electronics

Article

Performance of Differential Evolution Algorithms for Indoor Area Positioning in Wireless Sensor Networks

Shu-Hung Lee [1], Chia-Hsin Cheng [2,*], Kuan-Hsien Lu [2], Yeong-Long Shiue [2] and Yung-Fa Huang [3,*]

[1] School of Intelligent Manufacturing and Automotive Engineering, Guangdong Business and Technology University, Zhaoqing 526020, China; umaillee@gmail.com
[2] Department of Electrical Engineering, National Formosa University, Huwei 632301, Taiwan; nfumsee@gmail.com (K.-H.L.); syl@nfu.edu.tw (Y.-L.S.)
[3] Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 413310, Taiwan
* Correspondence: chcheng@nfu.edu.tw (C.-H.C.); yfahuang@cyut.edu.tw (Y.-F.H.); Tel.: +886-4-2332-3000 (Y.-F.H.)

Abstract: In positioning systems in wireless sensor networks, the accuracy of localization is often affected by signal distortion or attenuation caused by environmental factors, especially in indoor environments. Although using a combination of K-Nearest Neighbor (KNN) algorithm and fingerprinting matching can reduce positioning errors due to poor signal quality, the improvement in accuracy by increasing the number of reference points and K values is not significant. This paper proposes a Differential Evolution-based KNN (DE-KNN) method to overcome the performance limitations of the KNN algorithm and enhance indoor area positioning accuracy in WSNs. The DE-KNN method aims to improve the accuracy and stability of indoor positioning in wireless sensor networks. According to the simulation results, in a simple indoor environment with four reference points, when the sensors are deployed in both fixed and random arrangements, the positioning accuracy was improved by 29.09% and 30.20%, respectively, compared to using the KNN algorithm alone. In a complex indoor environment with four reference points, the positioning accuracy was increased by 32.24% and 33.72%, respectively. When the number of reference points increased to five, in a simple environment, the accuracy improvement for both fixed and random deployment was 20.70% and 26.01%, respectively. In a complex environment, the accuracy improvement was 23.88% and 27.99% for fixed and random deployment, respectively.

Keywords: indoor positioning system; wireless sensor network; K-Nearest neighbor algorithm; differential evolution algorithm; DE-KNN

check for updates

Citation: Lee, S.-H.; Cheng, C.-H.; Lu, K.-H.; Shiue, Y.-L.; Huang, Y.-F. Performance of Differential Evolution Algorithms for Indoor Area Positioning in Wireless Sensor Networks. *Electronics* **2024**, *13*, 705. https://doi.org/10.3390/electronics13040705

Academic Editors: Syed Muhammad Raza and Duc Tai Le

Received: 22 December 2023
Revised: 5 February 2024
Accepted: 7 February 2024
Published: 9 February 2024

1. Introduction

The Global Positioning System (GPS) is currently the most common used method in outdoor positioning systems and plays a crucial role in such systems. However, GPS utilizes line-of-sight transmission and is hindered by obstacles in indoor environments, rendering it ineffective for indoor applications [1–3]. In contrast to outdoor positioning methods, indoor positioning systems achieve higher accuracy by leveraging the geographical information within buildings. Within indoor positioning systems, two main types can be classified: area-based positioning and precise positioning. Area-based positioning divides the indoor environment into multiple zones without the need for explicit boundary or shape specification, computing the approximate position of the target within the designated zone. Conversely, precise positioning places more emphasis on accurately determining the target's position, striving to minimize errors. The better the accurate error of the precise positioning method approaches zero, the better its performance.

In recent years, the Internet of Things (IoT) has garnered extensive attention and recognition from the industry due to advancements in science and technology [4–6]. The essential aspect of the Internet of Things (IoT) is to facilitate communication between objects in order to create interconnected networks, surpassing traditional human-to-human communication. It consists of an application layer, network layer, and perception layer, where wireless sensor networks play a crucial role in the network and perception layers, forming part of the IoT infrastructure. Wireless sensor networks construct a network by connecting independent sensor nodes, finding broad applications in home monitoring, military applications, smart cities, environmental monitoring, healthcare fields, and indoor positioning, among other applications [7–13]. Nevertheless, in such applications, sensor nodes are usually placed in fixed positions and lack mobility. However, when sensor nodes are deployed on moving objects, ensuring precise location information of the target becomes crucial. As a result, there is a significant focus on exploring and researching effective positioning techniques for accurately obtaining target position information.

Various novel techniques have been proposed to address the localization issue in wireless sensor networks lately. These techniques include machine learning-based approaches such as the DFL framework introduced by [14], which utilizes multiple convolutional neural network (CNN) layers and a deep belief network (DBN) based on restricted Boltzmann machines (RBM) for feature recognition and extraction, forming a convolutional deep belief network (CDBN). Even in scenarios with reduced data dimensions and low signal-to-noise ratios (SNRs), this method achieves an accuracy of up to 98%. In addition, pattern recognition-based localization methods have also been developed, such as the fingerprint-based indoor 2D localization method that combines received signal strength indicator (RSSI) and magnetometer measurements for locating mobile robots [15]. By incorporating the measurement information of magnetic field strength between the mobile unit and anchor nodes, the method achieves over 35% improvement when compared to results obtained by utilizing only RSSI or magnetic sensor data, particularly in scenarios with high magnetic field variance. Similarly, a cost-effective sensor architecture-based efficient wireless sensor network fingerprint localization system has been established using an indoor fingerprint dataset and four custom machine learning models [16]. Additionally, one study [17] achieved a lower localization error under the same anchor node ratio and wireless sensor network configuration by utilizing the DEEC-Gauss gradient distance algorithm. The DEEC-Gauss gradient distance algorithm demonstrates lower positioning error than five other algorithms including weighted centroid localization algorithm under the same conditions, but its accuracy may be limited in scenarios of uneven node density. Moreover, a pattern matching method based on Spline curves has been proposed to verify the location of sensor nodes, offering low power consumption and processing time without requiring any additional hardware. Multiple cubic Bezier curves are generated using the received signal strength indicator (RSSI) values from four nodes at a time, enabling efficient detection of changes in node position and the magnitude of such changes. The algorithm is implemented in the Cooja simulator, achieving a location verification accuracy of up to 90%. A parameter called the RSSI Range Factor (RRF) is introduced to estimate the degree of location change, with an accuracy of up to 99% [18].

Furthermore, optimization-based localization learning algorithms are also introduced. For instance, Fute, E.T. et al. [19] improves the performance of particle swarm optimization (PSO) by utilizing an approach called FPSOTS, where each particle uses a tabu search form to determine its best local neighbor and increase the convergence possibilities towards better solutions. On the other hand, they introduced restrictions and performance checks, allowing only better particles from the search space constructed by constraint analysis to evolve around the initial solution obtained through trilateration. This approach achieved fast convergence and better accuracy in locating unknown nodes in WSN compared to recent techniques such as HPSOVNS, NS-IPSO, ECS-NL, and GTOA, but it also increased computational complexity relative to PSO. Mehrabian, H. et al. [20] combined RSSI with pedestrian dead reckoning (PDR) algorithms. They introduced a novel Weight-Based Optimization (WBO) filter to optimize RSSI data, along with the measurement data from accelerometer and compass sensors, and utilizes sensor fusion techniques to achieve a positioning accuracy of up to 68 cm. In addition, a localization scheme combining artificial fish swarm algorithm (AFSA) with a region segmentation method (RSM), hybrid adaptive visual pursuit (HAVP) method, and dynamic AF selection (DAFS) method is proposed in [21], in which the total average positioning error was reduced by 96.1%, and the positioning time was shortened by 26.4% using the HAVP for the target positioning. Reference [22] introduces an algorithm that ensures robustness against environmental irregularities for localizing sensor nodes within regions delineated by anchor node networks, with the objective of achieving higher precision at the lower boundary, while also offering an analytical framework for sensor localization. Shilpi and Kumar, A. proposed that the method improves localization accuracy in a variety of isotropic, O-shaped anisotropic, and S-shaped anisotropic wireless sensor networks, thereby reducing the influence of various anisotropy factors by utilizing the nonparametric Jaya algorithm (JA) and range-aware reliable anchor pairs (RAP) selection method, which provides better localization accuracy compared to four existing node localization methods, including Distance Vector (DV)-maxHop [23]. An iterative bounding box algorithm enhanced by a Kalman filter, which effectively reduces localization error without the need for additional equipment or increased communication costs, was presented in [24]. Agarwal, N. et al. proposed a node localization scheme based on the Aquila optimization algorithm (IAOAB-NLS) simulates fish and bat behaviors. This approach exhibits the ability to accurately determine the coordinates of nodes within a network. Experimental results consistently confirm the effectiveness of the IAOAB-NLS model, irrespective of fluctuations in network parameters [25]. Additionally, an optimization-based localization learning algorithm (OLLA) is proposed by [26], which demonstrates good performance in indoor and outdoor scenarios.

In the realm of wireless sensor network localization algorithms, other approaches have also been proposed, for instance, the DV-Hop algorithm based on rotation group methods and weighted normalization, also known as CMWN-DV-Hop. This algorithm exhibits excellent performance in reducing localization errors, and because of its low sensitivity to the number of nodes, the CMWN-DV-Hop (a = 10) algorithm can be applied to large-scale wireless sensor networks [27]. Moreover, Fawad, M. et al. proposed an improved DV-Hop algorithm, named the Hop-correction and energy-efficient DV-Hop (HCEDV-Hop) algorithm, to achieve efficient and accurate localization while reducing energy consumption [28]. To obtain accurate localization results, it is necessary to manage the topology structure within the DV-Hop algorithm effectively. Topology management encompasses strategies related to node deployment, connectivity, and route selection, which can influence the communication quality and transmission performance among nodes [29,30]. By carefully designing the network topology, interference and hop count between nodes can be minimized, thereby improving the stability and accuracy of data transmission, and enhancing the localization accuracy of the DV-Hop algorithm. Additionally, for indoor area position estimation problems, a method based on distributed GNSS sensors and area azimuth sensors is proposed. By establishing a positional deviation (PD) vector compensation and sequential fusion (SF) framework, this method estimates the PD vector recursively for accurate positioning derived from GNSS sensors and area azimuth sensors [31].

In sensor networks, various measurement techniques can be utilized to achieve the localization of unidentified transmitters, including the angle of arrival (AOA), time of arrival (TOA), time difference of arrival (TDOA), or RSSI [32–35]. Among these options, RSSI has an advantage due to its cost-effectiveness and implementation simplicity, as it does not require additional hardware expenses. However, RSSI is susceptible to environmental variations and may result in positioning errors [36–39]. Compared to outdoor positioning, indoor positioning systems require higher accuracy, making indoor localization more challenging than outdoor localization.

Due to the need for increased accuracy in indoor positioning systems, there have been investigations into incorporating artificial intelligence algorithms in order to enhance the precision of indoor area-based positioning. Reference [40] proposed the INTRI indoor positioning method, combining fingerprint matching and trilateration to enhance indoor positioning accuracy. References [41–45] utilized adjusted K-nearest neighbor (KNN) parameters to improve indoor positioning precision. Xie, Y. et al. introduced an indoor positioning method using KNN with Spearman distance [46]. References [47–50] utilized Differential Evolution (DE) to enhance positioning algorithm performance.

The motivation behind this research is to address the challenges in indoor positioning systems and improve the accuracy of area-based positioning within wireless sensor networks. Our contribution lies in examining the constraints and suggesting enhancements for utilizing the combination of the KNN algorithm and the RSSI channel model to enhance the precision of area-based positioning. Additionally, we propose the implementation of the KNN algorithm in combination with Differential Evolution (DE) for indoor positioning to further improve the accuracy of area-based positioning within wireless sensor networks. The method proposed in this study is to enhance the accuracy of indoor area localization. Similar to other indoor positioning techniques, it can be applied to various fields that require indoor positioning technology, including indoor navigation, indoor mapping, emergency management, smart buildings, logistics and warehousing, healthcare, and industrial monitoring. In general, indoor positioning technology plays a significant role in modern society, offering convenience and benefits to people's daily lives and work.

The rest of this paper is outlined as follows. Section 2 provides an overview of the RSSI channel model, the KNN algorithm, and the differential evolution (DE) algorithm. Section 3 explains the methodology employed in this paper, where the KNN algorithm is used for target point area-based positioning, and the DE-KNN algorithm is incorporated to enhance the localization precision. In Section 4, a comparative analysis of the area positioning simulations using KNN and DE-KNN algorithms is presented. Finally, Section 5 concludes the paper.

2. Related Technologies

2.1. RSSI Channel Model

The RSSI channel model, referred to as the propagation path loss model in the literature [51–53], can be categorized into two types: the free space propagation model and the log-distance path loss model. These models differ in how they estimate the signal strength received by the receiver in various scenarios. A free space propagation model, such as the Friis free space model, is employed when there are no obstructions or line of sight (LOS) between the transmitter and receiver. In this model, the received signal strength at the receiver is inversely proportional to the square of the distance. On the other hand, the logarithmic distance path loss model is used to estimate the average power of the received signal in different environments. Through theoretical derivations and research studies, it has been observed that the average power of the received signal undergoes exponential attenuation as the distance increases.

The power loss induced by the propagation path is a stochastic variable when the transmitter and receiver are at a constant distance from each other. This variability arises due to changes in the interference encountered by the signal in natural environments. Hence, in simulation analysis, we can only capture this phenomenon by employing a logarithmic normal distribution. As a result, the received power can be expressed as

$$P_r(d)[\mathrm{dBm}] = P_t[\mathrm{dBm}] - PL(d)[\mathrm{dB}], \tag{1}$$

and the path loss can be represented as

$$PL(d)[\mathrm{dB}] = PL(d_0) + 10n \cdot log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \tag{2}$$

where X_σ is a zero-mean Gaussian random variable with the standard deviation σ. n represents the Path Loss Exponent, which indicates the rate of path loss; d_0 is the Close-in Distance, representing the distance very close to the transmitting end; d represents the distance between the transmitting end and the receiving end.

In indoor environments, signals may propagate through multiple paths before reaching the receiver. When signals from different paths arrive at the receiver with different time delays and superimpose on each other, it can cause interference and signal distortion. Additionally, indoor environments contain various objects such as walls, furniture, and human bodies, which can partially or completely obstruct the propagation path of the signal, resulting in signal attenuation. Moreover, surfaces like walls, ceilings, and floors can reflect signals, altering the direction and path of signal propagation. Furthermore, refraction between different media can lead to signal loss and changes in the propagation path. In addition, signals in indoor environments encounter free space path loss, power attenuation, molecular scattering, and other propagation effects, causing a decrease in signal strength [54–56]. These are potential environmental factors that can cause signal distortion or attenuation indoors. Signal distortion and attenuation directly affect RSSI in indoor environments, thereby impacting positioning accuracy.

2.2. Fingerprint Matching

Fingerprint matching, also known as pattern matching [44], is divided into two stages: the training phase and the template matching phase. The training phase involves collecting a large amount of known sample data from the environment and recording it in a sample database. The template matching phase involves comparing the unknown data with the samples in the sample database, finding the most similar data, and displaying the result, as shown in Figure 1.

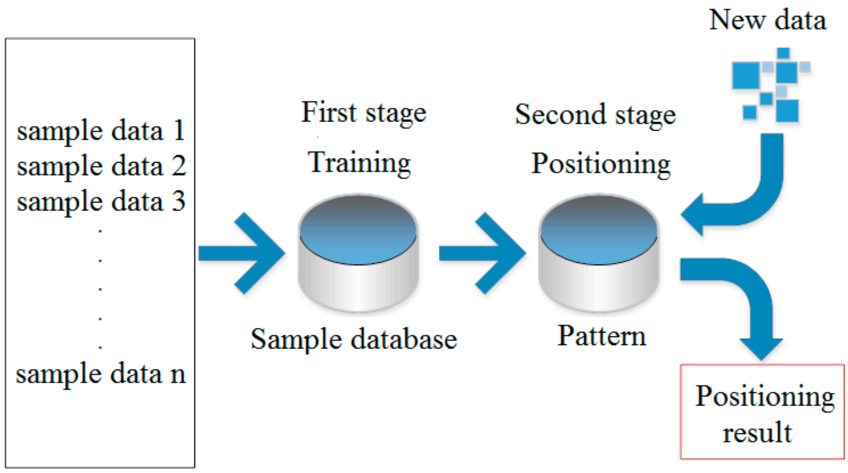

Figure 1. Fingerprint matching diagram.

2.3. K Nearest Neighbor Algorithm

The Nearest Neighbor in Signal Strength (NNSS) algorithm [5,40–43] is a classification algorithm that searches for the most similar data in the sample database to an unknown data point. It assigns a class label based on the closest match in the database. In general, the NNSS algorithm uses Euclidean distance as the measure of similarity, as shown by

$$\|\boldsymbol{X}-\boldsymbol{Y}\| = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\ldots\ldots+(x_n-y_n)^2} \tag{3}$$

where $\boldsymbol{X} = (x_1, x_2, x_3, \ldots, x_n)$ and $\boldsymbol{Y} = (y_1, y_2, y_3, \ldots, y_n)$ represent two coordinate points in n-dimension space.

The KNN algorithm is an algorithm where K is a constant. KNN utilizes the nearest neighbor algorithm to find the K most similar data instances, and it determines the class label of an unknown data instance based on majority voting. In Figure 2, for example, when K = 4, the unknown data instance would be classified as belonging to Class A.

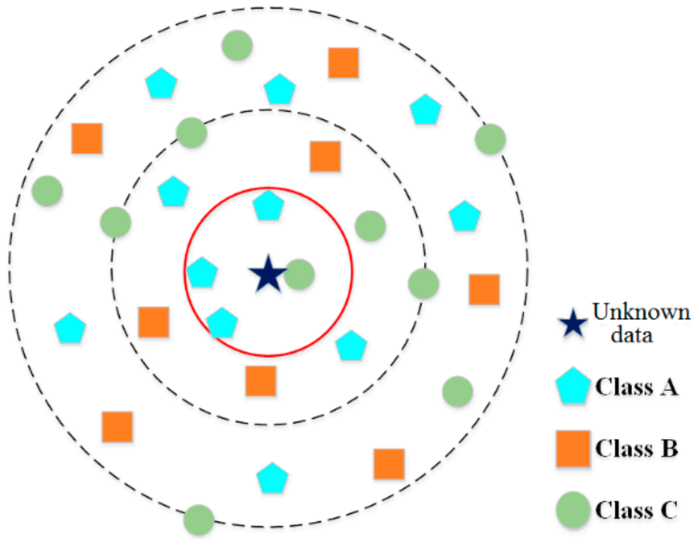

Figure 2. KNN algorithm classification illustration.

2.4. Differential Evolution Algorithm

The Differential Evolution algorithm (DE), also known as Differential Evolution or Differential Evolutionary Algorithm, was first proposed in 1995 by Storn and Price for solving the problem of Chebyshev polynomial fitting. It was formally presented as a commonly used differential evolution algorithm at the first IEEE International Conference on Evolutionary Computation in 1996 [47–50]. DE is based on the diversity among individuals in a population, introducing the differences between individuals to others and observing whether this difference can bring positive effects in order to obtain evolutionary advantages.

DE algorithm, Particle Swarm Optimization (PSO) [57,58], Genetic Algorithm (GA) [59], and other bio-inspired evolutionary algorithms share the same challenge of easily getting trapped in local optima. The reason behind this lies in the fact that most bio-inspired evolutionary algorithms are similar as brute force search algorithms, conducting limited iterations within a population to find the optimal solution. As the number of iterations increases, these algorithms tend to converge globally. Therefore, both individual diversity and the number of iterations can impact the ability of the algorithm to search for the global optimum.

2.4.1. Initialization

In the DE algorithm, the population is initialized randomly. It involves generating NP sets of variable dimension d vectors based on the user-defined population size (*NP*) and dimension (*D*), as shown by

$$\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iD}], i = 1, 2, 3, \ldots, NP \tag{4}$$

and

$$\boldsymbol{x}_{ij} = a_j + rand \times \left(\mathrm{b}_j - \mathrm{a}_j\right)\ i = 1, 2, \ldots, NP,\ j = 1, 2, \ldots, D \tag{5}$$

respectively, where a_j and b_j are the upper and lower bounds of x_{ij}, and *rand* ranges from 0 to 1.

2.4.2. Fitness

Fitness is a crucial parameter in the Differential Evolution algorithm, and its calculation method varies depending on the specific system design. In this paper, the fitness value is calculated using the Euclidean distance measure.

2.4.3. Mutation

There are six common mutation methods in the Differential Evolution algorithm [60]. The concept of mutation involves randomly selecting several variable vectors from the population (*NP*) and using a mutation weighting factor (*F*) to perform operations on these vectors. This results in the generation of a composite vector (Donor Vector) $V_{i,g}$, also known as the mutation vector. In the literature [60], it is mentioned that there are six ways to perform DE mutation operations, as shown in Equations (6)–(11).

- DE/rand/1:

$$V_{i,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}) \tag{6}$$

- DE/best/1:

$$V_{i,g} = X_{best,g} + F \cdot (X_{r1,g} - X_{r2,g}) \tag{7}$$

- DE/rand-to-best/1:

$$V_{i,g} = X_{i,g} + F \cdot \left(X_{best,g} - X_{i,g}\right) + F \cdot (X_{r1,g} - X_{r2,g}) \tag{8}$$

- DE/best/2:

$$V_{i,g} = X_{best,g} + F \cdot (X_{r1,g} - X_{r2,g}) + F \cdot (X_{r3,g} - X_{r4,g}) \tag{9}$$

- DE/rand/2:

$$V_{i,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}) + F \cdot (X_{r4,g} - X_{r5,g}) \tag{10}$$

- DE/rand-to-best/2:

$$V_{i,g} = X_{i,g} + F \cdot (X_{best,g} - X_{i,g}) + F \cdot (X_{r1,g} - X_{r2,g}) + F \cdot (X_{r3,g} - X_{r4,g}) \tag{11}$$

In Equations (5)–(10), $(r1 \neq r2 \neq r3 \neq r4 \neq r5 \neq i) \in$ range [1, *NP*], where $r1$ to $r5$ are randomly determined and unique values, and *NP* represents the population size. $X_{best,g}$ represents the variable vector with the best fitness value in the population during the g-th iteration, where g denotes the g-th iteration. In this paper, Equation (5) is used to calculate the mutation vector.

2.4.4. Crossover

In the DE algorithm, crossover can be divided into two types: binomial crossover and exponential crossover [60], which generate the trial vector $\boldsymbol{u}_{i,g}$. Before introducing these two crossover methods, it is necessary to define the crossover rate (CR), which ranges from 0 to 1. The crossover rate determines the probability of each component in the trial vector being inherited from either the target vector or the trial vector.

Binomial crossover is the most common used crossover method in the algorithm. In the g-th iteration, the j-th component of the trial vector $\boldsymbol{u}_{i,g}$ is selected from either the target vector $\boldsymbol{x}_{j,i,g}$ or the donor vector $\boldsymbol{v}_{j,i,g}$, as shown by

$$\mathbf{u}_{i,g} = \mathbf{u}_{j,i,g} = \begin{cases} \mathbf{v}_{j,i,g} & \text{if } (rand_j(0,1) \leq C_r) \\ \mathbf{x}_{j,i,g} & \text{otherwise} \end{cases} \tag{12}$$

Exponential crossover: in the case of exponential crossover, two integers N and L are chosen randomly within the range [1, *NP*]. N represents the starting point of crossover for the target vector $\boldsymbol{x}_{j,i,g}$ and donor vector $\boldsymbol{v}_{j,i,g}$, whereas L indicates the stopping point of crossover for the components, as shown by

$$\mathbf{u}_{i,g} = \mathbf{u}_{j,i,g} = \begin{cases} \mathbf{v}_{j,i,g} & \text{if } j = \|N\|, \|N+1\|, \ldots, \|N+L-1\| \\ \mathbf{x}_{j,i,g} & \text{otherwise j} \in [1, D] \end{cases} \tag{13}$$

where $\|\cdot\|$ represents the distance between N and D.

2.4.5. Selection

Selection is a concept based on the survival of the fittest, which primarily uses the fitness function as a criterion. It calculates the fitness value of the trial vector $\boldsymbol{u}_{i,g}$ or the target vector $\boldsymbol{x}_{i,g}$, compares the fitness values of the trial vector and the target vector, and keeps the one with the better fitness value as the target vector for the next iteration, denoted as $\boldsymbol{x}_{i,g+1}$, as shown by

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & \text{if } f(\mathbf{u}_{i,g}) \geq f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases} \tag{14}$$

2.4.6. Termination

In evolutionary algorithms, the most common used termination condition is the maximum number of iterations, denoted as G. When the iteration reaches this maximum value, the algorithm stops evolving. Additionally, there are two other termination conditions: fitness threshold and improvement rate threshold. The fitness threshold specifies that the algorithm stops evolving when the fitness value of the obtained solution reaches a certain threshold. The improvement rate threshold checks whether the fitness value improves significantly in each iteration, and when the fitness value converges, the algorithm stops evolving.

2.5. The Process of the Differential Evolution Algorithm

According to the introduction of the Differential Evolution algorithm mentioned above, the algorithm steps are shown as follows. Finally, according to the sequence of initializing the population, evaluating the fitness, mutation, mating, selection, and termination criteria, we can obtain the flowchart of the Differential Evolution algorithm, as shown in Figure 3.

Step 1. Randomly initialize the population.
Step 2. Evaluate the fitness values.
Step 3. Mutation operation: Generate trial vectors by performing mutation on the target vectors.
Step 4. Crossover operation: Mate the trial vectors with the target vectors to produce offspring vectors.
Step 5. Evaluate the fitness values of both target and trial vectors, and select the ones with better fitness values as the target vectors for the next iteration.
Step 6. Repeat Steps 3 to 5 until the termination condition is met. Output the best solution when the termination condition is satisfied.

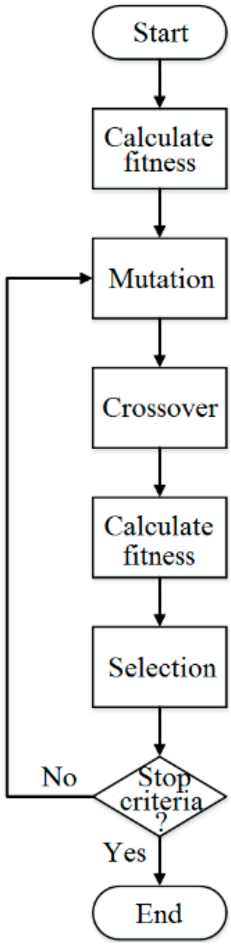

Figure 3. Flowchart of differential evolution algorithm.

3. Area Positioning Methods

In this section, the methods of target positioning in wireless sensor networks based on the DE algorithm and KNN are described in detail. The RSSI channel model described in Section 2 is used to estimate the distance between the target point and the algorithm individual, and the highest point of the food source (RSSI value) is taken as the estimate point by the individual moving several times. In addition, this paper also discusses the impact of the number of sensors and sensors deployment on the system.

3.1. KNN Area Positioning Method

Based on the fingerprint matching method described in Section 2, each training point is represented by a feature vector in the localization model, as shown by

$$\mathbf{E}_i = \left\langle e_1^i, e_2^i, e_3^i \ldots, e_j^i \right\rangle \tag{15}$$

where e_j^i represents the RSSI value received at training point i from reference point j, similarly, a feature vector is created for the localization target point T, as shown by

$$\mathbf{T} = \langle t_1, t_2, t_3 \ldots, t_j \rangle \tag{16}$$

where t_j represents the RSSI value received from reference point j. Then, by calculating the geometric distance expressed by

$$\|\mathbf{T}, \mathbf{E}_i\| = \sqrt{\sum_{j=1}^{n} \left(t_j - e_j^i\right)^2} \tag{17}$$

with the feature vectors from Equations (15) and (16), the K-Nearest Neighbor average algorithm is applied to estimate the position of the target point. The K-Nearest Neighbor average algorithm uses a constant value for K. When K is set to 4, it means that the positioning is based on the nearest 4 training point positions. After representing these four training point positions in coordinates, the x and y coordinates are averaged, as shown in Equation (18). The estimated position is then determined based on the average result. As depicted in Figure 4, the target point is potentially located in the L5 region, expressed by

$$(x, y) = \left(\frac{\sum_{n=1}^{K} x_n}{K}, \frac{\sum_{n=1}^{K} y_n}{K} \right) \tag{18}$$

where x_n and y_n represent the coordinate values of n-th point.

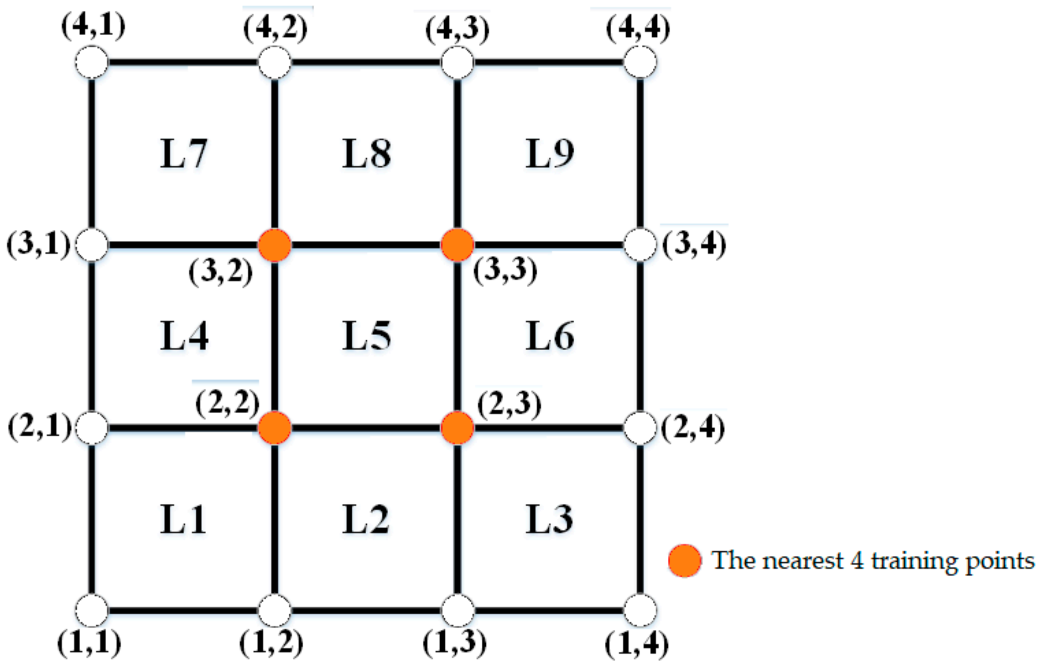

Figure 4. Illustration of K-nearest neighbor average algorithm localization.

3.2. Target Positioning

To enhance the realism of this research, target point localization is divided into two categories: random-target-point deployment and fixed-target-point deployment. Random deployment refers to the random placement of target points to be positioned within a single area. In this approach, 100 target points are generated randomly within the designated area for a total of 100 positioning trials. On the other hand, fixed deployment involves the fixed placement of target points within the same area. This method utilizes four predetermined points, with each point being positioned 25 times to yield a total of 100 positioning trials, illustrated in Figure 5a,b, respectively. Random-target-point localization entails randomly distributing the target points within a given area to examine their localization accuracy. Conversely, fixed-target-point localization involves positioning four fixed points, each subjected to 25 localization attempts, resulting in a cumulative total of 100 localization instances. These categories are employed to ascertain the efficacy of the localization algorithm under varying deployment scenarios.

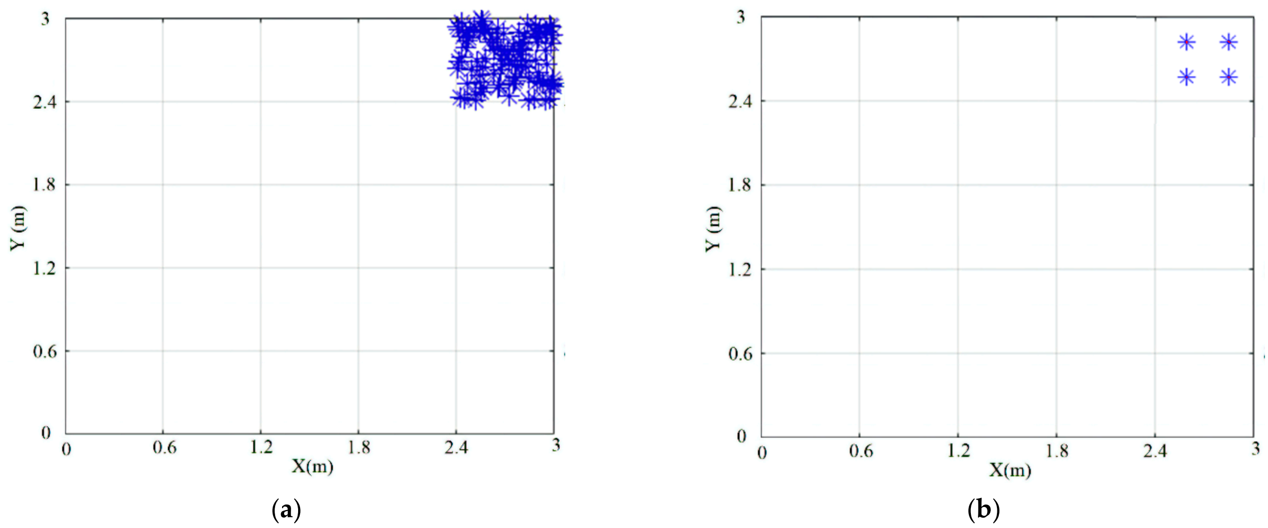

Figure 5. Illustration of (**a**) random deployment and (**b**) fixed deployment.

3.3. DE-KNN Area Positioning Method

This section presents our proposed approach for target localization combining KNN with DE. Based on the DE flowchart, as shown in Figure 3, the flowchart of the target positioning method combining KNN and DE (DE-KNN) is shown in Figure 6.

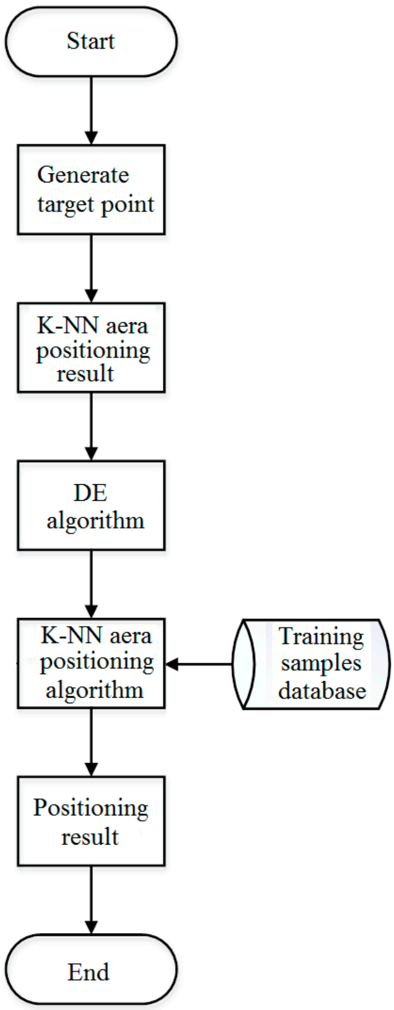

Figure 6. The procedure of DE-KNN area positioning method.

The processes begin with the initialization of the population. A total of *NP* variable vectors is generated according to either fixed or random deployment, with the size determined by the number of reference points. After the generation process, KNN is used to find N nearest neighbors with accurately located RSSI values in the population, and their averages of RSSI values are taken as the best target vector for calculating the fitness value.

The fitness value is calculation involves computing the geometric distance between the best target vector found in the initialized population and each target point. The fitness value's magnitude is used to assess the similarity between the target point and the best target point; a smaller value indicates higher similarity, whereas a larger value indicates lower similarity.

In the crossover step, we employ binomial crossover with a preset crossover rate (CR) of 0.9. In both the target vector and the mutant vector, each element is compared with a random number between 0 and 1. If the random number is less than the crossover rate (CR), the corresponding element from the mutant vector is selected; otherwise, the element from the target vector is chosen. This process generates a trial vector, whose fitness value is then computed. If the fitness value of the trial vector is lower than that of the original target vector, it replaces the original target vector as the target vector for the next iteration. Otherwise, the original target vector remains unchanged. The algorithm terminates after 100 iterations, at which point the results are outputted.

After obtaining the results from DE, we perform another round of local positioning using KNN and analyze the simulated experimental results.

4. Simulation Results and Discussion

4.1. Simulation Environment and Settings

This study focuses on evaluating and comparing the performance of the KNN and DE-KNN algorithms in indoor area positioning methods using a wireless sensor network. The network covers a 2D area measuring 6 m × 6 m, which is divided into 25 small regions. Each of these regions undergoes 100 simulated positioning instances, and the resulting data is analyzed statistically for evaluation purposes. Two configurations are used in the simulation: one with 4 reference points and another with 5. In the former configuration, the 4 reference points are positioned at the centers of the small regions located in the four corners of the 6 m × 6 m square, as shown in Figure 7. In the latter configuration, an additional reference point is added to the central small area, resulting in a total of 5 reference points, as shown in Figure 8.

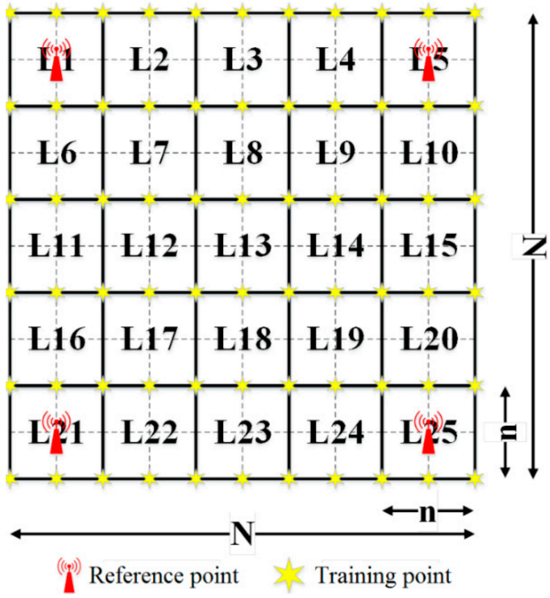

Figure 7. Illustration of 4 reference points.

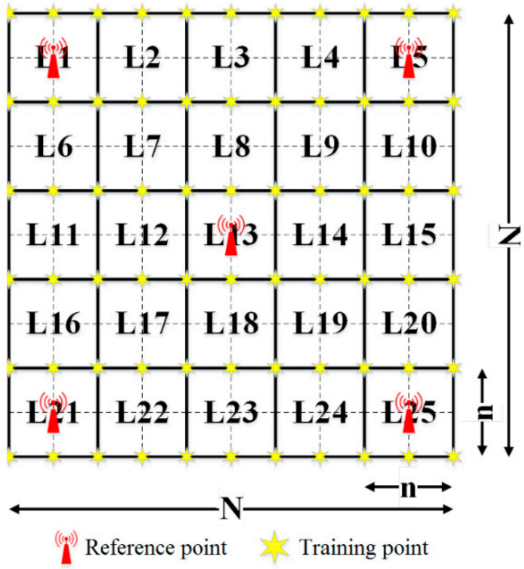

Figure 8. Illustration of 5 reference points.

In order to establish a sample database, 66 training points are gathered by utilizing channel model parameters that focus on the received signal strength indicator (RSSI) from the reference points. The target point, which is to be located, also receives RSSI readings from the reference points and compares them with the sample database to identify the positions of the K most similar training points. The location of the target point is estimated through a majority decision based on these identified training points. The simulation parameters for the RSSI channel model can be found in Table 1, where σ = 2 represents a simple environment and 10 represents a more complex environment. The DE algorithm parameters are listed in Table 2. The simulation tool employed in this study is MATLAB 2015B edition. Through extensive simulations and statistical analyses based on these parameters, this study aims to evaluate and compare the effectiveness and accuracy of the proposed DE-KNN algorithm in indoor positioning. The findings from these simulations provide valuable insights into the optimization and advancement of indoor positioning technologies, contributing to the overall improvement of indoor positioning systems under various scenarios.

Table 1. RSSI channel model parameters.

Parameters	Value
Transmission Power	2 mW
Carrier Frequency	2.4 GHz
Path Loss Exponent	4.5
Reference Distance	0.2 m
Antenna gains G_t, G_r	1
Standard Deviation of Shadowing Fading	2 dBm, 10 dBm

Table 2. DE algorithm parameters.

Parameters	Value
Number of Population *NP*	100
Mutation Weight Factor *F*	0.5
Crossover Rate *CR*	0.9

In the simulation of KNN and DE-KNN area positioning method, random deployment and fixed deployment topologies were considered in this paper. The K values used in KNN algorithm for simulation are 3, 4, 5 and 6.

4.2. Simulation Results of KNN Area Positioning Method

In this study, our objective was to evaluate the average accuracy of localization through simulations by manipulating the K value of the KNN algorithm and the number of reference points. We utilized both fixed and random deployment methodologies for the sensors in order to analyze their effect on localization accuracy in a simple environment with a standard deviation of 2 ($\sigma = 2$) and a complex environment with a standard deviation of 10 ($\sigma = 10$). Our focus was specifically on assessing the influence of KNN-related parameters on the accuracy of area positioning.

In consideration of the study conducted in indoor environments with four reference points, the average accuracy of localization for various K values is presented in Tables 3 and 4. The deployment of sensors is considered in both fixed and random arrangements. Furthermore, in the case of indoor environments with five reference points, the average localization accuracy for different K values is shown in Tables 5 and 6. These findings shed light on how the positioning of sensors and the choice of K value affect the effectiveness of indoor localization.

Table 3. Average accuracy of random and fixed deployments with 4 RPs in a simple environment $\sigma = 2$ for KNN.

K Value	Fixed Deployment	Random Deployment
3	67.24%	65.84%
4	69.08%	68.96%
5	75.2%	73.44%
6	71.88%	69.16%

Table 4. Average accuracy of random and fixed deployments with 4 RPs in a complex environment $\sigma = 10$ for KNN.

K Value	Fixed Deployment	Random Deployment
3	62.96%	64.32%
4	66.96%	65.4%
5	72.64%	70.28%
6	68.24%	64.6%

Table 5. Average accuracy of random and fixed deployments with 5 RPs in a simple environment $\sigma = 2$ for KNN.

K Value	Fixed Deployment	Random Deployment
3	78.08%	70.88%
4	84.28%	74.8%
5	78.6%	75.08%
6	72.72%	69.4%

Table 6. Average accuracy of random and fixed deployments with 5 RPs in a complex environment $\sigma = 10$ for KNN.

K Value	Fixed Deployment	Random Deployment
3	74.92%	70.52%
4	79.24%	72.68%
5	78%	74.92%
6	69.24%	67.88%

According to Tables 3 and 4, it can be observed that for the KNN algorithm-based localization system with four reference points, the optimal average positioning accuracy is achieved with a K value of 5, regardless of whether the deployment is fixed or random.

However, the highest accuracy achieved is 75.2%, which is attained only when using the fixed deployment method in a simple environment.

Referring to Tables 5 and 6, when there are five reference points, for the fixed deployment method, the best average positioning accuracy is obtained with a K value of 4, both in simple and complex environments. On the other hand, for the random deployment method, the optimal average positioning accuracy is achieved with a K value of 5 in both environments. The highest accuracy recorded is 84.2%, which is achieved when using the fixed deployment method with a K value of 4 in a simple environment.

Based on comprehensive simulation results, it can be concluded that using five reference points improves the performance of the KNN localization method compared to four reference points. This improvement holds true in both simple and complex environments, regardless of fixed or random deployment. However, the performance improvement is more significant in fixed deployment compared to random deployment.

Moreover, the choice of K value impacts the efficiency of KNN, with larger K values leading to increased computational complexity but not necessarily improved performance. For example, simulation results demonstrate that with four reference points, regardless of the environment complexity, K = 5 yields optimal localization accuracy for both fixed and random deployments. With five reference points, in simple environments, K = 4 provides the best localization accuracy for fixed deployment, while K = 5 achieves the highest accuracy for random deployment.

Additionally, it is observed that even with optimization adjustments considering different environments and relevant parameters, the KNN algorithm localization still exhibits an error rate exceeding 10%.

4.3. Simulation Results of DE-KNN Area Positioning Method

Similarly, to the assessment of KNN localization method performance as described in Section 4.2, the evaluation of the DE-KNN positioning method's performance is conducted by adjusting the K value and reference point quantity of the DE-KNN algorithm in both a simple environment ($\sigma = 2$) and a complex environment ($\sigma = 10$) using fixed and random sensor deployment. This analysis aims to assess the impact of relevant parameters on the performance of DE-KNN localization method for regional positioning.

Tables 7 and 8 present the average localization accuracy of the DE-KNN area positioning method for different K values in a simple and complex indoor environment with four reference points, considering both fixed and random sensor deployment.

Table 7. Average accuracy of random and fixed deployments with 4 RPs in a simple environment $\sigma = 2$ for DE-KNN.

K Value	Fixed Deployment	Random Deployment
3	99.76%	98.24%
4	100%	100%
5	100%	100%
6	100%	99.96%

Table 8. Average accuracy of random and fixed deployments with 4 RPs in a complex environment $\sigma = 10$ for DE-KNN.

K Value	Fixed Deployment	Random Deployment
3	99.76%	99.92%
4	100%	100%
5	100%	100%
6	100%	99.56%

In a simple and complex indoor environment with five reference points, Tables 9 and 10 display the average localization accuracy of the DE-KNN area positioning method for different K values.

Table 9. Average accuracy of random and fixed deployments with 5 RPs in a simple environment σ = 2 for DE-KNN.

K Value	Fixed Deployment	Random Deployment
3	96.48%	94.52%
4	100%	100%
5	100%	99.72%
6	100%	99.96%

Table 10. Average accuracy of random and fixed deployments with 5 RPs in a complex environment σ = 10 for DE-KNN.

K Value	Fixed Deployment	Random Deployment
3	97.08%	98%
4	99.92%	100%
5	100%	99.96%
6	99.92%	100%

The simulation results from Tables 7–10 reveal that the DE-KNN positioning method has the potential to greatly enhance the accuracy of localization, ideally achieving a 100% accuracy rate.

4.4. Comparison of Simulation Results for KNN and DE-KNN Area Positioning Methods

The comparative results of the localization accuracy performance for KNN and DE-KNN positioning methods, utilizing a system with four reference points, under simple and complex environments and different sensor deployment approaches are presented in Tables 11 and 12. Similarly, the comparison of the localization performance for a system with five reference points in different environments is presented in Tables 13 and 14.

Table 11. Comparison of average accuracy for KNN and DE-KNN methods under fixed and random deployment in a simple environment σ = 2 with 4 reference points.

K Value	Fixed Deployment			Random Deployment		
	KNN	DE-KNN	Difference	KNN	DE-KNN	Difference
3	67.24%	99.76%	32.52%	65.84%	98.24%	32.40%
4	69.08%	100%	30.92%	68.96%	100%	31.04%
5	75.2%	100%	24.80%	73.44%	100%	26.56%
6	71.88%	100%	28.12%	69.16%	99.96%	30.80%
Average	70.85%	99.94%	29.09%	69.35%	99.55%	30.20%

Table 12. Comparison of average accuracy for KNN and DE-KNN methods under fixed and random deployment in a complex environment σ = 10 with 4 reference points.

K Value	Fixed Deployment			Random Deployment		
	KNN	DE-KNN	Difference	KNN	DE-KNN	Difference
3	62.96%	99.76%	36.80%	64.32%	99.92%	35.60%
4	66.96%	100%	33.04%	65.4%	100%	34.60%
5	72.64%	100%	27.36%	70.28%	100%	29.72%
6	68.24%	100%	31.76%	64.6%	99.56%	34.96%
Average	67.70%	99.94%	32.24%	66.15%	99.87%	33.72%

Table 13. Comparison of average accuracy for KNN and DE-KNN methods under fixed and random deployment in a simple environment σ = 2 with 5 reference points.

K Value	Fixed Deployment			Random Deployment		
	KNN	DE-KNN	Difference	KNN	DE-KNN	Difference
3	78.08%	96.48%	18.40%	70.88%	94.52%	23.64%
4	84.28%	100%	15.72%	74.8%	100%	25.20%
5	78.6%	100%	21.40%	75.08%	99.72%	24.64%
6	72.72%	100%	27.28%	69.4%	99.96%	30.56%
Average	78.42%	99.12%	20.70%	72.54%	98.55%	26.01%

Table 14. Comparison of average accuracy for KNN and DE-KNN methods under fixed and random deployment in a complex environment σ = 10 with 5 reference points.

K Value	Fixed Deployment			Random Deployment		
	KNN	DE-KNN	Difference	KNN	DE-KNN	Difference
3	74.92%	97.08%	22.16%	70.52%	98%	27.48%
4	79.24%	99.92%	20.68%	72.68%	100%	27.32%
5	78%	100%	22.00%	74.92%	99.96%	25.04%
6	69.24%	99.92%	30.68%	67.88%	100%	32.12%
Average	75.35%	99.23%	23.88%	71.50%	99.49%	27.99%

Based on the comparative results from Tables 11–14, it can be observed that in a simple indoor environment with four reference points, the positioning accuracy was improved by 29.09% and 30.20%, respectively, when the sensors were deployed in fixed and random arrangements, compared to using the KNN algorithm alone. In a complex indoor environment with four reference points, the positioning accuracy was increased by 32.24% and 33.72%, respectively. When the number of reference points increased to five, in a simple environment, the accuracy improvement for both fixed and random deployment was 20.70% and 26.01%, respectively. In a complex environment, the accuracy improvement was 23.88% and 27.99% for fixed and random deployment, respectively.

Furthermore, when utilizing the KNN positioning method alone, it resulted in an error rate greater than 10% for the best localization performance. However, by incorporating the proposed DE-KNN area positioning method, the localization performance can be significantly enhanced, achieving a 100% accuracy rate.

4.5. Further Discussion on the DE-KNN Method

The proposed DE-KNN method combines the KNN algorithm with the DE algorithm to enhance localization accuracy. For the KNN algorithm, integrating it with other classifiers allows for obtaining more accurate and robust classification results [61,62]. Similarly, by incorporating the DE algorithm with other optimization algorithms, hybrid strategies can be formed to leverage their individual strengths and improve the performance of the DE algorithm for specific problems [63,64]. Therefore, combining the DE-KNN method with additional optimization algorithms or machine learning techniques holds the potential for further improving localization accuracy.

The computational complexity of the DE algorithm is O(NP $\times$ D $\times$ G), where NP is the population size, D is the dimensionality of the problem, and G is the number of generations [65]. On the other hand, the computational complexity of the KNN algorithm is O(N $\times$ D), where N is the number of data samples and D is the dimensionality of each sample's features [66]. Therefore, the computational complexity of the DE-KNN algorithm is O(NP $\times$ D $\times$ G). Furthermore, as the number of sensor nodes increases, the complexity of communication and coordination between nodes also increases. Applying the DE algorithm to large-scale wireless sensor networks while maintaining performance may pose challenges. Similarly, the KNN algorithm exhibits poor scalability as the number of training data points increases. Processing time and memory requirements increase linearly with the size of the dataset, limiting its applicability in large-scale sensor network scenarios. Therefore, for the DE-KNN method, its scalability has limitations.

In real-world wireless sensor networks, the DE-KNN algorithm faces numerous limitations and challenges. These include the necessity to plan the deployment of reference points and training points, and to collect the Received Signal Strength Indicator (RSSI) values from reference points to establish a sample database of locations where training points are located. When the positions of reference points change, the data in the sample database must be updated in real-time. Additionally, factors such as computational complexity, scalability, energy consumption and dynamic environments are all issues and challenges to consider during implementation.

5. Conclusions

This study aims to investigate the performance of utilizing the DE algorithm for indoor area positioning in Wireless Sensor Networks. The simulation results demonstrate that the proposed DE-KNN positioning method significantly improves the localization accuracy compared to using the KNN algorithm alone. In a simple indoor environment with four reference points, the deployment of sensors in fixed and random arrangements increases the positioning accuracy by 29.09% and 30.20%, respectively. Similarly, in a complex indoor environment, the positioning accuracy is enhanced by 32.24% and 33.72%, respectively. When the number of reference points increases to 5, in the simple environment, the accuracy improvement for fixed and random deployment is 20.70% and 26.01%, respectively. In the complex environment, the accuracy improvement for fixed and random deployment is 23.88% and 27.99%, respectively.

In addition, the utilization of the KNN positioning method alone yields a localization performance with an error rate greater than 10% even under optimal conditions. Conversely, by adopting the proposed DE-KNN positioning method introduced in this study, the localization performance can be greatly enhanced, achieving a remarkable positioning accuracy rate of 100%.

In the future, we will further compare the DE-KNN method with other existing localization algorithms to validate its advantages. Additionally, we will investigate the effects of different parameters in the DE-KNN method, such as population size and mutation rate, as well as the number and arrangement strategy of reference points, on localization accuracy.

Author Contributions: Conceptualization, C.-H.C. and Y.-F.H.; Investigation, C.-H.C., K.-H.L., Y.-L.S. and Y.-F.H.; Methodology, S.-H.L., Y.-L.S. and Y.-F.H.; Software, C.-H.C. and K.-H.L.; Supervision, C.-H.C. and Y.-F.H.; Funding acquisition, Y.-F.H.; Writing—original draft, C.-H.C., S.-H.L., K.-H.L. and Y.-F.H.; Writing—review and editing, S.-H.L., C.-H.C. and Y.-F.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Council (NSTC), Taiwan, with grant number NSTC 112-2221-E-324-010.

Data Availability Statement: The original contributions presented in the study are included in the article, and no additional source data are required.

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

References

1. Tubaishat, M.; Madria, S. Sensor Networks: An Overview. *IEEE Potentials* **2003**, *22*, 20–23. [CrossRef]
2. Cai, S.; Liao, W.; Luo, C.; Li, M.; Huang, X.; Li, P. CRIL: An Efficient Online Adaptive Indoor Localization System. *IEEE Trans. Veh. Technol.* **2017**, *66*, 4148–4160. [CrossRef]
3. Xu, E.-Y.; Ding, Z.; Dasgupta, S. Target Tracking and Mobile Sensor Navigation in Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* **2011**, *12*, 177–186. [CrossRef]
4. Mazon-Olivo, B.; Pan, A. Internet of Things: State-of-the-Art, Computing Paradigms, and Reference Architectures. *IEEE Lat. Am. Trans.* **2022**, *20*, 49–63. [CrossRef]

5. Jamshed, M.A.; Ali, K.; Abbasi, Q.H.; Imran, M.A.; Ur-Rehman, M. Challenges, Applications, and Future of Wireless Sensors in Internet of Things: A Review. *IEEE Sens. J.* **2022**, *22*, 5482–5494. [CrossRef]
6. Nurlan, Z.; Zhukabayeva, T.; Othman, M.; Adamova, A.; Zhakiyev, N. Wireless Sensor Network as a Mesh: Vision and Challenges. *IEEE Access* **2022**, *10*, 46–67. [CrossRef]
7. Chettri, L.; Bera, R. A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems. *IEEE Internet Things J.* **2019**, *7*, 16–32. [CrossRef]
8. Asghari, P.; Rahmani, A.M.; Javadi, H.H.S. Internet of Things Applications: A Systematic Review. *Comput. Netw.* **2019**, *148*, 241–261. [CrossRef]
9. Czekster, R.M.; Grace, P.; Marcon, C.; Hessel, F.; Cazella, S.C. Challenges and Opportunities for Conducting Dynamic Risk Assessments in Medical IoT. *Appl. Sci.* **2023**, *13*, 7406. [CrossRef]
10. Han, G.; Jiang, J.; Zhang, C.; Duong, T.Q.; Guizani, M.; Karagiannidis, G.K. A Survey on Mobile Anchor Node Assisted Localization in Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2220–2243. [CrossRef]
11. 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]
12. Coluccia, A.; Fascista, A. A Review of Advanced Localization Techniques for Crowdsensing Wireless Sensor Networks. *Sensors* **2019**, *19*, 988. [CrossRef] [PubMed]
13. Thi, D.B.; Phuong, T.H.T. A Narrow Beam Steering Antenna Array for Indoor Positioning Systems Based on Wireless Sensor Network. *IEEE Access* **2022**, *10*, 89022–89030. [CrossRef]
14. Abdullah, O.A.; Al-Hraishawi, H.; Chatzinotas, S. Deep Learning-Based Device-Free Localization in Wireless Sensor Networks. In Proceedings of the 2023 IEEE Wireless Communications and Networking Conference (WCNC), Glasgow, UK, 26–29 March 2023; pp. 1–6. [CrossRef]
15. Sarcevic, P.; Csik, D.; Odry, A. Indoor 2D Positioning Method for Mobile Robots Based on the Fusion of RSSI and Magnetometer Fingerprints. *Sensors* **2023**, *23*, 1855. [CrossRef] [PubMed]
16. Alhmiedat, T. Fingerprint-Based Localization Approach for WSN Using Machine Learning Models. *Appl. Sci.* **2023**, *13*, 3037. [CrossRef]
17. Aroba, O.J.; Naicker, N.; Adeliyi, T.T. Node Localization in Wireless Sensor Networks Using a Hyper-Heuristic DEEC-Gaussian Gradient Distance Algorithm. *Sci. Afr.* **2023**, *19*, e01560. [CrossRef]
18. Paul, A.; Sinha, S. Received Signal Strength-Based Location Verification Technique in Wireless Sensor Networks Using Spline Curve. *J. Supercomput.* **2023**, *79*, 10093–10116. [CrossRef]
19. Fute, E.T.; Pangop, D.N.; Tonye, E. A New Hybrid Localization Approach in Wireless Sensor Networks Based on Particle Swarm Optimization and Tabu Search. *Appl. Intell.* **2023**, *53*, 7546–7561. [CrossRef]
20. Mehrabian, H.; Ravanmehr, R. Sensor Fusion for Indoor Positioning System through Improved RSSI and PDR Methods. *Future Gener. Comput. Syst.* **2023**, *138*, 254–269. [CrossRef]
21. Lee, S.-H.; Cheng, C.-H.; Lin, C.-C.; Huang, Y.-F. Target Positioning and Tracking in WSNs Based on AFSA. *Information* **2023**, *14*, 246. [CrossRef]
22. Yuvarasu, M.; Balaram, A.; Chandramohan, S.; Sharma, D.K. A Performance Analysis of an Enhanced Graded Precision Localization Algorithm for Wireless Sensor Networks. *Cybern. Syst.* **2023**, 1–16. [CrossRef]
23. Shilpi; Kumar, A. A Localization Algorithm Using Reliable Anchor Pair Selection and Jaya Algorithm for Wireless Sensor Networks. *Telecommun. Syst.* **2023**, *82*, 277–289. [CrossRef]
24. Mani, R.; Rios-Navarro, A.; Sevillano-Ramos, J.-L.; Liouane, N. Improved 3D Localization Algorithm for Large-Scale Wireless Sensor Networks. *Wirel. Netw.* **2023**, 1–16. [CrossRef]
25. Agarwal, N.; Gokilavani, M.; Nagarajan, S.; Saranya, S.; Alsolai, H.; Dhahbi, S.; Abdelaziz, A.S. Intelligent Aquila Optimization Algorithm-Based Node Localization Scheme for Wireless Sensor Networks. *Comput. Mater. Contin.* **2023**, *74*, 141–152. [CrossRef]
26. Yadav, P.; Sharma, S.C.; Singh, O.; Rishiwal, V. Optimized Localization Learning Algorithm for Indoor and Outdoor Localization System in WSNs. *Wirel. Pers. Commun.* **2023**, *130*, 651–672. [CrossRef]
27. Zhao, Q.; Xu, Z.; Yang, L. An Improvement of DV-Hop Localization Algorithm Based on Cyclotomic Method in Wireless Sensor Networks. *Appl. Sci.* **2023**, *13*, 3597. [CrossRef]
28. Fawad, M.; Khan, M.Z.; Ullah, K.; Alasmary, H.; Shehzad, D.; Khan, B. Enhancing Localization Efficiency and Accuracy in Wireless Sensor Networks. *Sensors* **2023**, *23*, 2796. [CrossRef] [PubMed]
29. Haq, M.Z.U.; Khan, M.Z.; Rehman, H.U.; Mehmood, G.; Binmahfoudh, A.; Krichen, M.; Alroobaea, R. An Adaptive Topology Management Scheme to Maintain Network Connectivity in Wireless Sensor Networks. *Sensors* **2022**, *22*, 2855. [CrossRef] [PubMed]
30. Gholampour, V.; Shiva, M. Adaptive Topology Control for Wireless Sensor Networks. In Proceedings of the Second IFIP International Conference on Wireless and Optical Communications Networks (WOCN 2005), Dubai, United Arab Emirates, 6–8 March 2005; pp. 546–550. [CrossRef]
31. Pardhasaradhi, B.; Srinath, G.; Raghu, J.; Srihari, P. Position Estimation in Indoor Using Networked GNSS Sensors and a Range-Azimuth Sensor. *Inf. Fusion.* **2023**, *89*, 189–197. [CrossRef]
32. Kang, X.; Wang, D.; Shao, Y.; Ma, M.; Zhang, T. An Efficient Hybrid Multi-Station TDOA and Single-Station AOA Localization Method. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 5657–5670. [CrossRef]

33. Cui, W.; Liu, Q.; Zhang, L.; Wang, H.; Lu, X.; Li, J. A Robust Mobile Robot Indoor Positioning System Based on Wi-Fi. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 172988141989666. [CrossRef]
34. Cao, S.; Chen, X.; Zhang, X.; Chen, X. Combined Weighted Method for TDOA-Based Localization. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1962–1971. [CrossRef]
35. Ahmed, S.; Abbasi, A.; Liu, H. A Novel Hybrid AoA and TDoA Solution for Transmitter Positioning. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Lloret de Mar, Spain, 29 November–2 December 2021; pp. 1–7. [CrossRef]
36. Wang, S.; Waadt, A.; Burnic, A.; Xu, D.; Kocks, C.; Bruck, G.H.; Jung, P. System Implementation Study on RSSI Based Positioning in UWB Networks. In Proceedings of the 2010 7th International Symposium on Wireless Communication Systems, York, UK, 19–22 September 2010; pp. 36–40.
37. Espinoza, A.; Reyes, E.; Tapia, M.; Ruiz, E.; Ruiz, J.; Cortez, J. Performance Comparison Between Simulated and Real Case Scenario of RSSI-Based Localization Algorithms on a WSAN. *IEEE Lat. Am. Trans.* **2016**, *14*, 115–121. [CrossRef]
38. Zanella, A.; Bardella, A. RSS-Based Ranging by Multichannel RSS Averaging. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 10–13. [CrossRef]
39. Kumar, S.M.; Sinha, S. Improved RSSI Based 3D Localization for Indoor Wireless Sensor Network. In Proceedings of the 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 1–2 July 2022; pp. 478–483.
40. He, S.; Chan, S.-H.G. INTRI: Contour-Based Trilateration for Indoor Fingerprint-Based Localization. *IEEE Trans. Mob. Comput.* **2017**, *16*, 1676–1690. [CrossRef]
41. Bengherbia, B.; Toubal, A.; Leboukh, A.; Saidi, M.A. The Influence of k-NN Parameters on the Localization Accuracy of WSNs in Indoor Environment. In Proceedings of the International Conference on Information, Communication Technology and System (ICTS), 2014, Surabaya, Indonesia, 24–25 September 2014.
42. Adege, A.B.; Yayeh, Y.; Berie, G.; Lin, H.P.; Yen, L.; Li, Y.R. Indoor Localization Using K-Nearest Neighbor and Artificial Neural Network Back Propagation Algorithms. In Proceedings of the 2018 27th Wireless and Optical Communication Conference (WOCC), Hualien, Taiwan, 30 April–1 May 2018; pp. 1–2. [CrossRef]
43. Peng, X.; Chen, R.; Yu, K.; Ye, F.; Xue, W. An Improved Weighted K-Nearest Neighbor Algorithm for Indoor Localization. *Electronics* **2020**, *9*, 2117. [CrossRef]
44. Hu, J.; Liu, D.; Yan, Z.; Liu, H. Experimental Analysis on Weight K-Nearest Neighbor Indoor Fingerprint Positioning. *IEEE Internet Things J.* **2019**, *6*, 891–897. [CrossRef]
45. Xu, S.; Chen, C.-C.; Wu, Y.; Wang, X.; Wei, F. Adaptive Residual Weighted K-Nearest Neighbor Fingerprint Positioning Algorithm Based on Visible Light Communication. *Sensors* **2020**, *20*, 4432. [CrossRef] [PubMed]
46. Xie, Y.; Wang, Y.; Nallanathan, A.; Wang, L. An Improved K-Nearest-Neighbor Indoor Localization Method Based on Spearman Distance. *IEEE Signal Process. Lett.* **2016**, *23*, 351–355. [CrossRef]
47. Liu, J.; Xu, B.G.; Wan, Y.H.; Tang, S.L.; Ding, X.K.; Wan, Q. A Range-Free Location Method Based on Differential Evolution Algorithm. In Proceedings of the 2013 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, Chengdu, China, 21–25 July 2013; pp. 381–384. [CrossRef]
48. Zhao, S.; Chai, S.; Zhang, B. Dynamic Differential Evolution Strategy Localization for Wireless Sensor Networks in Three-Dimensional Space. In Proceedings of the Cybersecurity and Cyberforensics Conference (CCC), Chengdu, China, 27–29 July 2016. [CrossRef]
49. Vahdat, A.R.; NourAshrafoddin, N.; Shiry Ghidary, S. Mobile Robot Global Localization Using Differential Evolution and Particle Swarm Optimization. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 1527–1534. [CrossRef]
50. Qiao, D.; Pang, G.K.H. A Modified Differential Evolution with Heuristic Algorithm for Nonconvex Optimization on Sensor Network Localization. *IEEE Trans. Veh. Technol.* **2015**, *65*, 1676–1689. [CrossRef]
51. Xiong, J.-Q.; Qin, Q.; Zheng, K.-M. A Distant Measurement Wireless Localization Correction Algorithm Based on RSSI. In Proceedings of the Seventh International Symposium on Computation Intelligence and Design, Hangzhou, China, 13–14 December 2014; pp. 176–278.
52. Mahmud, M.I.; Abdelgawad, A.; Yanambaka, V.P.; Yelamarthi, K. Packet Drop and RSSI Evaluation for LoRa: An Indoor Application Perspective. In Proceedings of the IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 14 June–31 July 2021; pp. 913–914.
53. Yao, L.; Peng, X.; Shi, D.; Liu, B. Design of Indoor Positioning System Based on RSSI Algorithm. In Proceedings of the International Conference on Management Science and Software Engineering (ICMSSE), Chengdu, China, 9–11 July 2021; pp. 145–148.
54. Ullmo, D.H.; Baranger, U. Wireless Propagation in Buildings: A Statistical Scattering Approach. *IEEE Trans. Veh. Techn.* **1999**, *48*, 947–955. [CrossRef]
55. De-La-Llana-Calvo, Á.; Lázaro-Galilea, J.-L.; Gardel-Vicente, A.; Rodríguez-Navarro, D.; Bravo-Muñoz, I.; Espinosa-Zapata, F. Characterization of Multipath Effects in Indoor Positioning Systems by AoA and PoA Based on Optical Signals. *Sensors* **2019**, *19*, 917. [CrossRef]
56. Malarski, K.; Thrane, J.; Bech, M.; Macheta, K.; Christiansen, H.; Petersen, M.; Ruepp, S. Investigation of Deep Indoor NB-IoT Propagation Attenuation. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5. [CrossRef]

57. Poli, R.; Kennedy, J.; Blackwell, T. Particle Swarm Optimization. *Swarm Intell.* **2007**, *1*, 33–57. [CrossRef]
58. Yang, Q. A New Localization Method Based on Improved Particle Swarm Optimization for Wireless Sensor Networks. *IET Softw.* **2022**, *16*, 251–258. [CrossRef]
59. Jiang, B.; Zhang, W.; Zhang, P. Research on an Improved Genetic Algorithm Which Can Improve the Node Positioning Optimized Solution of Wireless Sensor Networks. In Proceedings of the 2009 International Conference on Computational Science and Engineering, Vancouver, BC, Canada, 29–31 August 2009; pp. 949–954. [CrossRef]
60. Triguero, I.; García, S.; Herrera, F. A Preliminary Study on the Use of Differential Evolution for Adjusting the Position of Examples in Nearest Neighbor Classification. Proceedings of IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010; pp. 1–8. [CrossRef]
61. Farid, D.M.; Sworna, N.S.; Amin, R.; Sadia, N.; Rahman, M.; Liton, N.K.; Mukta, M.S.H.; Shatabda, S. Boosting k-Nearest Neighbor (kNN) Classification Using Clustering and Adaboost Methods. In Proceedings of the 2022 IEEE Region 10 Symposium (TENSYMP), Mumbai, India, 1–3 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
62. Zhu, X.; Ying, C.; Wang, J.; Li, J.; Lai, X.; Wang, G. Ensemble of ML-KNN for Classification Algorithm Recommendation. *Knowl.-Based Syst.* **2021**, *221*, 106933. [CrossRef]
63. Liu, J.; Wu, Y.; Xu, J.; Liu, X. A Novel Hybrid Algorithm Based on Lion Swarm Optimization and Differential Evolution Algorithm. 2023. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4355601 (accessed on 1 January 2023).
64. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Song, Y.; Xu, J. An Improved Differential Evolution Algorithm and Its Application in Optimization Problem. *Soft Comput.* **2021**, *25*, 5277–5298. [CrossRef]
65. Cui, L.; Huang, Q.; Li, G.; Yang, S.; Ming, Z.; Wen, Z.; Lu, N.; Lu, J. Differential evolution algorithm with tracking mechanism and backtracking mechanism. *IEEE Access* **2018**, *6*, 44252–44267. [CrossRef]
66. Ray, S. An Analysis of Computational Complexity and Accuracy of Two Supervised Machine Learning Algorithms—K-Nearest Neighbor and Support Vector Machine. In *Data Management, Analytics and Innovation, Proceedings of ICDMAI 2020, New Delhi, India, 17–19 January 2020*; Springer: Singapore, 2021; Volume 1, pp. 335–347.