

Article

Effectiveness of Data Augmentation for Localization in WSNs Using Deep Learning for the Internet of Things

Jehan Esheh * and Sofiene Affes * 

EMT Centre (Energy, Materials and Telecommunications), INRS (Institut National de la Recherche Scientifique), Université du Québec, Montréal, QC H5A 1K6, Canada

* Correspondence: jehan.esheh@inrs.ca (J.E.); sofiene.affes@inrs.ca (S.A.)

Abstract: Wireless sensor networks (WSNs) have become widely popular and are extensively used for various sensor communication applications due to their flexibility and cost effectiveness, especially for applications where localization is a main challenge. Furthermore, the Dv-hop algorithm is a range-free localization algorithm commonly used in WSNs. Despite its simplicity and low hardware requirements, it does suffer from limitations in terms of localization accuracy. In this article, we develop an accurate Deep Learning (DL)-based range-free localization for WSN applications in the Internet of things (IoT). To improve the localization performance, we exploit a deep neural network (DNN) to correct the estimated distance between the unknown nodes (i.e., position-unaware) and the anchor nodes (i.e., position-aware) without burdening the IoT cost. DL needs large training data to yield accurate results, and the DNN is no stranger. The efficacy of machine learning, including DNNs, hinges on access to substantial training data for optimal performance. However, to address this challenge, we propose a solution through the implementation of a Data Augmentation Strategy (DAS). This strategy involves the strategic creation of multiple virtual anchors around the existing real anchors. Consequently, this process generates more training data and significantly increases data size. We prove that DAS can provide the DNNs with sufficient training data, and ultimately making it more feasible for WSNs and the IoT to fully benefit from low-cost DNN-aided localization. The simulation results indicate that the accuracy of the proposed (Dv-hop with DNN correction) surpasses that of Dv-hop.



Citation: Esheh, J.; Affes, S.

Effectiveness of Data Augmentation for Localization in WSNs Using Deep Learning for the Internet of Things. *Sensors* **2024**, *24*, 430. <https://doi.org/10.3390/s24020430>

Academic Editor: Gianluigi Ferrari

Received: 20 November 2023

Revised: 14 December 2023

Accepted: 22 December 2023

Published: 10 January 2024



Copyright: © 2024 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/>).

Keywords: range-free localization; neural networks; data augmentation; wireless sensor networks

1. Introduction

In recent decades, with the advancements in IoT technologies, the intelligent perception and management of objects have become achievable through the connection of things and people [1]. WSNs have played an increasingly significant role in the IoT by facilitating the real-time sensing, collecting, and processing of information. The inherent characteristics of node location make it an essential prerequisite for many functions. During the last decade, this topic has motivated extensive research endeavors that have resulted in several interesting localization algorithms [2]. As the demand for location-based services continues to grow, the accuracy of node localization significantly impacts various application areas, such as city surveillance, and smart homes [3,4]. So far, many localization algorithms have been proposed in the literature and mainly fall into two categories: range-based and range-free algorithms. Range-based localization utilizes the measurements of the received signal attributes such as angle of arriving signal (AOA) [5], received signal strength (RSS) [6], and time of arrival (TOA) [7]. While range-based algorithms are known for their high accuracy, they are impractical for WSNs due to their high-power requirements for communication between anchors and regular nodes, especially in the case of small battery-powered units. Furthermore, these algorithms are susceptible to interference and fading, often necessitating additional hardware, thereby burdening both the WSN and the IoT costs. Range-free

localization techniques are variations of the well-known Dv-hop algorithm that simply converts the numbers of hops into coordinates [8–10]. Such techniques do not require any additional hardware, in contrast to range-based methods, but are relatively much less accurate. To improve localization performance, researchers have recently resorted to machine learning (ML) or DL. Several ML techniques have been investigated in this context such as, namely, support vector machine (SVM), artificial neural network (ANN), etc. [11–14]. The key hurdle these techniques face in common is the requirement of large training data sets. The larger they are, the more accurate they will be in correcting the estimated node positions. However, such data, generated very often by anchors deployed in few numbers due to the expensive GPS technology they integrate, are relatively scarce, thereby limiting any potential ML-driven performance improvement. Therefore, to improve localization accuracy, we develop a precise, and cost-efficient DNN-based range-free localization approach for WSN applications in the IoT. To tackle the issue of limited data for training the proposed DNN, we are also working on an efficient DAS.

The rest of the paper is organized as follows. Section 2 introduces the range-free localization process. Section 3 describes the implementation of data augmentation in WSN application. The architecture processing of the DNN is described in Section 4. In Section 5, several experiments are performed, and the experimental results are analyzed. Finally, in Section 6 we conclude the paper with a summary.

2. Localization Process

The aim of sensor localization is to determine the locations of unknown nodes. The localization process identifies the positions of these unknown nodes based on input data. In our case, the input data consists of the locations of both real anchors and their virtual counterparts, along with the unknown node. An overview of the localization process is depicted in Figure 1.

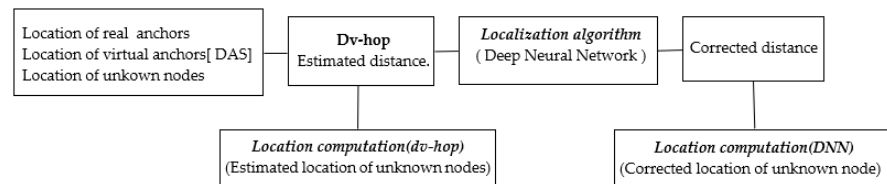


Figure 1. Overview of localization process.

This section explains the estimated distance and location computation process, while the remaining steps will be covered in the next section.

For estimation of the distance between anchor and unknown nodes, we assume each node communicates with an anchor node through a multi-hop path by using a localization algorithm [10]. Firstly, all unknown nodes in the network obtain minimal hop counts to every anchor node. During the second phase, when an anchor node obtains hop counts to other anchors, it calculates an average distance for one hop, which is subsequently disseminated to the entire network. Anchor node i estimates the average hop size using the following equation:

$$\text{Hopsize}_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{ij \neq h_{ij}}}, \quad (1)$$

where (x_i, y_i) and (x_j, y_j) are the known coordinates of anchors i and j , respectively, and h_{ij} is the minimum number of hops between them.

Upon receiving the hop size from the anchor nodes with the least hops between them, every unknown node computes its distance (d_i) to each anchor node I using the hop size

denoted earlier in Equation (1) as Hopsize_i and the minimum hop count denoted here as hops as follows:

$$d_i = \text{hops} \times \text{Hopsize}_i. \quad (2)$$

Hence, the location of the unknown node can be estimated by solving the following set of equations:

$$\begin{cases} (\hat{x} - x_1)^2 + (\hat{y} - y_1)^2 = d_1^2 \\ (\hat{x} - x_2)^2 + (\hat{y} - y_2)^2 = d_2^2 \\ \vdots \\ (\hat{x} - x_n)^2 + (\hat{y} - y_n)^2 = d_n^2 \end{cases}; \quad (3)$$

where (x_i, y_i) denote the coordinates of anchor $i = 1, \dots, n$ and $(\hat{x}, \hat{y})$ are the coordinates of the unknown node. Indeed, Equation (3) could be linearized and solved under the minimum mean square error (MMSE) criterion to estimate the coordinates of the unknown node $(\hat{x}, \hat{y})$ as follows:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \frac{-1}{2} (\psi^T \psi)^{-1} \psi^T \varphi, \quad (4)$$

where:

$$\psi = \begin{bmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \vdots & \vdots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{bmatrix}, \quad (5)$$

and

$$\varphi = \begin{bmatrix} d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_2^2 - d_n^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}. \quad (6)$$

3. Data Augmentation in WSN Application

Typical augmentation techniques applied to images involve a range of transformations such as translation, blurring, flipping, rotation, and the introduction of various types of noise to data samples. These techniques are well established in the field, and diverse DASs are tailored to specific problems. For example, in the context of the MNIST database of handwritten digits, researchers have explored augmentation techniques [15]. In the field of machine learning, especially for researchers working with techniques like Generative Adversarial Networks (GANs), the limited availability of large datasets for effective training poses a significant challenge. In response to this challenge, a novel concept known as “virtual big data” is introduced [16]. This concept involves the generation of synthetic or virtual datasets that mimic the characteristics of real-world data, offering a solution in situations where obtaining extensive real-world data is impractical. In consideration of the small datasets of chemical reactions, the data-driven model suffers from the difficulty of low accuracy in the prediction tasks of chemical reactions. To tackle this, the model integrated with the strategies of data augmentation [17]. The data augmentation is used to improve the performance of data-driven reaction prediction models by increasing the sample size using fake data augmentation [18].

In what follows, we will employ a similar approach by utilizing data augmentation to increase the dataset. This involves creating multiple copies of virtual anchors for each real anchor around its position, as illustrated in Figure 2. The coordinates of these new virtual anchors are mathematically represented by Equation (7). This data augmentation technique is applied to enhance the dataset, generating additional instances of virtual anchors to increase the training data.

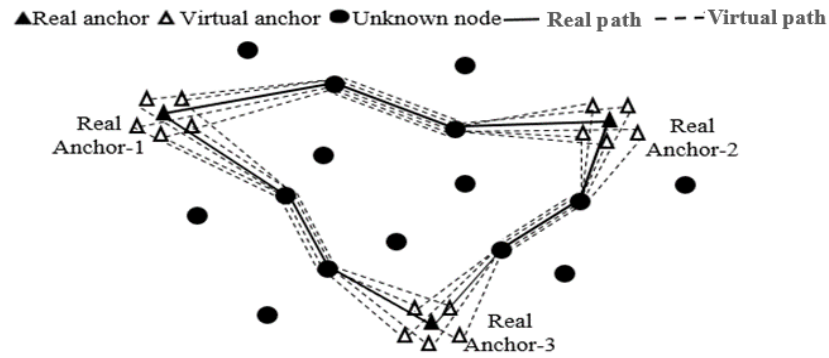


Figure 2. Creation of virtual anchors.

As shown in Figure 2, there are three anchors ($A = 3$), and each anchor is surrounded by five virtual anchors ($V = 5$). The coordinates $(\tilde{x}_k, \tilde{y}_k)$ of the virtual anchor near the real anchor i can be obtained by adding a span multiplied by a random Gaussian variation $(\Delta x_k, \Delta y_k)$, denoted as (Vcx, Vcy) , as depicted in Algorithm 1 (steps 6 and 7), to the coordinates of the anchor i (x_i, y_i) for $(i = 1, \dots, A)$ and $(k = 1, \dots, V)$ as follows:

$$\begin{cases} (\tilde{x}_1, \tilde{y}_1) = (x_i \pm \Delta x_1, x_i \pm \Delta y_1) \\ (\tilde{x}_2, \tilde{y}_2) = (x_i \pm \Delta x_2, x_i \pm \Delta y_2) \\ \vdots \\ (\tilde{x}_k, \tilde{y}_k) = (x_i \pm \Delta x_k, x_i \pm \Delta y_k) \end{cases} \quad (7)$$

where A and V denote the numbers of real and virtual anchors, respectively.

Algorithm 1. Generator for the coordinates of the virtual anchors

Input: BorderLength, NodeAmount, BeaconAmount, Span, V ;

C (generation of coordinates of all nodes);

Beacon = [$C(1,1:\text{BeaconAmount})$; $C(2,1:\text{BeaconAmount})$]; % Coordinates of real anchors

Output:

```

1. for  $V = 5:5:25$ 
2.      $k = \text{BeaconAmount} \times V$ ;
3.     for  $i = \text{NodeAmount} - 1 : \text{BeaconAmount} + 1$ ;
4.          $C(:, i + k) = C(:, i)$ ; % Shift unknown nodes to leave room for virtual ones
5.     end
6.      $n = 1$ ;
7.     for  $i = 1 : V + 1 : k + \text{BeaconAmount}$ 
8.          $Vcx = \text{Span} \times \text{randn}(1, V)$ ;
9.          $Vcy = \text{Span} \times \text{randn}(1, V)$ ;
10.        for  $j = 0 : V$ 
11.            if  $(j = 0)$ 
12.                 $C1(:, i + j) = \text{Beacon}(:, n)$ ;
13.            else
14.                 $C(1, i + j) = C(1, i) + Vcx(1, j)$ ;
15.                 $C(2, i + j) = C(2, i) + Vcy(1, j)$ ;
16.                Bind  $C(:, i + j)$  within  $(\text{BorderLength})^2$  square if outside
17.            end
18.        end
19.         $n = n + 1$ ;
20.    end
21.     $n = n - 1$ ; % Total number of real and virtual anchors  $n = \text{BeaconAmount} \times (V + 1)$ 

```

This augmentation strategy involving virtual anchors significantly expands the dataset, providing a more extensive and diverse set of training instances for the proposed DNN model. The input data for the proposed DNN framework is the distance between the real

anchors and their virtual anchors, and the unknown nodes. The data are then structured or shaped to be in a format suitable for training, often represented in matrix form. The format of the input data for the DNN is a single array form, as depicted in Figure 3.

$$\begin{array}{c}
 \begin{array}{c}
 B_1 \\
 VB_{11} \\
 VB_{12} \\
 \vdots \\
 VB_{1k} \\
 B_2 \\
 VB_{21} \\
 VB_{22} \\
 \vdots \\
 VB_{2k} \\
 \vdots \\
 B_i \\
 VB_{i1} \\
 VB_{i2} \\
 \vdots \\
 VB_{ik}
 \end{array}
 \begin{array}{c}
 U1 \quad U2 \quad \dots \quad U_j \\
 \left[\begin{array}{cccc}
 d_{11} & d_{12} & \dots & d_{1j} \\
 d_{(11)1} & d_{(11)2} & \dots & d_{(11)j} \\
 d_{(12)1} & d_{(12)2} & \dots & d_{(12)j} \\
 \vdots & \vdots & \ddots & \vdots \\
 d_{(1k)1} & d_{(1k)2} & \dots & d_{(1k)j} \\
 d_{21} & d_{22} & \dots & d_{2j} \\
 d_{(21)1} & d_{(21)2} & \dots & d_{(21)j} \\
 d_{(22)1} & d_{(22)2} & \dots & d_{(22)j} \\
 \vdots & \vdots & \ddots & \vdots \\
 d_{(2k)1} & d_{(2k)2} & \dots & d_{(2k)j} \\
 \vdots & \vdots & \ddots & \vdots \\
 d_{i1} & d_{i2} & \dots & d_{ij} \\
 d_{(i1)1} & d_{(i1)2} & \dots & d_{(i1)j} \\
 d_{(i2)1} & d_{(i2)2} & \dots & d_{(i2)j} \\
 \vdots & \vdots & \ddots & \vdots \\
 d_{(ik)1} & d_{(ik)2} & \dots & d_{(ik)j}
 \end{array} \right]
 \end{array}
 \end{array}
 \quad (A \times V + A) \times N_u$$

$$DNN \text{ Input distance} = [d_{11} \ d_{12}, \dots, d_{1j}; d_{(11)1} \ d_{(11)2} \ \dots \ d_{(11)j}; \dots \dots \dots; d_{(ik)1} d_{(ik)2} \ \dots, d_{(ik)j}]_{D_t,1}$$

Figure 3. The augmented input data format.

In Figure 3, B_i and VB_{ik} represent the locations of the real anchors and virtual anchors, respectively, where ($i = 1, \dots, A$), A being the total number of real anchors, and ($k = 1, \dots, V$), and V being the total number of virtual anchors. U_j denotes the unknown nodes, where ($j = 1, \dots, N_u$), with N_u being the total number of unknown nodes. The distance between (the real anchors and their virtual anchors), and the unknown nodes, is denoted as $d_{(ik)j}$. The training data size (D_t) is described in Equation (8):

$$D_t = (A \times V + A) \times N_u. \quad (8)$$

In this scenario, the dataset composition is determined by the number of real anchors ($A = 5$), unknown nodes ($N_u = 95$), and the presence or absence of virtual anchors ($V = 0$), the total data size (D_t) is calculated as 475 using the formula of Equation (7). However, introducing virtual anchors, as exemplified with $V = 20$, leads to a substantial increase in the total data size, reaching up to 9975, as shown in Table 1 and Figure 4.

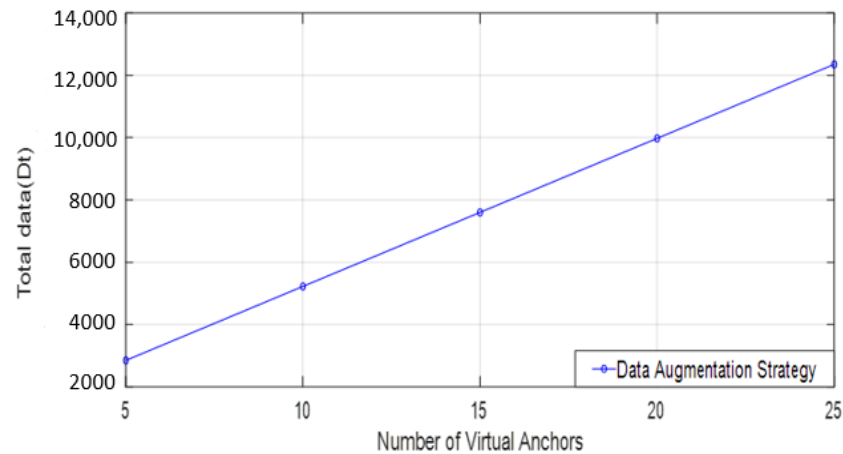


Figure 4. Total data size of virtual augmentation strategy.

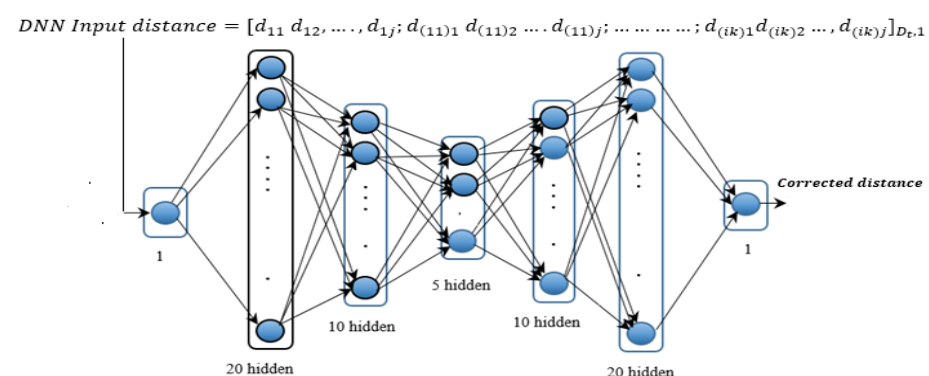
Table 1. Training data size.

Virtual Anchors	Anchors with Augmentation	Total Data (D_T)
5	30	2850
10	55	5225
15	80	7600
20	105	9975
25	130	12,350

4. DNN-Based Estimated Distance Correction

To obtain a deeper understanding of DNNs, it is necessary to revisit the basics of ANNs. ANNs have been used in several areas, such as engineering applications and WSN applications [14,19]. Several types of neural networks are described in [13]. Generally, an ANN can be defined as a system or mathematical model that consists of many nonlinear artificial neurons running in parallel and may be generated as one layered or multilayered. An ANN consists of a network of neurons organized in input layers, output layers and hidden layers. Different types of networks can be implemented by varying the structure of the weights and the activation functions of the neurons. Neural network systems can learn how to approximate relationships between inputs and outputs without being overcome by the complexity and size of the problem. The training of ANN using the backpropagation (BP) technique typically occurs in three main steps: the feedforward of input training, the backpropagation of the error, and the update of weights and biases.

ANNs, as highly efficient computational methods, find widespread applications in knowledge representation, machine learning, and predicting output responses in complex systems [20]. Recent advancements have underscored their effectiveness and led to notable achievements in diverse domains [21]. In the domain of ANN training, various processes have been employed [22]. This method is characterized by two essential stages: forward propagation, and backward propagation [23]. Localization system based on WSN and backpropagation-based BP-ANN have been practically implemented to detect and determine the position of an Alzheimer's patient in an indoor environment [14]. To achieve a minimal localization error, a thorough exploration of various DNN architectures was undertaken, considering different combinations of hidden layers and neurons. Through this iterative process, an optimal DNN architecture emerged, characterized by one input layer (referred to as DNN input distance), five hidden layers with neuron counts of (20, 10, 5, 10, and 20,) and a single output layer (referred to as corrected distance), as depicted in Figure 5.

**Figure 5.** Proposed DNN framework.

The process of localization accuracy involved the meticulous collection of DNN data input for the purpose of training, testing, and validation. The dataset was judiciously partitioned, allocating 70% for training, 15% for testing, and an additional 15% for validation. The iterative process for training the DNN was extended up to 1000 iterations, a crucial

step undertaken to enable the DNN to reach an optimal normalized root-mean-square error (NRMSE).

5. Simulation and Performance Analysis

The experimental region was defined by the parameters outlined in Table 2; the nodes were randomly deployed. We also carried out a series of experiments. This section presents one case as an example of node deployment under the influence of network settings, specifically by changing the amount of virtual anchor and span ranges, as illustrated in Figure 6a. This figure provides a visualization of unknown nodes enclosed by 5 real anchors, each surrounded by 20 virtual anchors, with a span equal to 1 m. Figure 6b showcases another scenario where unknown nodes and 5 real anchors are surrounded by 20 virtual anchors around each real anchor, but with a span equal to 6 m. These visual representations offer a concrete example of node deployment configurations, demonstrating the impact of the amount of virtual anchor, and varying span ranges in the simulation setup.

Table 2. Experimental parameter.

Contents of Experiments			
N_u	Number of unknown nodes	95	
A	Number of real anchors	5	
V	Number of virtual anchors	5, 10, 15, 20, 25	
S_a	Square area	$100 \times 100 \text{ m}^2$	
R	Communications range	15 m, 20 m, 25 m, 30 m	
σ	Node density	0.01	
S	Span	1 m, 3 m, 6 m, 9 m, 12 m	

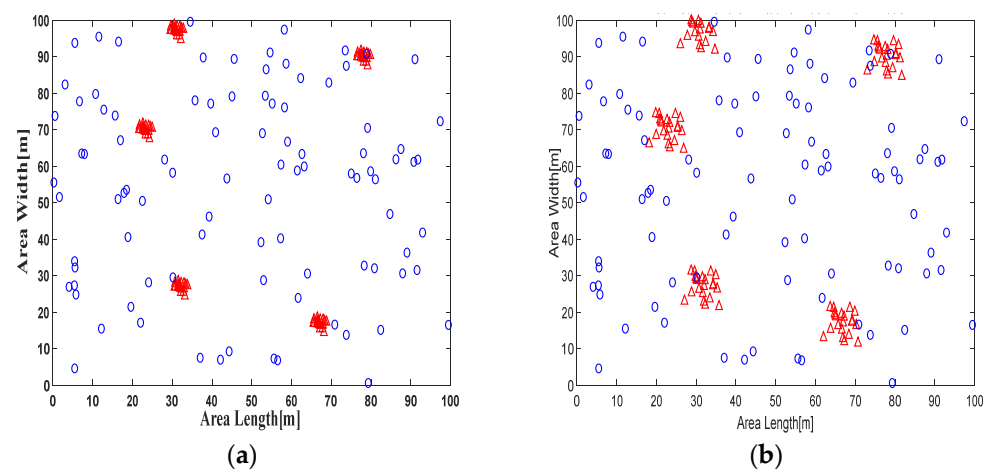


Figure 6. Sensor network configuration $\circ$: unknown nodes $\triangle$: real and virtual anchors (a,b).

5.1. Experiment Results

To verify the performance of the proposed Dv-hop + DNN correction algorithm, the simulations were separately carried out on the Dv-hop algorithm and the improved Dv-hop + DNN correction algorithm across various values of spans and node communication ranges within a randomly selected square area. The evaluation metric employed for this comparison was the normalized root-mean-square error (NRMSE), calculated using Equation (9):

$$\text{NRMSE} = \left(\sum_{i=0}^U \sqrt{(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2} / (N_u \times R) \right) \quad (9)$$

where (x_i, y_i) denotes the real position of the unknown node, and $(\hat{x}_i, \hat{y}_i)$ represents the estimated position (Dv-hop) or corrected position (Dv-hop + DNN correction) of the unknown node. The remaining parameters are defined in Table 1.

The comparison of the localization NRMSE for two different algorithms, Dv-hop and a proposed DNN-correction algorithm, are presented in Figure 7, and a comparison was made of various values of virtual anchors, in Figure 8 concerning span, and in Figure 9 with changes in the communication range. These figures illustrate that the accuracies of all algorithms improved, as expected, with an increasing span and communication range. However, the proposed approach consistently outperformed the Dv-hop algorithm in terms of localization accuracy. The error in localizing unknown nodes decreased with a higher number of virtual anchor nodes in the proposed approach. The positioning errors were smaller with the proposed approach compared to the Dv-hop algorithm, even when the number of virtual anchor nodes was the same. Gradually changing the number of virtual anchor nodes yielded different results in various scenarios. The proposed approach demonstrates increasing superiority over the Dv-hop approach as the number of virtual anchor nodes rises. Importantly, under the same conditions, the NRMSE of the proposed approach with an augmentation system was smaller than that of the Dv-hop algorithm.

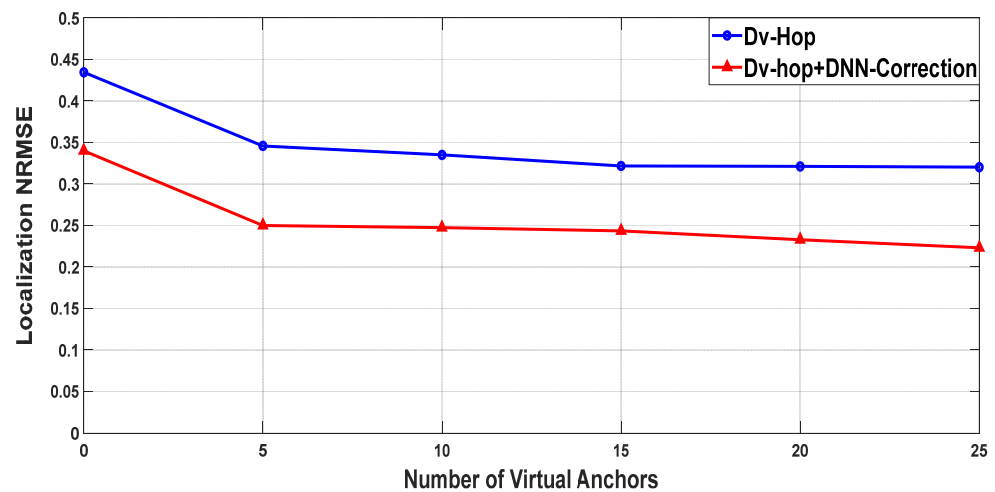


Figure 7. NRMSE performance with different virtual anchors.

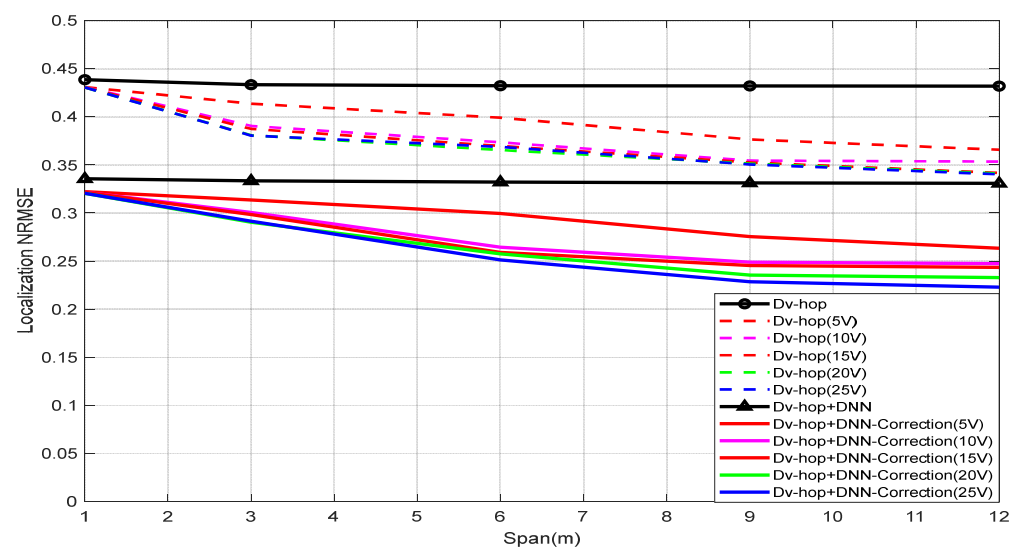


Figure 8. NRMSE performance versus span values.

Figure 10 illustrates the cumulative distribution function (CDF) of the localization NRMSE with experiment parameters, as shown in Table 2. From Figure 10a, using the proposed algorithm Dv-hop + DNNs correction without augmentation system 70%, and with Dv-hop + DNNs correction for (5 V, 10 V, 15 V, 20 V, and 25 V), 90%, 94%, 95%, 96%, and 98%, respectively, of the sensors could estimate their position with an NRMSE less than 0.2. In contrast, from Figure 10b with Dv-hop for (5 V, 10 V, 15 V, 20 V, and 25 virtual), 76%, 82%, 82%, and 91%, respectively, of the sensors could estimate their position, and only about 50% with Dv-hop without adding virtual anchors. This proves even more the accuracy of the proposed localization algorithm. Figure 10c shows when the span is very small (without and with 5 to 25 virtual), using the Dv-hop + DNN correction algorithm, 71% of nodes estimate their position within a NRMSE value of less than 0.2, and no more than 50% using Dv-hop (without and with 5 to 25 virtual), as shown in Figure 10d. Meanwhile, Figures 10e and 10f show the effect of communication range when reduced from 30 m to 20 m, for two case Dv-hop and Dv hop + DNNs. The Dv-hop + DNNs correction without augmentation system 50%, and with Dv-hop + DNNs correction for (5 V, 10 V, 15 V, 20 V, and 25 V), 78%, 81%, 82%, 86%, and 84%, respectively, of the sensors could estimate their position with an NRMSE less than 0.2. In contrast, from Figure 10f with Dv-hop for (5 V, 10 V, 15 V, 20 V, and 25 V), 55%, 62%, 69%, and 67% of the nodes achieve the same accuracy with and only about 50% with Dv-hop without adding virtual anchors. From the simulations, we observed that the proposed approach kept improving noticeably with a larger number of virtual anchor nodes up to 20 when the total training data size was 9975, as shown in Table 1. Performance gains started to saturate beyond that threshold.

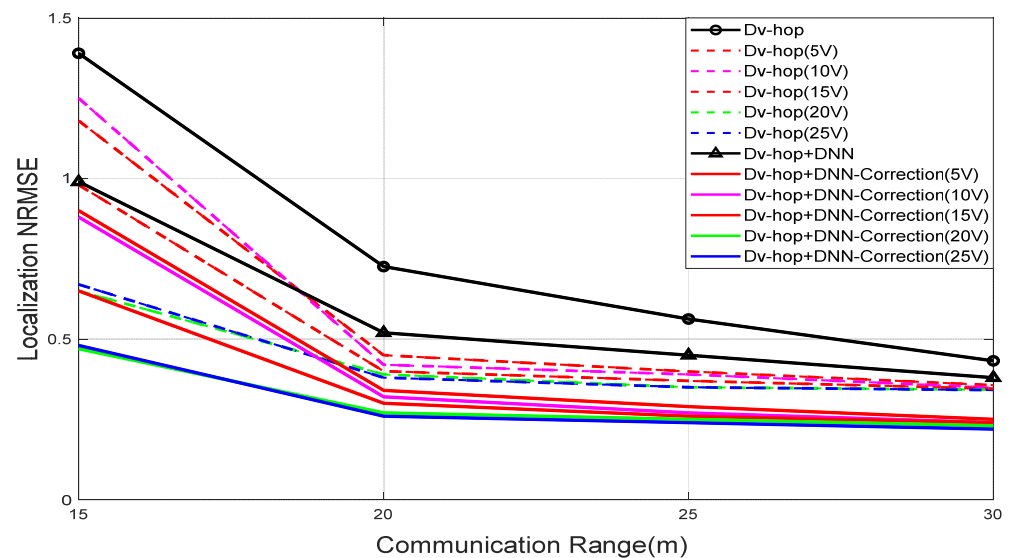


Figure 9. NRMSE performance versus communication range.

These further prove the efficiency of the Dv-hop + DNN algorithm, and that the DAS has a sufficient effect for correcting the location of coordinates of unknown nodes.

5.1.1. Effect of Span

The evaluation of the Dv-hop and Dv-hop + DNN correction algorithms was conducted with systematic adjustments made to the span (distance between real anchors and virtual anchors) at values of 1 m, 3 m, 6 m, 9 m, and 12 m. The foundational aspects of the WSN model, detailed in Table 2, remained unchanged throughout these experiments. The experimental outcomes, detailed in Tables 3 and 4, highlight the performance of the Dv-hop, Dv-hop + virtual, and Dv-hop + DNN correction algorithms under varying span values, assessed through NRMSE values. Remarkably, the NRMSE obtained by the proposed Dv-hop + DNN correction consistently ranked first, particularly when virtual anchors

were strategically introduced around the real anchors. This observation underscores the algorithm's superior performance in terms of both accuracy and cost efficiency.

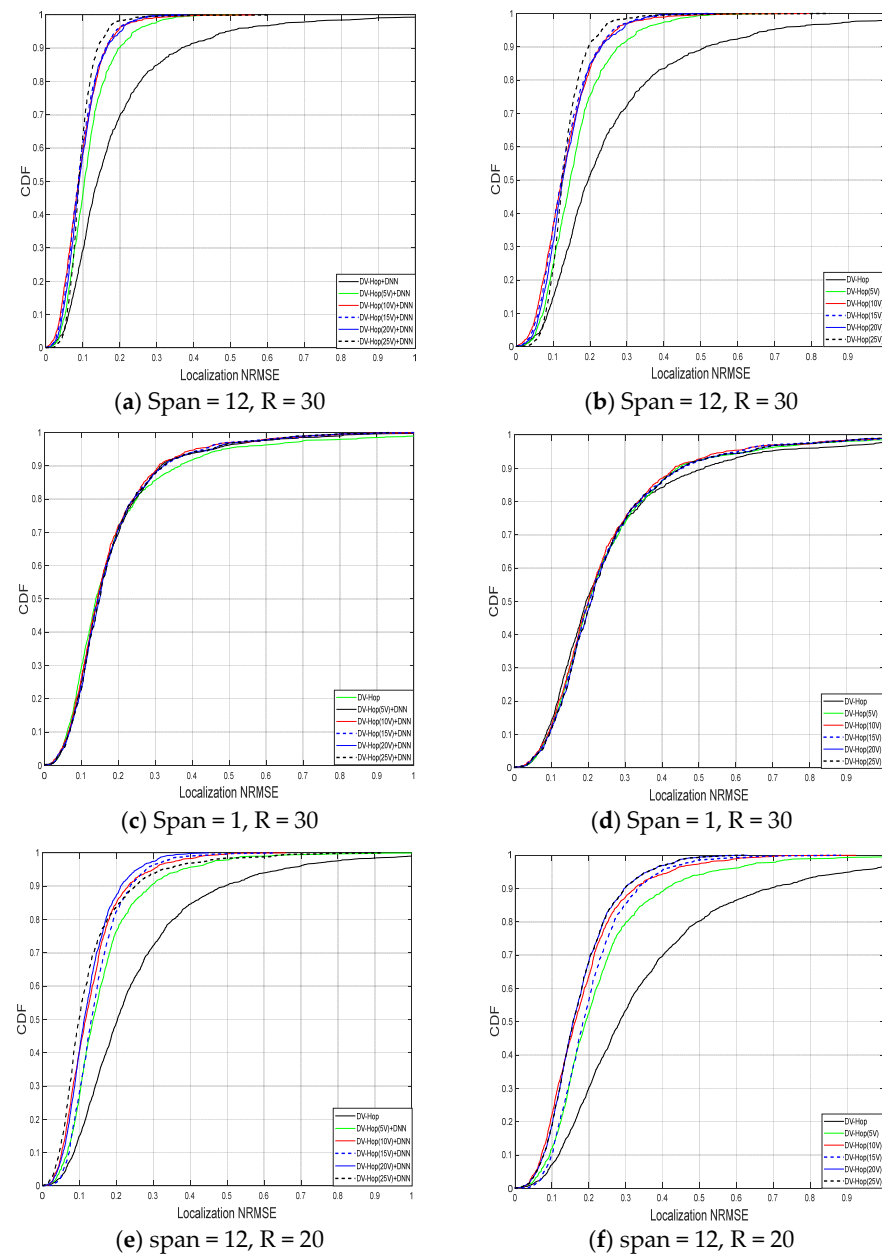


Figure 10. NRMSE CDFs for Dv-hop + DNN in (a,c,e) and Dv-hop in (b,d,f).

Table 3. Result of different values of span for Dv-hop, R = 30 m.

Span		1 m	3 m	6 m	9 m	12 m
NRMSE of	Dv-hop	43.87%	43.34%	43.24%	43.22%	43.20%
	Dv-hop (5 V)	43.09%	41.37%	39.91%	37.65%	36.58%
	Dv-hop (10 V)	43.09%	39.05%	37.35%	35.45%	35.35%
	Dv-hop (15 V)	43.07%	38.75%	36.97%	35.26%	34.17%
	Dv-hop (20 V)	43.06%	38.08%	36.55%	35.14%	34.11%
	Dv-hop (25 V)	43.06%	38.05%	36.87%	35.05%	34.02%

Table 4. Results of different values of span for Dv-hop + DNNs correction, R = 30 m.

Span		1 m	3 m	6 m	9 m	12 m
NRMSE of	Dv-hop + DNN	33.65%	33.05%	33.04%	33.04%	33.05%
	Dv-hop + DNN (5 V)	32.23%	31.35%	29.95%	27.55%	26.34%
	Dv-hop + DNN (10 V)	32.15%	30.05%	26.45%	24.89%	24.73%
	Dv-hop + DNN (15 V)	32.15%	29.85%	25.91%	24.54%	24.34%
	Dv-hop + DNN (20 V)	32.05%	29.05%	25.76%	23.55%	23.29%
	Dv-hop + DNN (25 V)	32.05%	29.15%	25.12%	22.85%	22.30%

5.1.2. Effect of Node Communication Range

To test the Dv-hop and Dv-hop + DNN correction algorithms under the effect of communication range. The communication range (R) of the nodes was systematically varied across different values, 15 m, 20 m, 25 m, and 30 m, while keeping the rest of the WSN model unchanged. The experimental results, presented in Tables 5 and 6, illustrate the performance of the algorithms under these varying communication ranges, with the evaluation based on NRMSE values. Lower NRMSE values indicate higher accuracy in node localization. Generally, an increase in the communication radius corresponds to a decrease in NRMSE values, signifying improved accuracy. It is noteworthy that indiscriminate expansion of the communication range may not always be the optimal strategy. While a larger communication range enhances the accuracy of hop counts between sensor nodes, leading to better connectivity, the optimal approach is contingent on the specific requirements of the WSN. A larger communication range provides sensor nodes with more neighbors, facilitating more accurate hop counts and enhanced connectivity, enabling each unknown node to leverage a greater number of anchors for improved self-localization.

Table 5. Result of different values of communication range for Dv-hop (span = 1).

R		15 m	20 m	25 m	30 m
NRMSE of	Dv-hop	139.89%	72.56%	56.34%	43.22%
	Dv-hop (5 V)	118.45%	45.45%	39.45%	35.58%
	Dv-hop (10 V)	115.67%	42.34%	37.87%	34.35%
	Dv-hop (15 V)	98.87%	40.65%	36.23%	34.17%
	Dv-hop (20 V)	65.82%	39.54%	35.05%	34.11%
	Dv-hop (25 V)	67.72%	38.67%	35.12%	34.02%

Table 6. Result of different values of communication range for DNNs (span = 12).

R		15 m	20 m	25 m	30 m
NRMSE of	Dv-hop + DNN	99.02%	52.55%	45.53%	34.05%
	Dv-hop + DNN (5 V)	90.12%	34.87%	29.34%	25.34%
	Dv-hop + DNN (10 V)	88.98%	32.56%	27.43%	24.73%
	Dv-hop + DNN (15 V)	65.44%	30.33%	26.82%	24.34%
	Dv-hop + DNN (20 V)	47.34%	24.36%	24.25%	23.49%
	Dv-hop + DNN (25 V)	48.54%	25.5%	23.56%	22.60%

5.2. Performance Analysis

This section explores the advantages of the proposed Dv-hop + DDNs correction compared to other state-of-the-art optimization algorithms, including those employed by EFPA-G [24] and WRCDv-hop [25]. Many contemporary optimization algorithms, such as those utilized by EFPA-G and WRCDv-hop, leverage intelligent optimizers to estimate the locations of unknown nodes. The complexity of these algorithms depends on various factors, including the number of real anchors and communication range, leading to increased accuracy at the expense of higher energy consumption. However, the use of many

real anchors can be cost-prohibitive and may pose challenges in terms of training data size. To address this issue, we introduced a DAS that virtually increases the number of anchors, mitigating the prohibitive cost associated with the deployment of many real anchors.

6. Conclusions

This article introduced an innovative and precise machine learning-based approach for range-free localization in WSN applications within the IoT. Our methodology presents a cost-effective distance estimation strategy through the development of DNN. The aim is to reduce localization errors and enhance accuracy without incurring additional hardware costs. Additionally, we proposed a DAS that virtually increases the number of anchors, significantly augmenting the training data and leading to more accurate localization. Simulation results illustrate the effectiveness of our DAS in range-free localization for WSNs, particularly with a limited number of real anchors. Notably, our proposed Dv-hop + DNNs correction surpasses the traditional Dv-hop algorithm, demonstrating superior localization accuracy.

Author Contributions: Methodology, J.E. and S.A.; software, J.E.; validation, J.E.; formal analysis, J.E.; data curation, J.E. and S.A.; writing—original draft preparation, J.E.; writing—review and editing, J.E.; supervision, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: Work supported in part by the Discovery Grants Program of NSERC (Natural Sciences and Engineering Research Council of Canada).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

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

References

1. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Niyato, D.; Dobre, O.; Poor, H.V. 6G Internet of things: A comprehensive survey. *IEEE Internet Things J.* **2022**, *9*, 359–383. [\[CrossRef\]](#)
2. Paul, A.K.; Sato, T. Localization in wireless sensor networks: A survey on algorithms, measurement techniques, applications, and challenges. *J. Sens. Actuator Netw.* **2017**, *6*, 24. [\[CrossRef\]](#)
3. Bianchi, V.; Ciampolini, P.; Munari, I.D. RSSI-based indoor localization and identification for Zigbee wireless sensor networks in smart homes. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 566–575. [\[CrossRef\]](#)
4. Sundhari, R.M.; Jaikumar, K. IoT assisted hierarchical computation strategic making (HCSM) and dynamic stochastic optimization technique (DSOT) for energy optimization in wireless sensor networks for smart city monitoring. *Comput. Commun.* **2020**, *150*, 226–234. [\[CrossRef\]](#)
5. Niculescu, D.; Nath, B. Ad hoc positioning system (APS) using AOA. In Proceedings of the IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428), San Francisco, CA, USA, 30 March–3 April 2003; Volume 3, pp. 1734–1743.
6. Kumar, P.; Reddy, L.; Varma, S. Distance measurement and error estimation scheme for RSSI based localization in wireless sensor networks. In Proceedings of the 2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN), Allahabad, India, 15–19 December 2009; pp. 1–4.
7. Voltz, P.J.; Hernandez, D. Maximum likelihood time of arrival estimation for real-time physical location tracking of 802.11 a/g mobile stations in indoor environments. In Proceedings of the Position Location and Navigation Symposium, Allahabad, India, 15–19 December 2004; pp. 585–591.
8. Wang, Y.; Wang, X.; Wang, D.; Agrawal, D.P. Range-free localization using expected hop progress in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2009**, *25*, 1540–1552. [\[CrossRef\]](#)
9. Boukerche, A.; Oliveira, H.A.B.F.; Nakamura, E.F.; Loureiro, A.A.F. Localization systems for wireless sensor networks. *IEEE Wirel. Commun.* **2007**, *14*, 6–12. [\[CrossRef\]](#)
10. Niculescu, D.; Nath, B. Ad hoc positioning system (APS). In Proceedings of the GLOBECOM'01, IEEE Global Telecommunications Conference (Cat. No.01CH37270), San Antonio, TX, USA, 25–29 November 2001.
11. Abu Alsheikh, M.; Lin, S.; Niyato, D.; Tan, H.-P. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1996–2018. [\[CrossRef\]](#)
12. Bhatti, G. Machine learning based localization in large-scale wireless sensor networks. *Sensors* **2018**, *18*, 4179. [\[CrossRef\]](#) [\[PubMed\]](#)

13. Chen, A.C.H.; Jia, W.K.; Hwang, F.J.; Liu, G.; Song, F.; Pu, L. Machine learning and deep learning methods for wireless network applications. *EURASIP J. Wirel. Commun. Netw.* **2022**, *2022*, 115. [[CrossRef](#)]
14. Zainab, M.; Sadik, K.G.; Ammar, H.M.; Ali, A.J. Neural network-based Alzheimer's patient localization for wireless sensor network in an indoor environment. *IEEE Access* **2020**, *8*, 150527–150538.
15. Baird, H.S. *Document Image Analysis. Chapter Document Image Defect Models*; IEEE Computer Society Press: Los Alamitos, CA, USA, 1995; pp. 315–325.
16. Lin, C.; Weidong, S. Virtual big data for GAN based data augmentation. In Proceedings of the 2019 IEEE International Conference on Big Data, Los Angeles, CA, USA, 9–12 December 2019.
17. Zhang, B.; Lin, J.; Du, L.; Zhang, L. Harnessing data augmentation and normalization preprocessing to improve the performance of chemical reaction predictions of data-driven mode. *Polymers* **2023**, *15*, 2224. [[CrossRef](#)]
18. Wu, X.; Zhang, Y.; Yu, J.; Chang, C.; Qiao, H.; Wu, Y.; Wang, X.; Eu, Z.; Duan, H. Virtual data augmentation method for reaction prediction. *Sci. Rep.* **2022**, *12*, 17098. [[CrossRef](#)] [[PubMed](#)]
19. Paul, A.; Prasad, A.; Kumar, A. Review on artificial neural network and its application in the field of engineering. *J. Mech. Eng. PRAKASH* **2022**, *1*, 53–61. [[CrossRef](#)]
20. Amirsadri, S.; Mousavirad, S.J.; Ebrahimpour-Komleh, H. A Levy_fight-based grey wolf optimizer combined with back-propagation algorithm for neural network training. *Neural Comput. Appl.* **2018**, *30*, 3707–3720. [[CrossRef](#)]
21. Yang, C.; Kim, H.; Adhikari, S.P.; Chua, L.O. A Circuit-based neural network with hybrid learning of backpropagation and random weight change algorithms. *Sensors* **2017**, *17*, 16. [[CrossRef](#)] [[PubMed](#)]
22. Tarigan, J.; Diedan, R.; Suryana, Y. Plate recognition using backpropagation neural network and genetic algorithm. *Procedia Comput. Sci.* **2017**, *116*, 365–372. [[CrossRef](#)]
23. Alaa, A.H.; Bekir, K.; Mohammad, S.S. Back-propagation algorithm with variable adaptive momentum. *Knowl. Based Syst.* **2016**, *114*, 79–87.
24. Jun, Z.; Ting, Y.; Wenwu, X.; Zhihe, Y.; Dan, Y. An enhanced flower pollination algorithm with gaussian perturbation for node location of a WSN. *Sensors* **2023**, *23*, 6463.
25. Cui, H.; Wang, S.; Zhou, C. A high-accuracy and low-energy range-free localization algorithm for wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.* **2023**, *2023*, 37. [[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.