location of sensor nodes has emerged as a significant challenge in the realm of Internet of Things (IoT) applications, including Wireless Sensor Networks (WSNs). The accurate determination of geographical coordinates for detected events holds pivotal importance in these applications. Despite DV-Hop gaining popularity due to its cost-effectiveness, feasibility, and lack of additional hardware requirements, it remains hindered by a relatively notable localization error. To overcome this limitation, our study introduces three new localization approaches that combine DV-Hop with Chicken Swarm Optimization (CSO). The primary objective is to improve the precision of DV-Hop-based approaches. In this paper, we compare the efficiency of the proposed localization algorithms with other existing approaches, including several algorithms based on Particle Swarm Optimization (PSO), while considering random network topologies. The simulation results validate the efficiency of our proposed algorithms. The proposed HW-DV-HopCSO algorithm achieves a considerable improvement in positioning accuracy compared to those of existing models.

Keywords: IoT; PSO; CSO; WSN; DV-Hop

check for updates

Citation: Hadir, A.; Kaabouch, N.; El Houssaini, M.-A.; El Kafi, J. Range-Free Localization Approaches Based on Intelligent Swarm Optimization for Internet of Things. *Information* **2023**, *14*, 592. https://doi.org/10.3390/info14110592

Academic Editor: Ruggero Lanotte

Received: 1 September 2023
Revised: 23 October 2023
Accepted: 29 October 2023
Published: 1 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The convergence of Micro-Electro-Mechanical Systems (MEMS) and the Internet of Things (IoT) has contributed to the development of tiny networked devices capable of detecting, monitoring, processing, and transmitting physical phenomena such as pressure and temperature [1,2]. RFID, Zigbee, and 5G are among the communication technologies used by these devices [3–7]. Each sensor node, which is associated with CPUs, power units, and transceivers, has the ability to transmit data to a central base station (BS) for analysis and interpretation [8]. These devices play an important role in WSNs, and are used in such diverse fields as military operations, target tracking, and environmental monitoring. Localization is a critical process in IoT applications such as smart cities, healthcare monitoring, traffic management, disaster alerts, and geographic routing [9–15]. Accurately determining a device's location allows for in-depth analysis of events reported by sensor nodes. The interpretation and meaningful understanding of detected events are significantly limited because of the lack of accurate localization. Figure 1 depicts the importance and necessity of localization in a variety of application fields. As illustrated by Figure 1, precisely determining a device's location is a key challenge for IoT applications.

GPS [16] is a widely recommended technology for accurately determining device locations in various application domains. However, due to the high cost of extra hardware, integrating auto-positioning technologies such as GPS into sensor devices may not consistently be the most cost-effective strategy. This has the potential to decrease the lifetime of the network. To address this concern, several localization techniques have been formulated revolving around the deployment of a limited set of GPS-equipped sensors referred to as anchor nodes. These anchor nodes have known positions and are used to approximate the unknown nodes' positions within wireless sensor networks. They aid in determining the location of randomly deployed nodes throughout the network using a localization technique, offering an alternative approach to achieve localization in IoT applications without the need for alternative devices to be associated with GPS receivers or auto-positioning capabilities. This cost-effective approach enables accurate device localization, thereby enhancing the comprehensive assessment and comprehension of real-time events. However, as depicted in Figure 2, obtaining precise sensor node locations remains a significant challenge.

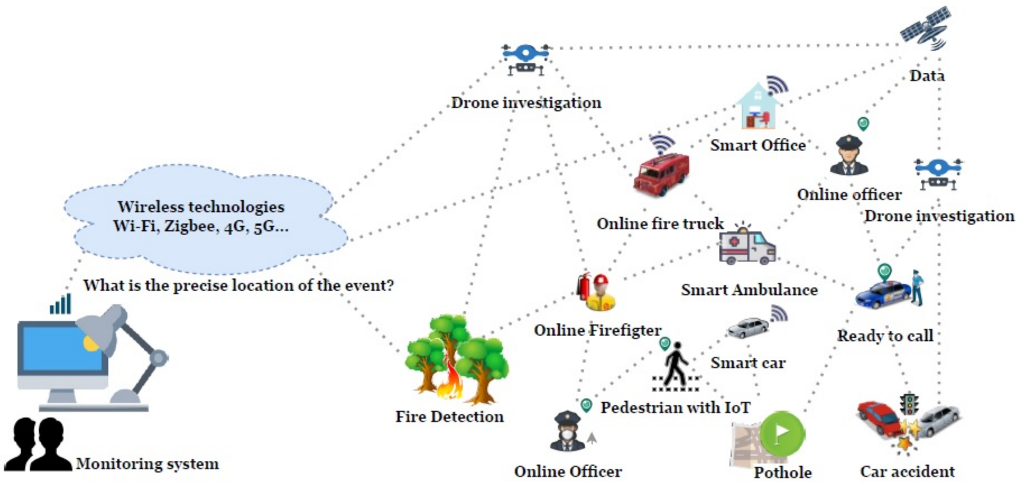


Figure 1. Relevance of localization in various domains.

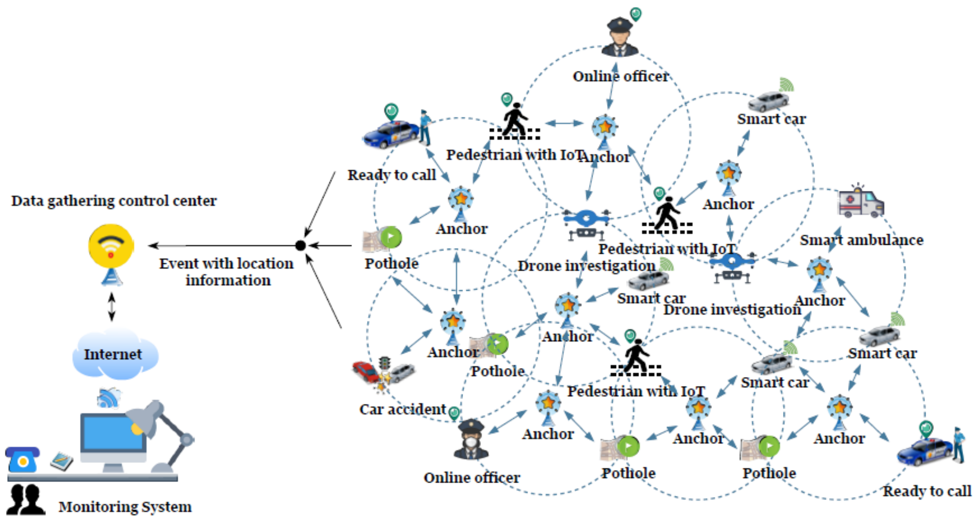


Figure 2. Example showing the process of localization.

Over the last few years, several localization approaches have been presented; these can be grouped into two main technologies, range-based and range-free [17]. Range-based categories include technologies such as Time of Arrival (ToA), Received Signal Strength Intensity (RSSI), Time Difference of Arrival (TDoA), and Angle of Arrival (AoA) [18–21]. These approaches apply either triangulation or trilateration techniques to determine the sensors' location by applying angles or distance information between synchronized sensor nodes.

Range-free approaches operate within critical scenarios; they need extra hardware, resulting in increased overhead, particularly in large-scale WSNs, as discussed in [22]. Alternatively, range-free techniques drive the sensor node positions from anchor node positions without requiring extra hardware. Range-free methods consider only information about the interconnection between anchors and unknown nodes, making them a more feasible alternative. These approaches use the exact location of nearby anchors to enable unknown nodes to calculate their locations within WSNs. Examples of range-free approaches include the DV-Hop algorithm, APIT technique, Centroid algorithm, and Amorphous [23–26]. APIT [24] calculates the locations of sensor nodes using the positions of a minimum of three anchor nodes. The algorithm obtains improved performance with a high percentage of anchors. The Centroid method [25] considers the barycenter of nearby anchors as the determined positions of nodes, avoiding the need for additional materials. Amorphous [26] is similar to DV-Hop in that it accurately determines the locations of unknown nodes. DV-Hop [23] is a well-known approach that has become the most recommended candidate for the localization of sensors in different kinds of topologies using a limited number of anchor nodes within a wireless sensor network [27]. In this approach, the sensor node position estimate is based on the average distance and the hops number, both of which are derived from each anchor node in the network. In this approach, the positions of nodes are calculated using a multilateration technique, which is applicable even when there are fewer than three neighboring anchor nodes.

In the existing literature, a number of localization algorithms that use the DV-Hop technique have been suggested to reduce the errors in estimated sensor node positions. Despite these efforts, localization accuracy remains inadequate, prompting a need for more enhancements. To address this gap, in this study we introduce three novel localization algorithms that combine DV-Hop with Chicken Swarm Optimization (CSO) [28]. These algorithms are developed to enhance and refine the localization accuracy of sensor nodes within WSNs. The key contributions of this study can be stated as follows:

- Three new algorithms to increase localization accuracy, denoted as DV-HopCSO, W-DV-HopCSO, and HW-DV-HopCSO.
- New and enhanced steps to increase localization accuracy.
- Evaluation of the presented algorithms considering four distinct types of complex topology.
- Verification of the reliability of the proposed localization approaches regarding their accuracy and error considering four distinct types of complex topology through comparisons with existing algorithms (DV-Hop, PSODV-Hop, MDV-Hop, W-DV-Hop, and HW-DV-Hop) involving the communication range, number of anchor nodes, total number of nodes, and maximum number of iterations.

The rest of this paper is structured as follows: Section 2 covers the background on localization algorithms from the literature; Section 3 presents the proposed algorithms based on DV-Hop and chicken swarm optimization; Section 4 presents the simulation results and performance analysis' and Section 5 concludes the paper and suggests future work.

2. Related Works and Background

In this work, our focus lies on approaches in the range-free category, and in particularly on the advancement of localization techniques based on the DV-Hop technique. One notable issue is that computing the average hop size value during the DV-Hop localization process can result in inaccurate estimates of the distance between nodes. Improving the accuracy when calculating the distance between sensor nodes could lead to a lower localization error. Various DV-Hop enhancements to tackle these concerns have been claimed in the state-of-the-art. Many researchers have recently made major attempts to encourage the use of weighted schemes and bio-inspired techniques to optimize and improve localization accuracy in both anisotropic and isotropic network topologies.

In [29], the authors presented a new scheme based on DV-Hop that uses weighted redundancy and optimized beacons. Their paper presented a weighted iterative technique for determining the best number of iterations and average hop size whlie integrating the weighted mean square error criterion. The hop error between anchors was utilized to assign weights to the signals, while the distance computation employed the optimal average hop size achieved in the previous phase. As a result, the positions of unknown nodes were derived using only a selection of beacon nodes with lower localization errors. Simulations revealed that the proposed variants outperformed previous localization methods, including the original DV Hop.

In [30], the authors proposed a modified hop count technique using mobile anchor nodes to reduce the localization error. The proposed technique was assessed based on communication range and the number of anchor nodes. The authors studied static wireless sensor networks with different numbers of mobile anchor nodes, and used their technique to design a similar solution with a static anchor node. Regarding the localization error, their results demonstrated that the suggested scheme surpasses the original DV-Hop.

In [31], the authors proposed an alternative modification to the DV-Hop technique to increase localization accuracy in anisotropic networks. The DV-Hop approach was improved in three steps, while the PSO and simulated annealing (SA) methods were combined to handle nonlinear equations and increase the localization accuracy of the sensor nodes. The results demonstrated the efficiency of their proposed DV-Hop compared to competitive methods in the literature. However, in this approach, communication overload among the sensor nodes in the network increases during multi-hop information forwarding.

In [32], the authors discussed an enhanced localization algorithm called DEIDV-Hop. This approach combines enhanced DV-Hop and DE techniques. The major aim of DEIDV-Hop is to correct any inaccuracies caused by the distance of the average hop size. To increase population diversity, the authors included a random mutation mechanism that impacts random individuals. This phase seeks to prevent the DE algorithm from experiencing premature convergence and search stagnation. Moreover, during the crossover process, they incorporated a segment of social learning from the PSO based on the newly created individual. Simulation results revealed that this DEIDV-Hop approach had a lower localization error and better stability compared to the current techniques in four distinct network settings.

In [33], the authors introduced an improved algorithm called LSDV-Hop. This novel approach us es the least squares theory to improve the precision of node localization. LSDV-Hop computes a transformation vector using the least-squares method in order to align the true and estimated positions of the anchor nodes. The simulation results showed that this LSDV-Hop approach surpasses the original DV-Hop.

In [34], the authors presented an online sequential strategy for DV-Hop-based localization. This approach is divided into three main stages. In the first phase, a unique technique is introduced for computing the average hop size between anchors. The typical DV-Hop technique is then modified, allowing it to be used as an online sequential localization technique in the next phase. To determine the locations of target nodes, the final phase applies a sequential technique in conjunction with a specified anchor set. In simulations, this scheme outperformed the original DV-Hop in terms of localization accuracy. Another enhanced scheme based on DV-Hop was presented in [35]. The proposed technique applies a new method to report the hop size distance, and uses Hyperbolic 2D instead of multilateration to estimate node locations. Simulations demonstrated that this scheme outperforms the original DV-Hop.

In [36], the authors proposed various enhancements based on DV-Hop, which they denoted iDV Hop1, iDV-Hop2, and Quad DV-Hop. Geometrical approaches are used in all three variants. The performance of these algorithms was evaluated using C-shaped and uniform network topologies. In terms of localization accuracy, their findings revealed that the iDV-Hop1, iDV-Hop2, and Quad DV-Hop algorithms all outperform DV-Hop.

In [37], a new DV-Hop-based scheme was presented that incorporates the PSO metaheuristic technique to reduce the localization error. The authors described two ways to reduce localization errors in a range-free localization approach, and addressed node localization faults by applying a well-defined set of equations. Their improved node localization technique resulted in a lower localization. Simulations revealed that this approach outperforms the original DV-Hop.

In [38], the authors presented a new optimized scheme that uses CSO to enhance localization precision when estimating node locations. They conducted an experiment in which they varied the network features and the number of hens in the algorithm to evaluate the performance of their new approach. They assessed its ability to improve on conventional CSO-based approaches and carried out a comparison between their scheme and PSO, finding that their technique can outperform PSO. This confirms the significance and potential of CSO-based approaches for enhancing network node localization. Table 1 below provides a comprehensive summary of DV-Hop localization algorithm variants. In the next section, we provide an extensive overview of the localization process in DV-Hop-based approaches and briefly discuss our previous weight-based DV-HOP localization algorithms.

2.1. Weight-Based Enhancements to DV-Hop

In this section, we provide a comprehensive explanation of the localization process in the DV-Hop approach. Additionally, we introduce two localization algorithms that we previously proposed in [39–41], denoted W-DV-Hop and HW-DV-Hop. These localization algorithms are designed with the intent of significantly improving the accuracy when estimating the sensor positions.

2.1.1. Basic DV-HOP Algorithm

In this section, we detail the process of localization using DV-Hop in WSNs. DV-Hop has demonstrated its ability to precisely determine the locations of sensor devices in both IoT applications and WSNs. The algorithm estimates the locations of these devices based on the relative distances between nearby anchor nodes. This solution enables precise positioning, making it the preferred choice for localization tasks in the aforementioned contexts. The basic DV-Hop algorithm consists of three main stages: (i) disseminating information; (ii) determining the average hop size distance; and (iii) estimating the position. These stages collectively contribute to estimation of the location of devices within a network.

In the first stage of the DV-Hop localization process, the anchor node (denoted as A_i) broadcasts a hello packet to initialize the network. As the hello packet is relayed through the network, the hop count value, which is initially 0, increases; each hop count reflects the number of traversed nodes. When a node, either an anchor or a target/unknown node (unknown nodes are denoted as U), receives the hello packet for the first time, it saves the location of the sending anchor node A_i and initializes $H_{i;u}$ as the hop count value collected in the packet. This hop count value is computed as the minimum number of hops required for node U to reach anchor node A_i. When node U receives a hello packet, it alters its $H_{i;u}$. If the received packet has a lower hop count than the existing $H_{i;u}$ value, then node U updates its $H_{i;u}$ with the new lower hop count and keeps the packet for further processing; otherwise, node U ignores this new higher $H_{i;u}$ value.

Throughout this stage, all sensor nodes diligently gather and maintain the hop count and send information to every anchor A_i. This meticulous recordkeeping ensures that accurate hop count data are available for other localization steps, facilitating the precise estimation of node locations within the network.

Table 1. Comprehensive summary of DV-Hop localization algorithm variants.

Authors	Summary	Strength of Approach	Anchors Based	Node Density	Localization Error	Approaches of Localization	Complexity	Accuracy
Chen et al. in [29]	A novel method that integrates redundancy, optimized beacons, and iterative techniques using hop error-based signal weighting and an optimal average hop size to improve localization.	Weighted redundancy and optimized beacons are used.	Yes	High	Medium	Iterative	Medium	Medium
Yanfei et al. in [30]	An approach that involves modifying the hop count technique and incorporating a mobile anchor node. The hop count computation is modified.	Both static and mobile anchor nodes are used.	Yes	High	Medium	Distributed	Medium	Medium
Shi et al. in [31]	A modified approach that combines Particle Swarm Optimization (PSO) and Simulated Annealing (SA) to handle nonlinear equations.	Uses the PSO and SA methods to address nonlinear equations. The anisotropic networks.	Yes	High	Lower	Iterative	High	High
Han et al. in [32]	DEIDV-Hop combines an enhanced DV-Hop algorithm with Differential Evolution (DE) to correct inaccuracies caused by the average hop size in the network.	Uses PSO with four distinct network topologies.	Yes	High	Lower	Iterative	High	High
Zhang et al. in [33]	A localization approach called LSDV-Hop that leverages the least squares theory to enhance the precision of node localization.	Transformation vector using the least-squares method.	Yes	High	Medium	Distributed	Medium	Medium
Messous et al. in [34]	An online sequential localization algorithm based on the DV-Hop technique.	Modifies DV-Hop into an online sequential localization algorithm	Yes	High	Medium	Distributed	High	Medium
Song et al. [35]	An approach based on a novel method for hop size distance calculation with 2D hyperbolic techniques.	Uses a new average hop size and 2D hyperbolic techniques	Yes	High	Medium	Iterative	Medium	Medium

Table 1. *Cont.*

Authors	Summary	Strength of Approach	Anchors Based	Node Density	Localization Error	Approaches of Localization	Complexity	Accuracy
Tomic et al. in [36]	Three enhanced algorithms (iDV-Hop1, iDV-Hop2, and Quad DV-Hop) based on the DV-Hop technique.	New approaches based on geometry.	Yes	High	Medium	Distributed	High	Medium
Singh et al. in [37]	A new DV-Hop-based scheme that incorporates PSO to minimize the localization error.	Combines DV-Hop and PSO	Yes	High	High	Iterative	High	High
Rabhi et al. in [38]	An optimized algorithm based on the chicken swarm optimization approach.	Uses chicken swarm optimization to achieve superior results in comparison with the PSO method.	Yes	High	High	Iterative	High	High

During this localization process stage, sensor nodes convey the hop value to the anchors. In the second step, anchor A_i determines the average hop distance $AHSize_i$. After the computation of $AHSize_i$, anchor A_i propagates this value across the network. The formula employed for determining the average hop distance $AHSize_i$ for each anchor A_i is expressed as follows:

$$AHSize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} H_{i,j}} \tag{1}$$

Here, $H_{i;j}$ is the hop count from A_i to A_j, where (x_i, y_i) and (x_j, y_j) represent the coordinates of anchors A_i and A_i, respectively. Additionally, $H_{i;u}$ is the hop count value from node A_i to the specific target node U.

Based on this formula, A_i, which is an anchor, saves its unique average hop size $AHSize_i$, which plays a crucial role in estimating the unknown nodes during the localization steps. When this has been computed, anchor node A_i broadcasts this information across the network, ensuring that all network nodes have access to it.

In the third stage, when a node U obtains $AHSize_i$, it multiplies it by $H_{i;u}$ (the number of hops to A_i) in order to obtain the estimated distance to every anchor node A_i, denoted as d_i, which can be estimated as follows:

$$d_i = AHSize_i \times H_i, \tag{2}$$

where the value of i varies between 1 and k anchor nodes and H_i is the number of hops to anchor node A_i.

Therefore, the estimated position of node U, denoted as (x, y), can be found using the following equation.

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ \quad \dots \\ (x - x_k)^2 + (y - y_k)^2 = d_k^2 \end{cases} \tag{3}$$

By applying the least squares approach to solve the aforementioned equation, an unknown node U can determine its estimated location U_{DV-Hop} as follows:

$$U_{DV-Hop} : \begin{bmatrix} x \\ y \end{bmatrix} = (A^T A)^{-1} A^T B, \tag{4}$$

where

$$A = -2 \times \begin{bmatrix} x_1 - x_k & y_1 - y_k \\ x_2 - x_k & y_2 - y_k \\ \dots & \dots \\ x_{k-1} - x_k & y_{k-1} - y_k \end{bmatrix} \tag{5}$$

$$B = \begin{bmatrix} d_1^2 - d_k^2 - x_1^2 + x_k^2 - y_1^2 + y_k^2 \\ d_2^2 - d_k^2 - x_2^2 + x_k^2 - y_2^2 + y_k^2 \\ \dots \\ d_{k-1}^2 - d_k^2 - x_{k-1}^2 + x_k^2 - y_{k-1}^2 + y_k^2 \end{bmatrix}. \tag{6}$$

It is important to note that the anchor nodes must not lie on a single line. If they do, the matrix $(A^T A)^{-1}$ becomes nonexistent due to the singularity of $A^T A$.

2.1.2. W-DV-Hop Localization Algorithm

In our prior research papers [39,40], we introduced a new scheme for the typical DV-Hop algorithm, which we denoted as W-DV-Hop. Similar to DV-Hop, W-DV-Hop uses three main steps: (i) dissemination of information within the network; (ii) a new technique for computing the average hop distance; and (iii) estimating the locations of the unknown nodes using trilateration.

More specifically, the first stage of the improved W-DV-Hop algorithm closely resembles the standard DV-Hop. A hello message containing the location of anchor node A_i is broadcast in the network, with the hop count from anchor to anchor initialized as 0. In the second stage, the average hop size between anchors A_i is determined through the use of a new weighted formula based on the mean [42]. In addition, in the improved W-DV-Hop algorithm the average hop size is determined using the mean square error approach [43]. The formula applied to compute $AHSize_i$ is as follows:

$$\xi_1 = \frac{1}{k-1}\sum_{j\neq i}(d_{i,j} - AHSize_i \times H_{i,j}^2), \tag{7}$$

where k represents the ratio of anchors, $AHSize_i$ is estimated under the assumption that $(\partial\xi_1 / AHSize_i = 0)$, $d_{i;j}$ is the distance between anchor nodes A_i and A_j based on their exact positions, which can be obtained through the GPS module, and $H_{i;j}$ is the minimum hop count between anchors A_i and A_j.

$$AHSize_i = \frac{\sum_{j\neq i} H_{i,j} \times d_{i,j}}{\sum_{j\neq i} H_{i,j}^2} \tag{8}$$

To compute the updated $AHSize_{new}$, the following formula is applied:

$$AHSize_{new} = \frac{\sum_{i=1}^{k} W_i \times AHSize_i}{\sum_{i=1}^{m} W_i}, \tag{9}$$

where

$$W_i = \frac{1}{\sum_{j\neq i} \left|AHSize_i - AHSize_j\right|}. \tag{10}$$

Here, W_i represents the weight assigned to every anchor A_i and k represents the percentage of anchors.

In the third phase, the unknown nodes apply the multilateration technique to estimate their locations. Upon receiving $AHSize_{new}$, a normal node U multiplies its hop count $Hi;u$ (hops to A_i) by $H_{i;j}$. This yields the estimated distance to A_i, denoted as d_i ($d_i = AHSize_{new} \times H_i$), where $i \in \{1, 2, \ldots, k\}$ represents the anchor nodes. The following equation is then created using the estimated position (x, y) of node U.

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 = d_1^2 \\ (x-x_2)^2 + (y-y_2)^2 = d_2^2 \\ \quad\quad\quad\quad \cdots \\ (x-x_k)^2 + (y-y_k)^2 = d_k^2 \end{cases} \tag{11}$$

Using the least square technique, the solution to the aforementioned equation enables an unknown node U to acquire its estimated position $U_{WDV-Hop}$ through the following formula:

$$U_{WDV-Hop} : \begin{bmatrix} x \\ y \end{bmatrix} = (A^T A)^{-1} A^T B, \tag{12}$$

where

$$A = -2 \times \begin{bmatrix} x_1 - x_k & y_1 - y_k \\ x_2 - x_k & y_2 - y_k \\ \dots & \dots \\ x_{k-1} - x_k & y_{k-1} - y_k \end{bmatrix} \tag{13}$$

$$B = \begin{bmatrix} d_1^2 - d_k^2 - x_1^2 + x_k^2 - y_1^2 + y_k^2 \\ d_2^2 - d_k^2 - x_2^2 + x_k^2 - y_2^2 + y_k^2 \\ \dots \\ d_{k-1}^2 - d_k^2 - x_{k-1}^2 + x_k^2 - y_{k-1}^2 + y_k^2 \end{bmatrix}. \tag{14}$$

It is important to note that the anchor nodes must not lie on a single line. If they do, the matrix $(A^T A)^{-1}$ becomes nonexistent due to the singularity of $A^T A$.

$$A = ((G^T G)^{-1}) G^T b \tag{15}$$

The location of node U is then calculated as follows.

$$\begin{cases} x_u = A(1) \\ y_u = A(2) \end{cases} \tag{16}$$

2.1.3. HW-DV-Hop Localization Algorithm

The HW-DV-Hop approach (Hyperbolic Weighted DV-Hop) is another improved method based on DV-Hop. HW-DV-Hop uses the following steps: (i) the anchors disseminate their precise locations; (ii) a new technique improves the average hop size; and (iii) the hyperbolic 2D schema is used to estimate the position of unknown nodes instead of the usual method.

The first stage of HW-DV-Hop is similar to the DV-Hop and W-DV-Hop approaches. Each anchor A_i sends a message that includes its position and number of hops between the anchor nodes. In the second step, $AHSize_i$ is determined using a new formula:

$$AHSize_{new} = \frac{\sum_{i=1}^{m} W_i \times AHSize_i}{\sum_{i=1}^{m} W_i}, \tag{17}$$

where

$$W_i = \frac{1}{\sum_{j \neq i} \left| AHSize_i - AHSize_j \right|}. \tag{18}$$

In the third phase, rather than applying the multilateration approach, the 2D hyperbolic location schema [44] is used to estimate the locations of the unknown nodes.

This approach assumes that (x_i, y_i) represents the coordinates of anchor node A_i and (x_u, y_u) represents the coordinates of unknown node U. The following formula is used to calculate $d_{i,u}$, which is the approximate distance between A_i and U.

$$d_{i,u}^2 = (x_i - x_u)^2 + (y_i - y_u)^2 \tag{19}$$

If $R_i = x_i^2 + y_i^2$ and $S_i = x_u^2 + y_u^2$, Equation (18) can be reformulated as follows.

$$d_{i,u}^2 - R^2 = -2x_i x_k - 2y_i y_k + S_i \tag{20}$$

Equation (19) can be represented in matrix form as

$$GA = b, \tag{21}$$

where $A = [x_u, y_u, S_u]^T$, G = $\begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_k & -2y_k & 1 \end{bmatrix}$, b = $\begin{bmatrix} d_{1,u}^2 - R_1 \\ d_{2,u}^2 - R_2 \\ \vdots \\ d_{k,u}^2 - R_k \end{bmatrix}$.

According to Equation (20), A can be solved using the following formula.

$$A = ((G^T G)^{-1}) G^T b \tag{22}$$

Thus, the coordinates of node U are calculated using the formula below.

$$\begin{cases} x_u = A(1) \\ y_u = A(2) \end{cases} \tag{23}$$

3. Proposed Localization Algorithms

In this section, we introduce a set of three novel localization approaches aimed at overcoming the limitations of DV-Hop and enhancing its accuracy by combining it with intelligent swarm optimization. While DV-Hop is widely recommended in WSNs, it often yields inaccurate positions for the sensor nodes, particularly in fields such as military and environmental monitoring. To enhance localization accuracy, we integrate CSO, which is a biologically inspired intelligent approach widely recognized for its efficiency in addressing complex problem scenarios.

3.1. Motivation Behind Using Intelligent Swarm Optimization

DV-Hop is known to suffer from localization errors caused by uncertainty in distance estimation, particularly as relates to the minimum hop count and average hop size. This leads to significant deviations between estimated and actual distances that greatly affect its localization precision. Localization in WSNs is a challenging NP-hard optimization problem. Nature-inspired approaches have emerged as suitable solutions for this problem, with swarm intelligence optimizations being particularly promising thanks to their adaptability, simplicity, self-organizing capacity, decentralization, and robustness. These algorithms draw inspiration from social behaviors observed in animal communities such as ants, fish, birds, and even chickens. Swarm intelligence optimization encompasses various approaches, including the PSO, Ant Colony Optimization, Chicken Swarm Optimization, and Artificial Bee Colony algorithms [28,45–47]. Recent research comparisons indicate that CSO and PSO are among the most effective optimization algorithms for localization problems in WSNs, offering numerous advantages over other techniques. For instance, PSO is advantageous in its utilization of minimal parameters to adjust the particle population, resulting in simplified parameter tuning. Moreover, it exhibits low spatial complexity thanks to its efficient utilization of small temporary storage to minimize memory requirements. The fast convergence speed of PSO is attributed to its selective sharing of solutions, where only the most optimistic particle can influence others. This enhances the algorithm's ability to rapidly converge towards optimal solutions. Furthermore, as demonstrated by its successful use in tackling an extensive variety of problem areas, PSO has excellent adaptability. Lastly, PSO offers an intuitive approach to interpreting and adapting solutions to specific problem contexts, facilitating ease of implementation and customization.

The aim of this paper is to improve the localization accuracy of algorithms based on chicken swarm optimization while achieving faster convergence. Our goal is to refine the positions of nodes within a specific area. To achieve this, we introduce extensions to the existing DV-Hop technique, which we call W-DV-Hop and HW-DV-Hop. These modifications incorporate biologically inspired optimization techniques by integrating CSO into the localization process. The process of localization consists of four key stages: (i) broadcasting and minimal hop computation; (ii) determining the average hop distance; (iii) localization for estimating the positions of unknown nodes; and (iv) CSO integration for determining the optimal positions of unknown nodes. Through this work, we endeavor to demonstrate the efficacy of leveraging CSO to improve the localization process and achieve improved accuracy in locating unknown sensor positions within WSNs.

3.2. CSO Optimization Approach

Chicken swarm optimization is a biologically inspired heuristic optimization approach that draws inspiration from the collective behavior of chickens. It falls under the category of swarm intelligence algorithms, which are widely used to tackle complex optimization problems by continuously seeking the best solution based on specific quality measures.

The CSO approach is based on the principles of self-organization, simplicity, and decentralization. Each member of the swarm represents a potential solution, and adapts its position and behavior to explore the search space. The interactions among swarm members facilitate information sharing, leading to updates in their positions and improvements in their individual and collective performance. This collective intelligence enables CSO to converge toward optimal solutions effectively. Due to its demonstrated effectiveness and competitiveness in various problem domains, CSO has become a favored optimization algorithm. By harnessing the collective behavior of chickens, CSO offers an efficient and adaptable approach to solving complex optimization problems. CSO uses a minimal number of parameters to adjust the particle population to simplify the optimization process. Furthermore, it guarantees efficient memory utilization, as it features low spatial complexity while using small temporary storage. CSO guarantees fast convergence by allowing the most optimistic chicken particles to share their solutions with others, offering an interactive and adaptable approach to interpreting and customizing solutions to specific problem contexts. The diversity of applications is ensured as well, as CSO has proven itself in many different areas.

The optimization process of CSO consists of four steps: (i) population initialization, parameter definition, and determining the fitness value of each individual calculation; (ii) sorting individuals by fitness value and determining their identities and subgroups; (iii) iteration of individuals with distinct identities based on different formulas and resorting them at regular intervals based on their new fitness values; (iv) iteration stops when the conditions are met, and the optimal solution is selected from among the population.

In the CSO approach, roosters with better fitness values are better able to find food in wider areas. Roosters with the best fitness values can cover a larger distance to find food than those with lower fitness values. The rooster movement function is as follows:

$$X_{(p,q)}^{(t+1)} = X_{(p,q)}^{t} \times (1 + \mathrm{Randn}(0, \sigma^2)) \tag{24}$$

$$\sigma^2 = \begin{cases} 1 & \text{if } f_p \leq f_k \\ \exp\left(\frac{f_k - f_p}{|f_p| + \epsilon}\right) & \text{otherwise} \end{cases} \quad \text{for } k \in [1, N],\ k \neq p \tag{25}$$

where the normal distribution of $Randn(0, \sigma^2)$ is around a mean of 0 and standard deviation σ^2, The index k refers to the stochastically determined rooster, f_k is the fitness value of the k-th rooster, and the minimum constant ε is applied to avoid the zero-division error.

The hens follow the roosters as group mates to find food, and can engage in stochastic pilfering of food discovered by the roosters. This can be illustrated using the following formulas:

$$X_{(p,q)}^{(t+1)} = X_{(p,q)}^{t} + \varphi_1 \times \mathrm{Rand} \times (X_{(r1,q)}^{t} - X_{(p,q)}^{t}) + \varphi_2 \times \mathrm{Rand} \times (X_{(r2,q)}^{t} - X_{(p,q)}^{t}), \tag{26}$$

$$\varphi_1 = \exp\left(\frac{f_p - f_{r1}}{|f_p| + \epsilon}\right), \tag{27}$$

$$\varphi_2 = \exp(f_{r2} - f_p), \tag{28}$$

where *Rand* represents a uniform random number in the range [0, 1], f represents the fitness value, r_1 represents an index of a roosters and it serves as the partner for the p-th hen, r_2 denotes the index of the selected rooster or hen from the group of individuals, where r_1 and r_2 are different, and φ_1 and φ_2 represent the selection coefficients. In their search for food, the chicks trail behind the mother hen; their behavior is expressed as follows:

$$X_{(p,q)}^{(t+1)} = X_{(p,q)}^{t} + \mathrm{FL} \times (X_{(s,q)}^{t} - X_{(p,q)}^{t}), \tag{29}$$

where $X_{(s,q)}^{t}$ signifies the position of the mother hen of the s-th chick's mother and FL is a constant randomly selected from the range of [0, 2].

Generally, in the realm of 2D WSNs the network localization challenge can be modeled by the following concept. Consider a network denoted as $Net = \{\mathrm{Sn}_1, \mathrm{Sn}_2, \ldots, \mathrm{Sn}_{k+n}\}$ in which there are n nodes and k anchors. The location of each sensor is represented by $A_i(x_i, y_i)$ and $U_i(xu_i, yu_i)$ for $i = 1$ to $k + n$. In this problem, it is assumed that the k anchor nodes A_i know their positions and that the objective is to estimate the n unknown nodes' positions of U_i. More precisely, the d_i can be estimated as shown in the second phase of each proposed DV-Hop variant. However, the estimation of this distance value always involves a higher degree of error.

In this paper, we present novel localization approaches using chicken swarm optimization to address the localization challenges in WSNs. In this study, the objective function of chicken swarm optimization is defined as follows.

$$f(x,y) = \min\left(\sum_{i=1}^{k}\left|\sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i\right|\right) \tag{30}$$

Using the objective function defined in Equation (26), the fitness function formula is as follows.

$$fitness = f(x, y) \tag{31}$$

Likewise, all chickens (roosters, hens, and chicks) update their positions using Equations (24), (26), and (29), respectively. The fitness of each particle is assessed using Equation (30). The optimal solution is considered to be the optimized position of the sensor node during this phase. Figure 3 provides the comprehensive localization process utilized in our proposed approach. In the following subsection, the CSO approach is applied to bridge the accuracy gap between DV-Hop and our improved algorithms.

3.3. CSO-Based Enhanced DV-Hop Algorithm

In this section, we discuss three localization approaches that are used in WSNs to enhance the accuracy and efficiency of sensor node position estimation. In Section 2, we introduced the W-DV-Hop and HW-DV-Hop algorithms. The DV-Hop, W-DV-Hop, and HW-DV-Hop approaches are further improved using an optimization technique based on the Chicken Swarm Optimization (CSO) approach. These improvements aim to accurately determine node locations in WSNs.

We assess different scenarios to confirm the efficacy of several localization algorithms in comparison with DV-Hop: MDV-Hop [35], PSODV-Hop [48], W-DV-Hop [39], and HW-DV-Hop [40]. All of the proposed techniques were developed and implemented in MATLAB, and we analyze their performance in static WSNs. The three new enhanced localization schemes are denoted as DV-HopCSO, W-DV-HopCSO, and HW-DV-HopCSO. These approaches consist of four stages, introduced as follows: (i) the locations of anchors are broadcast; (ii) the computation of the average hop size is enhanced; (iii) the hyperbolic 2D technique is adopted to optimize the unknown nodes' positions; and (iv) the CSO approach is integrated to enhance the accuracy of the localization process.

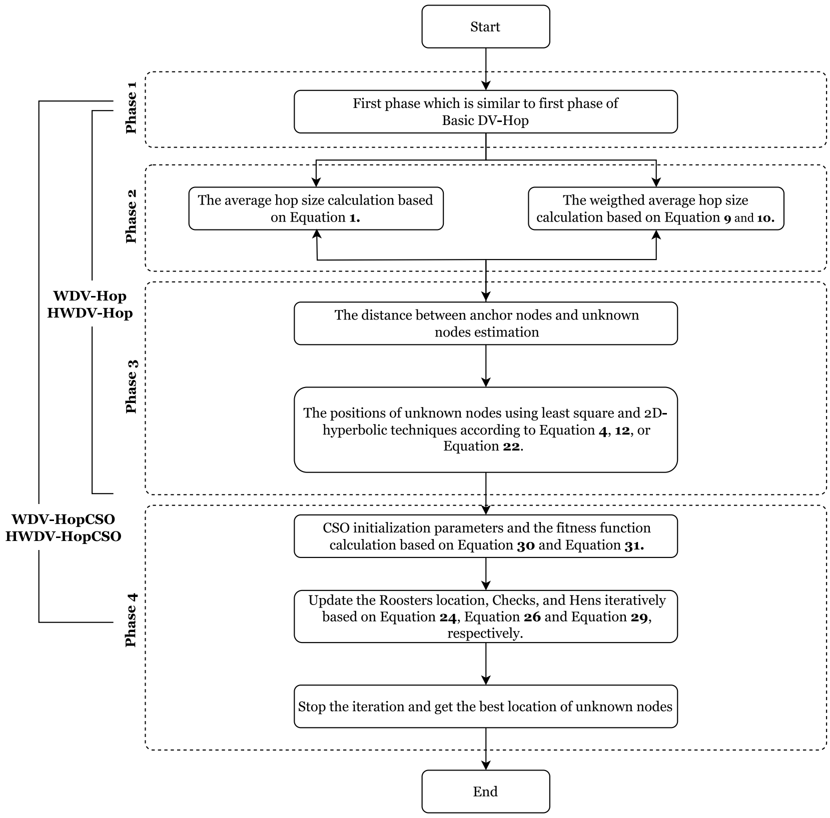


Figure 3. Flowchart of the proposed algorithms.

In the DV-Hop algorithm, the conventional three phases are kept and an additional fourth phase is introduced to integrate CSO. This phase results in improved accuracy when locating the positions of sensor nodes within the network.

For W-DV-HopCSO, the first and third stages are equivalent to those of W-DV-Hop, which is discussed in Section 2. Throughout the first phase, the anchor nodes transmit their position information. The weighted mean approach is used in the second phase to find the average hop size for each anchor node. Multilateration is used in the third phase to approximate the locations of unknown nodes. Finally, CSO is used in the fourth phase to minimize the positioning error of unknown nodes. Similarly, HW-DV-HopCSO is divided into four stages: (i) the locations of the anchors are broadcast; (ii) the average hop size computation is enhanced; (iii) the hyperbolic 2D solution is selected to determine the node positions; and (iv) CSO is used to optimize and refine the unknown node locations.

During the first step, the anchor nodes disseminate a message specifying their exact positions in the network. In the second step, we adopt the mean weighted method [42] to calculate the $AHSize_i$ utilizing the novel formula discussed in Section 3. In the third phase, the 2D hyperbolic location approach [43] is employed instead of multilateration to precisely identify the unknown nodes' coordinates. In the fourth stage, we use CSO to improve and refine the estimated positions. Furthermore, as detailed in this research, we use CSO to improve the precision of unknown node positions. The flowchart shown in Figure 3 provides an overview of the proposed hybrid localization methods, and the comprehensive pseudocode is illustrated in Algorithm 1.

Algorithm 1: Pseudocode of the Proposed Algorithms

Input: n nodes; k anchors; communication range R

Network topology distribution: square random; H-shaped; O-shaped; W-shaped
for i = 1 to n do
    for j = 1 to n do
        Distance calculation
        $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
        and initialize hop-count $H_{i,j} = 0$;
if $d_{i,j} <= R$ then
    $H_{i,j} = 1$;
else if $i == j$ then
    $H_{i,j} = 1$;
else
    $H_{i,j} = inf$;
end if
end for
end for

Creation of the Matrix of hop count between anchors based on the path algorithm;
Creation of distance matrix between anchors based on the path algorithm;

for i = 1 to k do
    Average hop size distance calculation $AHSize_i$ per anchor node based on Equation (1);
end for

for i = 1 to k do
    W_i weighted values calculation per anchor nodes based on Equation (18);
end for

New corrected average hop size distance $AHSize_{new}$ according to Equation (17);
for i = 1 to k do
    $d_{i,k} = AHSize_{new} \times H_{i,k}$ from i-th anchor to k-th unknown node;
end for

Determine the position of the unknown node using a 2D-hyperbolic based on Equation (23);

CSO parameters initialization and using the coordinate of nodes calculated according to
    2D-hyperbolic;
Evaluate the chickens' fitness values;
Made iteratively an update of the positions according to Equations (24), (26), and (29);
Stop the iteration when the goals are reached;
Output: the best locations of unknown nodes.

4. Simulation Results and Discussions

Comprehensive simulations were carried out using a MATLAB-based simulator to assess the performance of all the introduced algorithms. MATLAB is a widely endorsed numeric computing platform favored by many researchers for algorithm analysis, data examination, and scenario modeling. Hence, in this study we employed MATLAB 2019a to assess the performance of our algorithms for localization within static WSNs. Moreover, the performance of our developed algorithms was contrasted with that of the original DV-Hop, a PSO variant called PSODV-Hop [48], and the enhanced DV-Hop algorithms MDV-Hop [33], W-DV-Hop [38], and HW-DV-Hop [39] through a series of simulations.

The localization error and localization accuracy of each algorithm were evaluated by varying the simulation parameters, including the number of sensor nodes, ratio of sensor nodes to anchor nodes, range of communication R, and maximum number of iterations considering a square random network topology and three distinct types of complex network topologies (H-shaped, O-shaped, and W-shaped).

The localization accuracy used to confirm the efficiency of the introduced approach was determined by calculating the average localization error using the formula in Equation (33):

$$Localization\ error = \sqrt{(x^i_{\text{exact}} - x^i_{\text{estimated}})^2 + (y^i_{\text{exact}} - y^i_{\text{estimated}})^2} \quad (32)$$

$$Localization\ accuracy = \frac{\sum_{i=1}^{N} \sqrt{(x^i_{\text{exact}} - x^i_{\text{estimated}})^2 + (y^i_{\text{exact}} - y^i_{\text{estimated}})^2}}{N \times R} \quad (33)$$

where $(x^i_{exact}, y^i_{exact})$ represents the exact coordinate of sensor i and $(x^i_{estimated}, y^i_{estimated})$ represents the estimated coordinate of sensor i.

The localization error signifies the disparity between the accurate and estimated geographical position of the unknown node. This error is quantified according to the formula presented in Equation (32).

The evaluation of the new algorithms involved the following parameters: (i) the network topology, characterized by the ratio of sensor nodes to anchor nodes and the communication range R; (ii) the distribution of sensor nodes in a selected 100 m $\times$ 100 m area; and (iii) the communication range R used by all sensors in the network. A total of 30 experiments were carried out, and the reported values represent the average result derived from each of these experiments. Table 2 describes the simulation parameters in detail.

Table 2. Detailed simulation parameters.

Parameter	Value
Network	
Network topology	Square random, H-shaped, O-shaped, and W-shaped
Total runs	30
Length of area	100 m $\times$ 100 m
Number of nodes	200, 250, 300, 350, 400, 450, and 500
Number of anchor nodes	15%, 20%, 25%, 30%, 35%, and 40%
Communication range R	15 m, 20 m, 25 m, 30 m, and 35 m
PSO	
Number of iterations	50
particle sizes	20
Random values σ_1 and σ_1	[0, 1]
Learning coefficient C_1, C_2	1.5 and 2
Particle's velocity V_{max}	10
CSO	
Number of iterations	50
Size of chicken swarm	20
Number of roosters	8
Number of chicks	1
Number of hens	15

The efficiency of the introduced algorithms regarding localization accuracy was examined considering parameters such as the numbers of sensor nodes and anchor nodes, range of communication, maximum number of iterations, and time of complexity in four distinct types of topologies: square random, H-shaped, O-shaped, and W-shaped. The nodes within the square were distributed randomly in a defined area of interest. This distribution is the most commonly utilized topology for localization techniques. The H-shaped, O-shaped, and W-shaped topologies of networks are irregular, and are based on empty regions on two sides for the H-shaped topology, an internal side for the O-shaped topology, and three sides for the W-shaped topology. Figure 4 depicts the four network topologies used in our simulations. In the following subsections, we thoroughly discuss the simulation performance for each of the mentioned topologies.

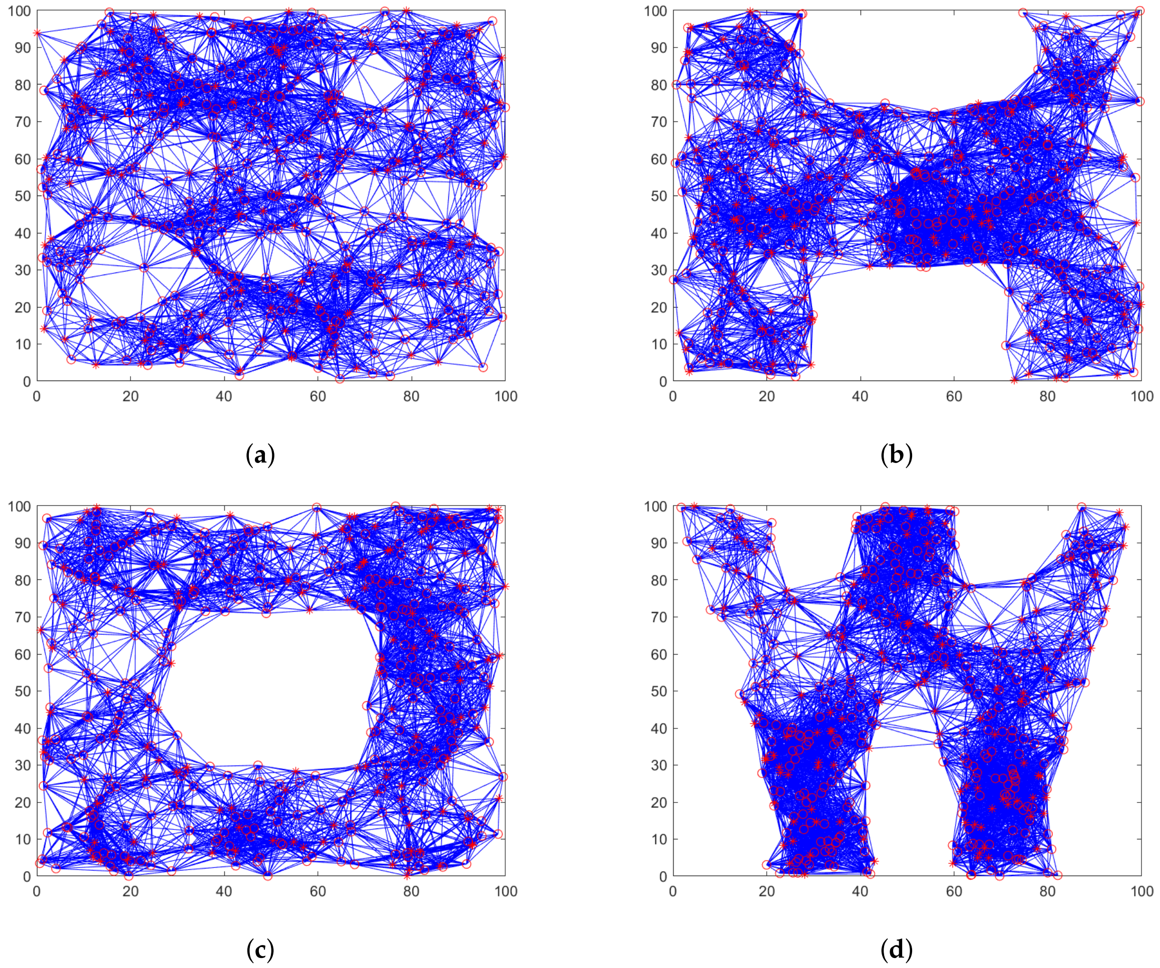
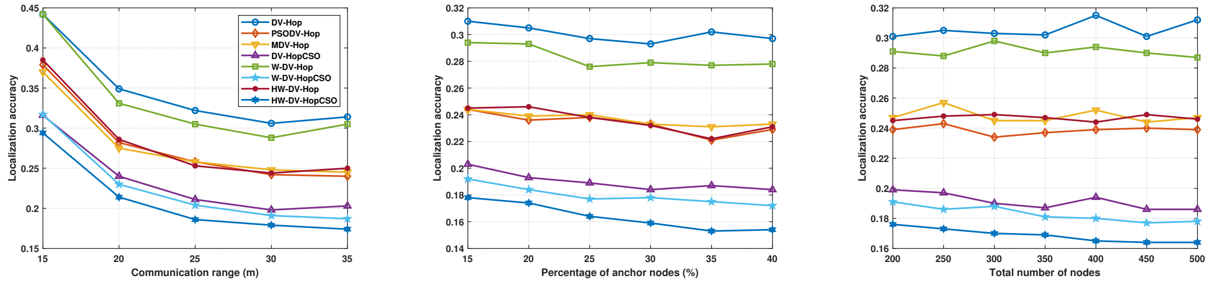


Figure 4. Node distribution vs. networks: (**a**) square random, (**b**) H-shaped, (**c**) O-shaped, (**d**) W-shaped (400 nodes, 30% of anchors, and R = 15 m).

4.1. Impact of Communication Range

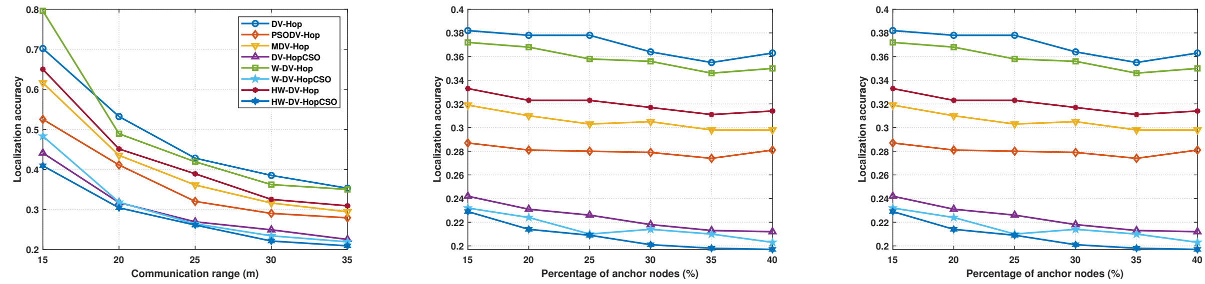
This section presents a comprehensive analysis of the introduced algorithms in order to assess their impacts and performance on four different kind of network topologies while varying the communication range: square random, H-shaped, O-shaped, and W-shaped. In this scenario, the number of nodes remained constant at 200, the percentage of anchors was set at 20%, and the range of communication R was varied from 15 m to 35 m.

The results, as depicted in Figures 5a–8a, reveal the influence of communication range on localization accuracy for each algorithm within the four network topologies. For each topology, the network becomes more connected as the communication range increases. The HW-DV-HopCSO exhibits respective localization accuracy improvements of 46%, 42%, 46%, and 33% compared to DV-Hop and W-DV-Hop and respective improvements of 36%, 33%, 26%, and 32% compared to HW-DV-Hop, MDV-Hop, and PSODV-Hop in random square, H-shaped, O-shaped, and W-shaped networks. In contrast, DV-HopCSO, W-DV-HopCSO, and HW-DV-HopCSO consistently outperform other localization algorithms across various network topologies. In addition, these algorithms outshine competitive localization algorithms, including PSODV-Hop, MDV-Hop, and HW-DV-Hop, across different network topologies. The HW-DV-HopCSO algorithm demonstrates the highest efficiency among the tested algorithms.



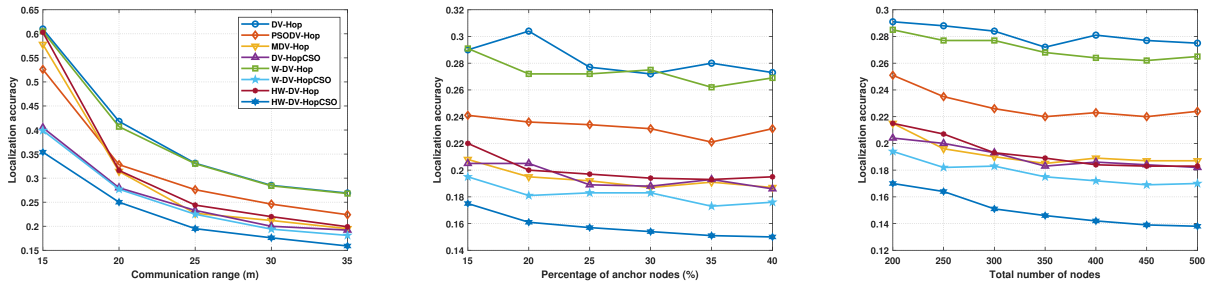
(**a**) (**b**) (**c**)

Figure 5. Localization accuracy in the square random network: (**a**) 15% of anchors and 200 nodes, (**b**) R = 30 m and 200 nodes, (**c**) 15% of anchors and R = 30 m.



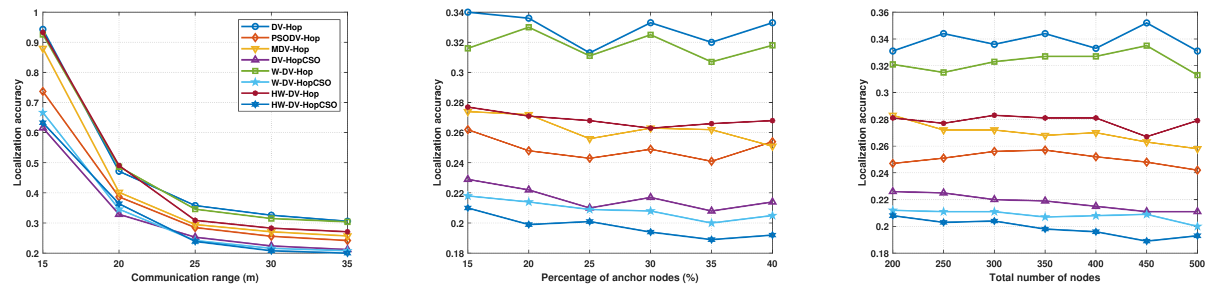
(**a**) (**b**) (**c**)

Figure 6. Localization accuracy in the H-shaped network: (**a**) 15% of anchors and 200 nodes, (**b**) R = 30 m and 200 nodes, (**c**) 15% of anchors and R = 30 m.



(**a**) (**b**) (**c**)

Figure 7. Localization accuracy in the O-shaped network: (**a**) 15% of anchors and 200 nodes, (**b**) R = 30 m and 200 nodes, (**c**) 15% of anchor nodes and R = 30 m.



(**a**) (**b**) (**c**)

Figure 8. Localization accuracy in the W-shaped network: (**a**) 15% of anchors and 200 nodes, (**b**) R = 30 m and 200 nodes, (**c**) 15% of anchors and R = 30 m.

4.2. Impact of the Percentage of Anchor Nodes

In this section, we present a thorough examination of the proposed approaches to evaluate their performance on the same four distinct networks while changing the percentage of anchor nodes within the network. In this configuration, the number of sensor nodes remained fixed at 200 and the communication range remained constant at 30 m while the percentage of anchor nodes was varied between 15% and 40%. The findings reveal that the accuracy of all the proposed localization methods is impacted by the percentage of anchor nodes.

Figures 5b–8b depict the results with respect to the percentage of anchor nodes. It can be seen that DV-Hop, W-DV-Hop, MDV-Hop, and HW-DV-Hop exhibit lower localization accuracy compared to HW-DV-HopCSO across the four different network topologies. Furthermore, HW-DV-HopCSO respectively enhances localization accuracy by 51%, 45%, 49%, and 43% compared to DV-Hop and W-DV-Hop and by about 34%, 33%, 25%, and 26% compared to HW-DV-Hop, MDV-Hop, and PSODV-Hop in the random square, H-shaped, O-shaped, and W-shaped networks. In addition, the proposed W-DV-HopCSO and HW-DV-HopCSO algorithms surpass the other approaches in terms of localization accuracy, offering significantly improved results compared to competitive algorithms in all of the different network topologies.

4.3. Impact of the Total Number of Nodes

In this section, our focus is on assessing the accuracy of localization for the proposed techniques within square random, H-shaped, O-shaped, and W-shaped network topologies. We comprehensively analyze their performance by examining the algorithm's ability to respond to changing the number of sensor nodes in the network. In this scenario, the number of sensor nodes ranged from 200 to 500, the communication range remained fixed at 30 m, and the percentage of anchor nodes was set at 20%. The results provide a comprehensive comparison of the proposed localization algorithms based on the number of sensor nodes. Figures 5c–8c depict the simulation results based on the number of nodes. It is essential to highlight that as the density of sensor nodes increases within the region of interest, the network achieves greater interconnectivity. In this scenarion, the HW-DV-HopCSO algorithm demonstrates significantly higher localization accuracy, approximately 48%, 47%, 50%, and 43% better than DV-Hop and W-DV-Hop and approximately 34%, 36%, 27%, and 26% better than HW-DV-Hop, PSODV-Hop, and MDV-Hop for random square, H-shaped, O-shaped, and W-shaped network topologies, respectively. Furthermore, in this experiment we observed that our DV-HopCSO, W-DV-HopCSO, and HW-DV-HopCSO algorithms yield better results regarding localization accuracy when contrasted with the competing methods. The HW-DV-HopCSO algorithm exhibits the most outstanding performance among the algorithms under consideration.

In all considered scenarios, there is a significant improvement in the performance of the HW-DV-HopCSO localization algorithm across the four different network topologies when modifying various simulation parameters. Our results indicate that integrating the CSO method to refine the locations of sensor nodes represents a valuable approach to localization in WSNs.

Table 3 provides a concise summary comparing the introduced and competing algorithms in terms of minimum, maximum, and mean localization accuracy.

Table 3. Simulation results and analysis for the random square, H-shaped, O-shaped, and W-shaped networks (200 nodes, 40 anchors, R = 30 m).

Network Topology	**Random Square**			**H-Shaped**			**O-Shaped**			**W-Shaped**		
Algorithm	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**	**AVG**
DV-Hop	0.032	0.693	0.282	0.021	1.065	0.379	0.053	0.806	0.282	0.017	0.772	0.274
DV-HopPSO	0.039	1.11	0.272	0.013	1.148	0.288	0.007	0.892	0.236	0.009	0.854	0.225
MDV-Hop	0.025	0.647	0.258	0.022	0.974	0.327	0.025	0.522	0.208	0.031	0.79	0.299

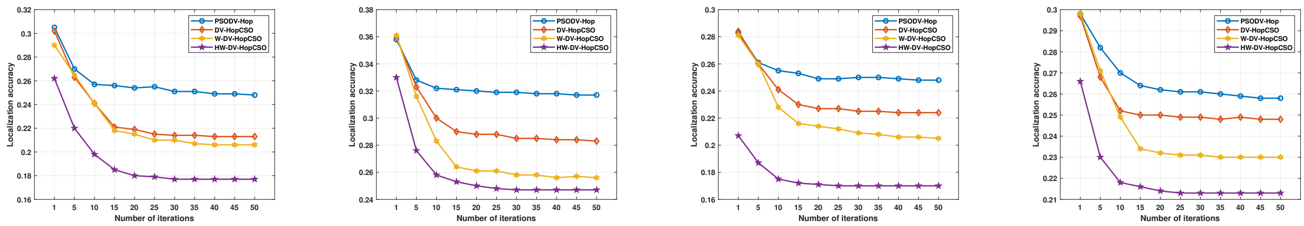
Table 3. *Cont.*

Network Topology	Random Square			H-Shaped			O-Shaped			W-Shaped		
Algorithm	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**	**AVG**
DV-HopCSO	0.017	0.595	0.201	0.021	0.804	0.233	0.019	0.564	0.191	0.017	0.772	0.212
W-DV-Hop	0.012	0.976	0.278	0.012	1.595	0.38	0.017	0.713	0.267	0.013	1.09	0.278
W-DV-HopCSO	0.012	0.708	0.182	0.012	0.969	0.252	0.017	0.61	0.186	0.013	0.752	0.187
HW-DV-Hop	0.034	0.816	0.243	0.009	0.821	0.269	0.024	0.785	0.244	0.006	0.71	0.286
HW-DV-HopCSO	0.016	0.646	0.174	0.009	0.729	0.200	0.008	0.456	0.194	0.006	0.538	0.201

4.4. Number of Iterations Analysis

In this section, we examine the influence of the iteration number on the presented localization approaches based on PSO and chicken swarm optimization (PSODV-Hop, DV-HopCSO, W-DV-HopCSO, and HW-DV-HopCSO) considering the same four distinct types of complex network topologies. In this configuration, the number of nodes remained constant at 200, the percentage of anchor nodes was set to 15%, and the communication range was 30 m, while the number of iteration was varied from 1 to 50. For CSO, the number of roosters was fixed at 8, the number of chicks at 1, and the number of hens at 15. Additionally, the particle velocity $Vmax$ and learning coefficients C_1 and C_2 for PSO were fixed at 10, 1.5, and 2, respectively, and the swarm size for both PSO and CSO was fixed at 20. Table 2 provides a summary of the parameters used for this simulation.

The results depicted in Figure 9 reveal that the HW-DV-Hop algorithm based on CSO is faster in terms of convergence speed in all four network topologies, in contrast to PSODV-Hop. The HW-DV-HopCSO algorithm demonstrates the robustness of using CSO for localization. It maintains a stable localization error compared to PSODV-Hop when increasing the maximum number of iterations, with the latter exhibiting an unstable localization error across different simulation scenarios and presented types of networks.



(**a**) (**b**) (**c**) (**d**)

Figure 9. Localization accuracy vs. number of iterations in different network topologies: (**a**) square random, (**b**) H-shaped, (**c**) O-shaped, and (**d**) W-shaped (200 sensor nodes, 15% of anchors, R = 30 m).

Table 4 provides a summary comparing the performance of the proposed algorithms to the existing ones as a function of the number of simulation trials.

Table 4. Simulation results and analysis in the random square, H-shaped, O-shaped, and W-shaped network topologies (200 nodes, 40 anchors, R = 30 m).

Network Topology	Random Square			H-Shaped			O-Shaped			W-Shaped		
Algorithm	**30**	**80**	**100**	**30**	**80**	**100**	**30**	**80**	**100**	**30**	**80**	**100**
DV-Hop	0.282	0.303	0.316	0.379	0.374	0.370	0.282	0.305	0.300	0.274	0.324	0.322
PSODV-Hop	0.272	0.243	0.247	0.288	0.286	0.287	0.236	0.257	0.256	0.225	0.243	0.245
MDV-Hop	0.258	0.225	0.230	0.327	0.288	0.286	0.208	0.199	0.197	0.299	0.255	0.265
DV-HopCSO	0.201	0.200	0.204	0.233	0.233	0.233	0.191	0.211	0.208	0.212	0.225	0.222
W-DV-Hop	0.278	0.288	0.298	0.38	0.367	0.365	0.267	0.289	0.281	0.278	0.310	0.312
W-DV-HopCSO	0.182	0.190	0.194	0.252	0.232	0.234	0.186	0.197	0.192	0.187	0.215	0.221
HW-DV-Hop	0.243	0.248	0.243	0.269	0.329	0.327	0.244	0.212	0.214	0.286	0.276	0.276
HW-DV-HopCSO	0.174	0.177	0.176	0.200	0.226	0.222	0.194	0.186	0.171	0.201	0.208	0.211

4.5. Analysis of Time Complexity

Wireless sensor networks comprise many small devices, each with a limited battery, and operate under stringent constraints in terms of capacity and energy consumption. Implementing complex algorithms in WSNs presents a considerable challenge in many fields. Due to these considerations, it is essential to conduct an evaluation of the time complexity of the proposed algorithms. This is particularly important when considering the NP-hard nature of the WSN localization problem. This scenario considered a WSN consisting of n sensor nodes with unknown locations and k anchors; the generation maximum for CSO and PSO were $MaxG1$ and $MaxG2$, respectively, the population size for CSO and PSO was $NP1$ and $NP2$, respectively, and the time interval for status updates in the chicken swarm was T. We employed the complexity to evaluate the time to completion required by an algorithm or, in the worst-case scenario, its overall execution time.

The time complexity of all presented algorithms was examined. During the first stage, all algorithms similarly create a hop count matrix; the time complexity of this stage is $O(k^2)$. In the second step, the complexity for DV-Hop and PSODV-Hop is $O(k^2)$ due to the way in which they calculate the average hop size. The W-DV-Hop, W-DV-HopCSO, HW-DV-Hop, HW-DV-HopCSO, and MDV-Hop algorithms all apply different weighted formulas to calculate the average hop size; despite these differences, the time complexity remains $O(k^2)$. In the third step, the complexity for DV-Hop, PSODV-Hop, and DV-HopCSO is $O(k * (n - K))$, as these algorithms all use the least squares approach to calculate the nodes' location. The time complexity for W-DV-Hop, W-DV-HopCSO, HW-DV-Hop, HW-DV-HopCSO, and MDV-Hop is $O(k * (n - K))$ in this stage, which is due to their reliance on the hyperbolic 2D integration technique rather than the trilateration approach.

In the fourth phase, the time complexity of PSODV-Hop is $O(MaxG1 * NP1 * (n - k))$, while for W-DV-HopCSO and HW-DV-HopCSO, which use CSO, it is $O(MaxG2 * NP2 * (n - k))$. An extra time cost of $O(k * NP1)$ is required due to the fitness function calculation for the PSO-based algorithms, while $O(k * NP2)$ is needed for the CSO-based proposed algorithms. In addition, $O(MaxG1 * NP1)$ and $O(Max21 * NP2)$ are required to update the locations, resulting in the time complexity of the W-DV-HopCSO and HW-DV-HopCSO algorithms, which use CSO, being $O(MaxG2 * NP2 * (n - k))$. Table 5 provides a summary of the time complexity for the tested algorithms.

Table 5. Time complexity analysis.

Algorithm	Time Complexity	Space Complexity
DV-Hop	$O(k^2)$	$O(1)$
PSODV-Hop	$O(MaxG1 * NP1 * (n - k))$	$O(NP1)$
MDV-Hop	$O(MaxG1 * NP1 * (n - k))$	$O(NP1)$
DV-HopCSO	$O(MaxG2 * NP2 * (n - k))$	$O(NP2)$
W-DV-Hop	$O(k^2)$	$O(1)$
W-DV-HopCSO	$O(MaxG2 * NP2 * (n - k))$	$O(NP2)$
HW-DV-Hop	$O(k^2)$	$O(1)$
HW-DV-HopCSO	$O(MaxG2 * NP2 * (n - k))$	$O(NP2)$

5. Conclusions

In this paper, we showcase several novel localization algorithms designed to accurately determine the locations of unknown sensor nodes. We proceed to evaluate and analyze these new algorithms across four distinct network topologies, focusing on metrics such as localization error and accuracy.

More specifically, we enhance the basic DV-Hop algorithm by incorporating additional steps to enhance the precision of position estimation for unknown nodes. Through extensive simulations, we thoroughly assess the performance of these new algorithms within four different random network topologies in static WSNs. These simulations involve a number of variations in parameters, including the number of nodes, percentage of anchor nodes, communication range, and maximum number of iterations. Our results highlight the exceptional performance of the HW-DV-HopCSO algorithm as compared to the standard DV-Hop and other competing algorithms in all the considered network experiments. Both the HW-DV-HopCSO and W-DV-HopCSO algorithms exhibit lower localization errors compared to the standard DV-Hop and other algorithms in the tested random square, H-shaped, O-shaped, and W-shaped network topologies. Furthermore, the simulation outcomes demonstrate that the HW-DV-HopCSO and W-DV-HopCSO algorithms consistently outperform the standard DV-Hop and other state-of-the-art approaches. In our ongoing research work, we intend to conduct future simulations to further evaluate the proposed algorithms in mobile wireless sensor networks. Additionally, the presented algorithms can be extended to more accurately calculate the locations of sensor nodes in three dimensions.

Author Contributions: Conceptualization, A.H.; methodology, A.H. and N.K.; software, A.H.; validation, A.H. and N.K.; formal analysis, A.H., M.-A.E.H. and N.K.; investigation, A.H.; resources, A.H.; data curation, A.H., M.-A.E.H. and J.E.K.; writing—original draft preparation, A.H. and N.K.; writing—review and editing, A.H., M.-A.E.H., J.E.K. and N.K.; visualization, A.H.; supervision, N.K.; project administration, A.H., N.K., M.-A.E.H. and J.E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zrelli, A. Hardware, software platforms, operating systems and routing protocols for Internet of Things applications. *Wirel. Pers. Commun.* **2022**, *122*, 3889–3912. [CrossRef]
2. Soldatos, J.; Gusmeroli, S.; Malo, P.; Di Orio, G. Internet of things applications in future manufacturing. In *Digitising the Industry Internet of Things Connecting the Physical, Digital and Virtual Worlds*; River Publishers: Nordjylland, Denmark, 2022; pp. 153–183.
3. Suresh, S.; Chakaravarthi, G. RFID technology and its diverse applications: A brief exposition with a proposed Machine Learning approach. *Measurement* **2022**, *195*, 111197. [CrossRef]
4. Nourildean, S.W.; Hassib, M.D.; Mohammed, Y. Internet of things based wireless sensor network: A review. *Indones. J. Electr. Eng. Comput. Sci.* **2022**, *27*, 246–261. [CrossRef]
5. Gliga, L.I.; Khemmar, R.; Chafouk, H.; Popescu, D. A Survey of Wireless Communication Technologies for an IoT-connected Wind Farm. *Wirel. Pers. Commun.* **2022**, *122*, 2253–2272. [CrossRef]
6. Jin, Y.; Cui, L. Discovering Influential Positions in RFID-Based Indoor Tracking Data. *Information* **2020**, *11*, 330. [CrossRef]
7. Alhayani, B.; Kwekha-Rashid, A.S.; Mahajan, H.B.; Ilhan, H.; Uke, N.; Alkhayyat, A.; Mohammed, H.J. 5G standards for the Industry 4.0 enabled communication systems using artificial intelligence: Perspective of smart healthcare system. *Appl. Nanosci.* **2023**, *13*, 1807–1817. [CrossRef]
8. Kabashkin, I. Availability of Services in Wireless Sensor Network with Aerial Base Station Placement. *J. Sens. Actuator Netw.* **2023**, *12*, 39. [CrossRef]
9. Asha, A.; Arunachalam, R.; Poonguzhali, I.; Urooj, S.; Alelyani, S. Optimized RNN-based performance prediction of IoT and WSN-oriented smart city application using improved honey badger algorithm. *Measurement* **2023**, *210*, 112505. [CrossRef]
10. Ghazal, T.M.; Hasan, M.K.; Alshurideh, M.T.; Alzoubi, H.M.; Ahmad, M.; Akbar, S.S.; Al Kurdi, B.; Akour, I.A. IoT for Smart Cities: Machine Learning Approaches in Smart Healthcare—A Review. *Future Internet* **2021**, *13*, 218. [CrossRef]
11. Oğur, N.B.; Al-Hubaishi, M.; Çeken, C. IoT data analytics architecture for smart healthcare using RFID and WSN. *ETRI J.* **2022**, *44*, 135–146. [CrossRef]
12. Khalifeh, A.; Gupta, M.; Almomani, O.; Khasawneh, A.M.; Darabkh, K.A. Smart remote sensing network for early warning of disaster risks. In *Nanotechnology-Based Smart Remote Sensing Networks for Disaster Prevention*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 303–324.
13. Hadir, A.; Zine-Dine, K.; Bakhouya, M.; El Kafi, J.; El Ouadghiri, D. Towards an integrated geographic routing approach using estimated sensors position in WSNs. In Proceedings of the 2018 International Conference on High Performance Computing & Simulation (HPCS), Orleans, France, 16–20 July 2018; pp. 943–948.
14. Naghibi, M.; Barati, H. EGRPM: Energy efficient geographic routing protocol based on mobile sink in wireless sensor networks. *Sustain. Comput. Inform. Syst.* **2020**, *25*, 100377. [CrossRef]
15. Prasanth, A.; Jayachitra, S. A novel multi-objective optimization strategy for enhancing quality of service in IoT-enabled WSN applications. *Peer-Netw. Appl.* **2020**, *13*, 1905–1920. [CrossRef]
16. Arafat, M.Y.; Alam, M.M.; Moh, S. Vision-based navigation techniques for unmanned aerial vehicles: Review and challenges. *Drones* **2023**, *7*, 89. [CrossRef]

17. Singh, P.; Mittal, N.; Salgotra, R. Comparison of range-based versus range-free WSNs localization using adaptive SSA algorithm. *Wirel. Netw.* **2022**, *28*, 1625–1647. [CrossRef]
18. Rizk, H.; Elmogy, A.; Yamaguchi, H. A robust and accurate indoor localization using learning-based fusion of Wi-Fi RTT and RSSI. *Sensors* **2022**, *22*, 2700. [CrossRef] [PubMed]
19. Aubry, A.; Babu, P.; De Maio, A.; Fatima, G.; Sahu, N. A Robust Framework to Design Optimal Sensor Locations for TOA or RSS Source Localization Techniques. *IEEE Trans. Signal Process.* **2023**, *71*, 1293–1306. [CrossRef]
20. Wang, W.; Wang, G.; Ho, K.; Huang, L. Robust TDOA localization based on maximum correntropy criterion with variable center. *Signal Process.* **2023**, *205*, 108860. [CrossRef]
21. Chen, H.; Cao, L.; Yue, Y. TDOA/AOA Hybrid Localization Based on Improved Dandelion Optimization Algorithm for Mobile Location Estimation Under NLOS Simulation Environment. *Wirel. Pers. Commun.* **2023**, *131*, 2747–2772. [CrossRef]
22. Dou, Z.; Yao, Z.; Zhang, Z.; Lu, M. A Lidar-assisted Self-localization Technology for Indoor Wireless Sensor Networks. *IEEE Internet Things J.* **2023**, *10*, 17515–17529. [CrossRef]
23. Niculescu, D.; Nath, B. DV based positioning in ad hoc networks. *Telecommun. Syst.* **2003**, *22*, 267–280. [CrossRef]
24. Latha, T.S.; Rekha, K.B.; Ferede, A.W. Lightweight APIT with Bat Optimization with Simulated Annealing Localization for Resource-Constrained Sensor Networks. *Wirel. Commun. Mob. Comput.* **2023**, *2023*, 7982038. [CrossRef]
25. Qu, S.; Bao, Z.; Yin, Y.; Yang, X. MineBL: A Battery-Free Localization Scheme with Binocular Camera for Coal Mine. *Sensors* **2022**, *22*, 6511. [CrossRef] [PubMed]
26. Tripathy, P.; Khilar, P. An ensemble approach for improving localization accuracy in wireless sensor network. *Comput. Netw.* **2022**, *219*, 109427. [CrossRef]
27. Liu, J.; Liu, M.; Du, X.; Stanimirovi, P.S.; Jin, L. An improved DV-Hop algorithm for wireless sensor networks based on neural dynamics. *Neurocomputing* **2022**, *491*, 172–185. [CrossRef]
28. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A new bio-inspired algorithm: Chicken swarm optimization. In Proceedings of the Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, 17–20 October 2014; Proceedings, Part I 5; Springer: Berlin/Heidelberg, Germany, 2014; pp. 86–94.
29. Chen, T.; Hou, S.; Sun, L.; Sun, K. An Enhanced DV-Hop Localization Scheme Based on Weighted Iteration and Optimal Beacon Set. *Electronics* **2022**, *11*, 1774. [CrossRef]
30. Yanfei, J.; Kexin, Z.; Liquan, Z. Improved DV-hop location algorithm based on mobile anchor node and modified hop count for wireless sensor network. *J. Electr. Comput. Eng.* **2020**, *2020*, 9275603. [CrossRef]
31. Shi, Q.; Wu, C.; Xu, Q.; Zhang, J. Optimization for DV-Hop type of localization scheme in wireless sensor networks. *J. Supercomput.* **2021**, *77*, 13629–13652. [CrossRef]
32. Han, D.; Yu, Y.; Li, K.C.; de Mello, R.F. Enhancing the sensor node localization algorithm based on improved DV-hop and DE algorithms in wireless sensor networks. *Sensors* **2020**, *20*, 343. [CrossRef]
33. Zhang, S.; Li, J.; He, B.; Chen, J. LSDV-Hop: Least Squares Based DV-Hop Localization Algorithm for Wireless Sensor Networks. *J. Commun.* **2016**, *11*, 243–248. [CrossRef]
34. Messous, S.; Liouane, H. Online sequential DV-hop localization algorithm for wireless sensor networks. *Mob. Inf. Syst.* **2020**, *2020*, 8195309. [CrossRef]
35. Song, G.; Tam, D. Two novel DV-Hop localization algorithms for randomly deployed wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 187670. [CrossRef]
36. Tomic, S.; Mezei, I. Improvements of DV-Hop localization algorithm for wireless sensor networks. *Telecommun. Syst.* **2016**, *61*, 93–106. [CrossRef]
37. Singh, S.P.; Sharma, S.C. A PSO based improved localization algorithm for wireless sensor network. *Wirel. Pers. Commun.* **2018**, *98*, 487–503. [CrossRef]
38. Rabhi, S.; Ouahab, A.; Baddou, S.; Chetouah, K. An Improved Algorithm Based on Chicken Swarm Optimization for Localization in Wireless Sensor Networks. *Adv. Model. Anal. B* **2021**, *64*, 34–39. [CrossRef]
39. Hadir, A.; Zine-Dine, K.; Bakhouya, M.; El Kafi, J. An optimized DV-hop localization algorithm using average hop weighted mean in WSNs. In Proceedings of the 2014 5th Workshop on Codes, Cryptography and Communication Systems (WCCCS), El Jadida, Morocco, 27–28 November 2014; pp. 25–29.
40. Hadir, A.; Zine-Dine, K.; Bakhouya, M.; Kafi, J.E. Novel localisation algorithms in wireless sensor networks. *Int. J. Wirel. Mob. Comput.* **2019**, *16*, 80–96. [CrossRef]
41. Hadir, A.; Regragui, Y.; Garcia, N.M. Accurate range-free localization algorithms based on PSO for wireless sensor networks. *IEEE Access* **2021**, *9*, 149906–149924. [CrossRef]
42. Nagpal, R.; Shrobe, H.; Bachrach, J. Organizing a global coordinate system from local information on an ad hoc sensor network. In Proceedings of the Information Processing in Sensor Networks, Palo Alto, CA, USA, 22–23 April 2003; pp. 333–348.
43. Rockmore, D.N.; Healy, D.M., Jr. *Modern Signal Processing*; Cambridge University Press: Cambridge, UK, 2004; Volume 46.
44. Chan, Y.T.; Ho, K.C. A simple and efficient estimator for hyperbolic location. *IEEE Trans. Signal Process.* **1994**, *42*, 1905–1915. [CrossRef]
45. Gad, A.G. Particle swarm optimization algorithm and its applications: A systematic review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2531–2561. [CrossRef]

46. Xie, X.; Tang, Z.; Cai, J. The multi-objective inspection path-planning in radioactive environment based on an improved ant colony optimization algorithm. *Prog. Nucl. Energy* **2022**, *144*, 104076. [CrossRef]
47. Kaya, E.; Gorkemli, B.; Akay, B.; Karaboga, D. A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105311. [CrossRef]
48. Singh, S.P.; Sharma, S.C. Implementation of a PSO based improved localization algorithm for wireless sensor networks. *IETE J. Res.* **2019**, *65*, 502–514. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.