International Journal of Geo-Information

MDPI

Article

A Robust Two-Module Framework for BLE-Based Indoor Pedestrian Tracking Using Smartphones

Ning Zhou [1], Gaofeng Jiang [1,*], Dongsheng Zhao [2] and Yuchen Yang [3]

[1] School of Civil Engineering, NingboTech University, Ningbo 315100, China; zhouning@nbt.edu.cn
[2] School of Environment and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China; dszhao@cumt.edu.cn
[3] Ningbo Langda Technology Co., Ltd., Ningbo 315100, China; yuchen.yang99@outlook.com
* Correspondence: gfjiang@nbt.edu.cn

Abstract

Indoor pedestrian tracking systems utilizing Bluetooth Low Energy (BLE) technology typically rely on the Weighted K-Nearest Neighbor (WKNN) algorithm for position estimation. However, this method suffers from high computational load in fingerprint matching and significant spatial ambiguity in positioning results. To solve these problems, this paper introduces an improved version of the WKNN algorithm, termed the Improved Self-Adaptive WKNN (ISAWKNN), by incorporating a Dual-Factor Similarity-based K-means (DFS-K-means) clustering approach and a dynamic adaptive K-value selection strategy. Additionally, an improved particle filter (IPF) is developed that uses the Subtraction-Average-Based Optimizer (SABO) to refine the particle distribution. Consequently, the ISAWKNN and IPF algorithms are integrated to form a two-module algorithm named ISAWKNN-IPF, where the initial position estimate from the ISAWKNN algorithm serves as a measurement input for the IPF algorithm to achieve a precise position estimate. Experimental results demonstrate that the ISAWKNN-IPF algorithm not only reduces positioning error by approximately 50% compared to the traditional WKNN approach but also achieves superior performance relative to several state-of-the-art methods. The proposed algorithm has the capability to enhance the quality of existing indoor location-based services.

Keywords: Bluetooth Low Energy (BLE); fingerprint-based positioning; indoor pedestrian tracking; particle filter (PF); Weighted K-Nearest Neighbor (WKNN)

Check for updates

Academic Editors: Wolfgang Kainz, Eliseo Clementini and Zoe Falomir

Received: 18 August 2026
Revised: 11 November 2026
Accepted: 16 November 2026
Published: 18 November 2026

Copyright: © 2026 by the authors. Published by MDPI on behalf of the International Society for Photogrammetry and Remote Sensing. 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.

1. Introduction

With the rapid advancement of smart cities and Internet of Things (IoT), the demand for high-precision, real-time location information has grown significantly, particularly given that over 80% of human activities globally occur indoors [1]. In this context, pedestrian dynamic positioning has emerged as a key enabler for next-generation location-based services (LBS). The proliferation of mobile terminals such as smartphones and smart bracelets, alongside rapid advancements in IoT technologies, has facilitated pedestrian positioning applications like smart retail navigation, industrial personnel management, and patient monitoring in smart healthcare systems [2,3]. Bluetooth Low Energy (BLE), a wireless personal area network technology designed for low-power, short-range communication, meets these needs by supporting intermittent data transmission, making it distinct from other indoor positioning technologies (such as UWB, ZigBee, and Wi-Fi) due to its widespread use, extremely low power consumption, simple deployment, and cost-effectiveness [4–6].

Consequently, BLE-based indoor positioning technology has garnered significant attention as a research focus within the domain of indoor pedestrian tracking.

Indoor multipath effects and non-line-of-sight (NLOS) propagation reduce the effectiveness of trilateration, making fingerprinting the preferred approach in BLE-based positioning systems [7]. BLE systems commonly use the Received Signal Strength Indicator (RSSI) as a key metric for signal strength measurement. BLE-based positioning performance depends critically on both the quality of the fingerprint database and the design of the matching algorithm. The weighted K-nearest neighbor (WKNN) algorithm provides a straightforward approach to fingerprint matching, but its application to pedestrian tracking involves several interacting challenges [8,9]. Searching the entire fingerprint database increases the matching workload as the number of reference points (RPs) grows. Moreover, similar RSSI patterns may occur at geographically separated locations, while differences in signal variability across beacons can affect the reliability of similarity measurements. A fixed number of neighbors may therefore be unsuitable for different local fingerprint distributions. Beyond individual position estimates, pedestrian tracking also requires temporal information to constrain successive estimates and reduce the influence of fluctuating RSSI measurements.

These challenges motivate a coordinated design of fingerprint matching and motion-based position estimation. Restricting the fingerprint search can improve matching efficiency, but the resulting candidate set must retain useful location information. Similarly, temporal filtering benefits from reliable fingerprint measurements and an appropriate pedestrian motion model. The relationships between search-region construction, neighbor selection, and subsequent filtering are therefore central to the design of the positioning framework.

To address the limitations of existing weighted K-nearest neighbor (WKNN) enhancement strategies and improve indoor pedestrian tracking performance, this paper proposes an improved self-adaptive WKNN (ISAWKNN) algorithm for initial target position estimation. Subsequently, to improve the trajectory smoothness, an improved particle filter (IPF) is developed that treats the initial position provided by ISAWKNN as the measurement input and produces the precise position estimate. By integrating these two stages, this paper proposes a complete two-module indoor pedestrian tracking framework named ISAWKNN-IPF. The main contributions of this article are listed as follows:

1. To reduce the computational load of fingerprint matching in the online phase, a Dual-Factor Similarity-based K-means (DFS-K-means) clustering method is proposed, which integrates signal and coordinate factors to cluster reference points into homogeneous regions during the offline phase.
2. An ISAWKNN algorithm is developed, where the Mahalanobis distance of RSSI replaces the conventional Euclidean distance as the fingerprint similarity metric, coupled with a self-adaptive method for determining the optimal K value.
3. An IPF algorithm is proposed to mitigate particle degeneracy and optimize particle distribution in the update stage; it embeds a Subtraction-Average-Based Optimization (SABO) strategy into the particle filter (PF) framework to refine the distribution of particles toward regions of higher measurement likelihood, thus driving samples to converge to the global optimal region for accurate positioning.
4. A two-module indoor pedestrian tracking algorithm ISAWKNN-IPF is constructed, which takes the initial position estimate from ISAWKNN as the measurement input of IPF to achieve high-precision positioning, and its effectiveness is verified through rigorous comparative experiments.

The remainder of the paper is organized as follows. Section 2 presents the theoretical foundations of the WKNN and PF algorithms and analyses their limitations. Section 3 details the principle of the proposed ISAWKNN-IPF algorithm. Section 4 describes the experimental

setup, presents the positioning performance results, and provides a comparative analysis. Section 5 discusses the experimental results and limitations of the proposed framework. Finally, Section 6 concludes the paper and outlines directions for future work.

2. Related Work

This section introduces the conventional WKNN and particle filter algorithms in Sections 2.1 and 2.2, respectively. Section 2.3 reviews representative positioning methods, compares their strengths and limitations, and discusses the motivation for the proposed framework.

2.1. WKNN Algorithm

As shown in Figure 1, fingerprint positioning consists of two phases: the offline phase (a.k.a. training phase) and the online phase (a.k.a. localization phase) [10]. In the offline phase, each Bluetooth beacon serves as an anchor point (AP). RPs with known coordinates are selected, and RSSI data from each AP are collected at every RP to form fingerprints that pair the RP coordinates with the corresponding RSSI measurements. All fingerprints are uploaded to the server to construct a fingerprint database. Consider that there are n anchor points and m reference points deployed in an indoor environment. A fingerprint database can be established as follows:

$$\mathbf{FP} = \begin{bmatrix} \mathbf{FP}_1 \\ \mathbf{FP}_2 \\ \vdots \\ \mathbf{FP}_m \end{bmatrix} = \begin{bmatrix} x_1, & y_1, & RSSI_{11}, & RSSI_{12}, & \cdots & RSSI_{1n} \\ x_2, & y_2, & RSSI_{21}, & RSSI_{22}, & \cdots & RSSI_{2n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ x_m, & y_m, & RSSI_{m1}, & RSSI_{m2}, & \cdots & RSSI_{mn} \end{bmatrix} \quad (1)$$

where $\mathbf{FP}_i$ $(i = 1,2,\ldots,m)$ denotes the fingerprint at the ith reference point, x_i,$y_i$$(i = 1,2,\ldots,m)$ denotes the coordinates of the ith reference point, and $RSSI_{ij}$ $(i = 1,2,\ldots,m;\ j = 1,2,\ldots,n)$ denotes the RSSI collected from the jth beacon at the ith reference point. In the online phase, the BLE tag collects RSSI values at the unknown location. Suppose that the RSSI data collected at the target point is denoted as $\boldsymbol{rssi} = [rssi_1,rssi_2,\ldots,rssi_n]$, the WKNN algorithm first evaluates the similarity between the target point and the ith reference point using the Euclidean distance of the RSSI, given by

$$D_{E,i} = \sqrt{\sum_{j=1}^{n} \left(rssi_j - RSSI_{ij}\right)^2} \quad (2)$$

The smaller the distance calculated by Equation (2), the higher the similarity between the two points. Then, all Euclidean distances are sorted in ascending order, and the top K reference points with the highest similarity are selected as the nearest neighbors. The weights for each of the selected nearest neighbors are set as follows:

$$w_i = \frac{1/D_{E,i}}{\sum_{i=1}^{K} 1/D_{E,i}} \quad (3)$$

where i denotes the indices of the selected K nearest neighbors. Finally, the coordinates of the target point are computed by

$$(x, y) = \sum_{i=1}^{K} w_i (x_i, y_i) \quad (4)$$

As discussed above, traditional WKNN employs a fixed K that cannot adapt to uneven RP density or varying indoor conditions, often leading to bias and noise. The weighting process in Equation (3) relies solely on the RSSI-based Euclidean distance as the fitness evaluation criterion. This single similarity metric fails to account for the spatial relationships among RPs and the target location. Consequently, some reference points that are considerably distant from the actual location may be erroneously included.

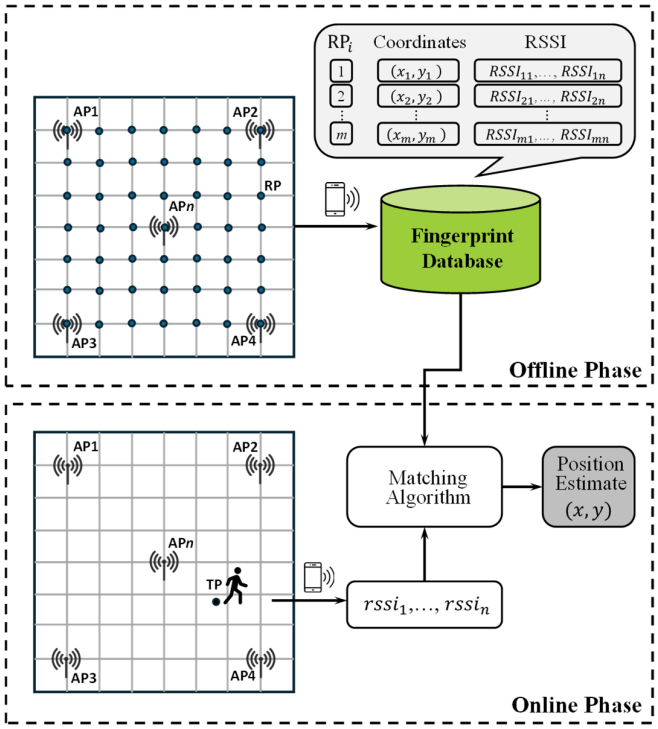


Figure 1. Fingerprint-based indoor positioning system.

2.2. Particle Filter (PF)

The particle filter, also known as the sequential Monte Carlo method, is a computational algorithm for estimating the state of a dynamic system over time. It is particularly well suited to nonlinear systems or those subject to non-Gaussian noise, where traditional methods such as the Kalman filter often perform poorly. The particle filter represents the probability distribution of the system's state using a group of random samples, called particles. Each particle represents a possible state of the system, and it carries an associated weight that reflects how likely that state is given the observed measurements. In the PF algorithm, a general form of the state-space model of a dynamic target can be defined as follows [11]:

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k \\ \boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \end{cases} \tag{5}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ are the state vector and measurement vector at time step k, respectively. $\boldsymbol{x}_k$ is usually assumed to follow a first-order Markov process, and $\boldsymbol{y}_k$ is conditionally independent given the states. $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are the additive process and measurement noise, respectively. $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are two known (probably non-linear) transitional and measurement functions, respectively.

PF uses a posterior distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ to represent the target state. $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ can be recursively approximated by a set of random particles $\{\boldsymbol{x}_k^i\}_{i=1}^{N_p}$ that are sampled from a known distribution $q(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$, given by

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \sum_{i=1}^{N_p} w_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i) \tag{6}$$

where $\delta(\cdot)$ is the Dirac delta function, N_p is the number of particles, and w_k^i is the normalized weight of the ith particle, calculated by

$$\tilde{w}_k^i = \tilde{w}_{k-1}^i \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k^i)p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i)}{q(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i, \boldsymbol{y}_k)} \tag{7}$$

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{N_p} \tilde{w}_k^i} \tag{8}$$

PF operates recursively through two main steps: prediction and update. In the prediction step, each particle is propagated forward in time according to the system's motion model. In the update step, when a new measurement becomes available, the weight of each particle is adjusted based on how well its predicted state matches the actual observation. To avoid degeneracy, wherein most particles have negligible weights after a few iterations, resampling is performed by discarding low-weight particles and duplicating high-weight ones to preserve representativeness. However, this conventional resampling strategy often reduces particle diversity, leading to particle impoverishment. In positioning applications, impoverishment can degrade accuracy and even cause filter divergence or tracking failure. To address this problem, recent work uses metaheuristic techniques (such as genetic algorithms (GA) [12], particle swarm optimization (PSO) [13], and firefly algorithms (FA) [14]) to refine resampling by more effectively matching the proposal distribution to the posterior. A comprehensive introduction of PF can be found in [15].

2.3. Comparative Analysis of Existing Methods

Building on the WKNN and PF principles introduced above, existing positioning methods can be examined in terms of search-space reduction, fingerprint weighting, adaptive neighbor selection, and temporal estimation. These aspects address different sources of positioning error and computational cost, and their interactions are relevant to the design of an integrated pedestrian tracking framework.

Search-space reduction improves matching efficiency by restricting the fingerprints considered during positioning. Representative strategies include regional constraints [16] and clustering reference points into partitions [17,18]. For regional constraints, Li et al. [16] restrict matching to a circular region and introduce a fault-tolerant mechanism to adjust that region. For fingerprint clustering, Skeleton-Assisted Learning-Based Clustering (SALC) [17] combines skeleton-based shortest-path similarity with time-varying RSS measurements, using map information and monitor points to support database reconstruction. Chen and Pu [18] combine Canopy and K-means clustering to partition the fingerprint database, followed by subset selection and Bayesian position estimation. These approaches reduce the matching workload through different mechanisms, but their effectiveness depends on retaining relevant RPs within the selected region or subset. Clustering methods also involve different information requirements, as illustrated by the map and monitoring information used in SALC. Therefore, search-space reduction requires balancing matching efficiency, candidate-set reliability, and the information needed to construct and maintain the search regions.

Fingerprint weighting addresses how selected RPs contribute to position estimation. Ma et al. [19] combine position estimates obtained from WKNN and probabilistic estimation, while Xue et al. [20] develop a weighting strategy based on the physical distance inferred from RSSI. Peng et al. [21] further combine RSS-space distance with a propagation-model-derived distance for RP selection and weighting. These approaches account for information beyond conventional RSSI Euclidean distance. However, propagation-based distances introduce dependence on signal-attenuation model parameters. Moreover, the reliability of the resulting weights remains linked to the selected fingerprints and the variability in their measurements, motivating joint consideration of similarity measurement and candidate selection.

Adaptive neighbor selection addresses the limitations of a fixed K value. Oh and Kim [22] determine the neighborhood using RSS-distance patterns and spatial exclusion rules, while Hu et al. [23] adjust K according to signal strength. Zhang et al. [24] incorporate

improved RSS similarity, position proximity, and historical trajectory information. Spatial constraints are also employed by Zhang and Mao [25] to restrict dispersed positioning results. These studies establish the value of adaptive neighborhoods and spatial information. Their different selection rules nevertheless require appropriate configuration, and methods using historical estimates must account for the reliability of those estimates. The relevant design question is therefore how signal characteristics and RP geometry should jointly guide neighborhood construction.

Learning-based methods provide alternative representations of the fingerprint–position relationship. Random Forest–Kernel Extreme Learning Machine (RF-KELM) [26] combines fingerprint transformation and random-forest-based access-point selection with a trained kernel extreme learning machine. More recently, Ferrero-López et al. [27] compare neural-network architectures for BLE fingerprint positioning and emphasize that training metrics alone may inadequately represent performance along actual trajectories. These methods offer learned nonlinear mappings while introducing model-training and selection requirements. Their findings also highlight the importance of signal preprocessing, missing measurements, and trajectory-based evaluation when assessing positioning performance.

Temporal estimation complements fingerprint matching by incorporating motion information across successive time steps. Optimization-based PF methods, such as the genetic optimization resampling method of Zhou et al. [12], improve particle distributions through additional optimization operations, with an associated increase in computational cost. Qian et al. [28] combine BLE fingerprint positioning with improved particle likelihood functions based on the means and variances of nearby fingerprints. Their method incorporates fingerprint statistics, whereas particle propagation uses an assumed stride length and random movement directions. For smartphone-based pedestrian tracking, IMU-derived PDR displacement provides an alternative basis for particle prediction. These approaches indicate that motion prediction, measurement representation, and particle refinement should be considered together, alongside their computational requirements.

Table 1 summarizes representative methods and their design trade-offs. The comparison concerns methodological characteristics rather than directly ranking accuracy results obtained using different datasets and experimental conditions. The comparison highlights the complementary roles of fingerprint organization, neighborhood construction, and temporal estimation. Errors introduced during candidate selection and matching can affect subsequent tracking, while additional particle refinement increases computational cost. Building on these research directions, the proposed framework jointly considers RSSI variability and RP coordinates in offline clustering, uses adaptive neighbor selection for initial positioning, and combines PDR-driven prediction with SABO-based particle refinement. Its evaluation examines the contributions and computational costs of these components within a common BLE pedestrian tracking framework.

Table 1. Comparison of representative RSSI-based indoor positioning methods.

Method	Input and Main Mechanism	Main Contribution	Limitation or Design Trade-Off
SALC [17]	Wi-Fi RSSI, floor-plan skeleton, and monitor points; spatially informed clustering and database reconstruction	Supports adaptation to time-varying signal conditions	Requires map information and monitor points
Dual-clustering Bayesian method [18]	Wi-Fi RSSI; Canopy–K-means clustering and Bayesian estimation	Reduces the fingerprint matching workload	Final estimation depends on the selected fingerprint subset

Table 1. *Cont.*

Method	Input and Main Mechanism	Main Contribution	Limitation or Design Trade-Off
RF-KELM [26]	Wi-Fi RSSI; fingerprint transformation, feature selection, and learned position prediction	Combines feature selection with nonlinear position estimation	Requires model training and consistency between training and positioning features
Improved WKNN [21]	Wi-Fi RSSI; fusion of RSS-space and propagation-model-derived distances	Incorporates complementary distance information into RP selection and weighting	Depends on signal-attenuation model parameters
Historical-information-based adaptive KNN [24]	Wi-Fi RSSI and historical position information	Combines adaptive neighborhoods with positional proximity	Relies partly on the reliability of historical estimates
BLE neural-network positioning [27]	Processed BLE RSSI fingerprints; neural-network position estimation	Learns nonlinear fingerprint–position relationships	Requires labeled training data, model selection, and trajectory-based validation
Improved BLE fingerprint PF [28]	BLE RSSI; likelihood functions using fingerprint means and variances	Incorporates fingerprint statistics into particle weighting	Particle propagation uses assumed strides and random directions

3. Methods

The flowchart of the proposed ISAWKNN-IPF algorithm is shown in Figure 2. The algorithm comprises two phases: offline fingerprint clustering and online position estimation. Given a fingerprint database, the offline phase partitions the reference points into homogeneous clusters using a DFS-K-means method that jointly considers signal and coordinate information, thereby constructing an enhanced fingerprint database. In the online phase, an initial estimate of the target's position is derived using the ISAWKNN algorithm. Subsequently, an improved particle filter refines this estimate by incorporating a target state transition model and treating the initial position as the measurement input, yielding a precise final position. In the following, the details of the two phases are described in Sections 3.1 and 3.2, respectively.

3.1. Offline Fingerprint Clustering Phase

3.1.1. Fingerprint Collection

Consider an indoor environment with m reference points and L BLE beacons with known coordinates. Assuming that N RSSI samples are collected from each beacon at the ith reference point, the RSSI fingerprint vector at this point can be expressed as follows:

$$\boldsymbol{RSSI}_i = \left[\frac{1}{N}\sum_{k=1}^{N} RSSI_{i,k}^{1}, \ldots, \frac{1}{N}\sum_{k=1}^{N} RSSI_{i,k}^{L} \right] \tag{9}$$

where $RSSI_{i,k}^{n}$ denotes the RSSI value obtained from the nth (n = 1,2,...,L) BLE beacons during the kth (k = 1,2,...,N) data collection at the ith reference point. Since data collections at each reference point are independent of one another, the covariance matrix of RSSI values at the ith reference point can be expressed as

$$\Sigma_i = diag\left(\sigma_{i,1}^2, \sigma_{i,2}^2, \ldots, \sigma_{i,L}^2\right) \tag{10}$$

where $\sigma_{i,n}^2$ represents the variance of the RSSI data from the nth BLE beacon at the ith reference point, calculated by

$$\sigma_{i,n}^2 = \frac{1}{N-1}\sum_{j=1}^{N}\left(RSSI_{i,j}^n - \frac{1}{N}\sum_{k=1}^{N} RSSI_{i,k}^n\right)^2 \quad (11)$$

During the construction of the fingerprint database, the RSSI vector from Equation (9), the covariance matrix from Equation (10), and the corresponding reference point coordinates are all uploaded to the database.

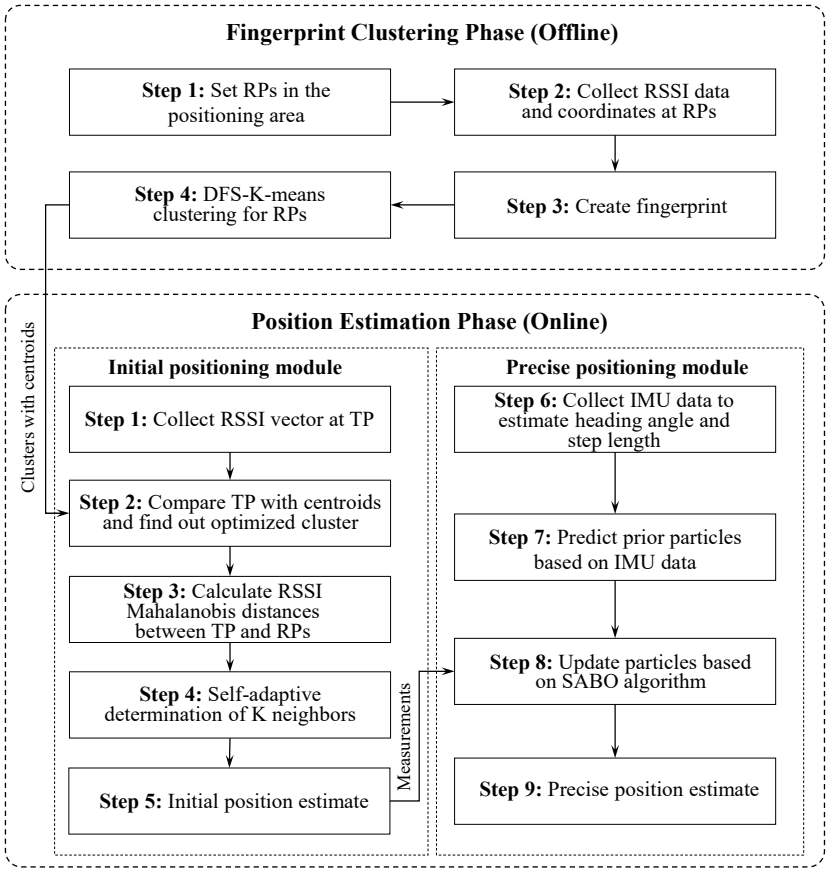


Figure 2. Flowchart of the ISAWKNN-IPF positioning algorithm.

3.1.2. DFS-K-Means Clustering

Traditional fingerprint positioning algorithms require matching the RSSI values collected at the target point with those collected at various reference points. When the number of reference points is large, this significantly increases the computational load during the online positioning phase. To address this issue, this paper proposes a DFS-K-means clustering algorithm for clustering offline fingerprint data. The steps of the DFS-K-means clustering algorithm are described as follows.

First, q points are randomly drawn from the m reference points to function as cluster centroids. The similarity between the remaining m-q reference points and the q centroids is computed using a metric that jointly accounts for signal strength and spatial coordinates. Given a reference point A and a cluster centroid B, the similarity $S(A,B)$ is evaluated using the product of the Mahalanobis distance of RSSI and the Euclidean distance of coordinates, given by

$$S(A,B) = -D_M(\boldsymbol{RSSI}_A, \boldsymbol{RSSI}_B) \times d_{AB} \quad (12)$$

where $D_M(\boldsymbol{RSSI}_A,\boldsymbol{RSSI}_B)$ denotes the Mahalanobis distance between the RSSI values of the two points, given by

$$D_M(\boldsymbol{RSSI}_A,\boldsymbol{RSSI}_B) = \sqrt{(\boldsymbol{RSSI}_A - \boldsymbol{RSSI}_B)\Sigma^{-1}(\boldsymbol{RSSI}_A - \boldsymbol{RSSI}_B)^{\mathrm{T}}} \tag{13}$$

where $\boldsymbol{RSSI}_A$ and $\boldsymbol{RSSI}_B$ are the RSSI vectors at reference point A and cluster centroid point B, respectively. Σ is the covariance matrix at cluster centroid point B calculated by Equation (10), and d_{AB} represents the Euclidean distance between their coordinates.

For each remaining reference point, calculate its similarity to the q cluster centroids using Equation (12) and assign it to the cluster with the highest similarity until all reference points have been processed. After clustering all reference points, update each cluster's centroid and compute the corresponding covariance matrix at the updated centroid. For each updated cluster centroid, reassign all reference points to clusters as described above, and repeat this procedure for a predefined number of iterations or until the centroids converge to those from the previous iteration. Consequently, all reference points are divided into q clusters. Since the Mahalanobis distance can scale distances using the covariance matrix, it is more effective in handling fingerprint features with different distributions and variances. The DFS-K-means clustering algorithm accounts for both the Mahalanobis distance of RSSI between two points and their positional coordinates during the similarity calculation. This can facilitate the assignment of nearby reference points to the same cluster, thereby improving clustering accuracy.

3.2. Online Position Estimation Phase

3.2.1. Initial Positioning Module Based on the ISAWKNN Algorithm

This paper proposes an ISAWKNN algorithm that can determine the K value adaptively. Suppose that the RSSI vector collected at the target point is denoted as rssi. First, calculate the RSSI Mahalanobis distance between the target point and the q centroids of each cluster. Rank the q centroids according to the RSSI Mahalanobis distance and select the cluster associated with the centroid having the smallest Mahalanobis distance. Suppose the selected cluster contains m_s reference points. Then, calculate the RSSI Mahalanobis distance between the target point and each reference point within the selected cluster using Equation (13), where $\boldsymbol{RSSI}_A$ and $\boldsymbol{RSSI}_B$ are replaced by $\boldsymbol{rssi}$ and $\boldsymbol{RSSI}_i$, respectively. Here, $\boldsymbol{RSSI}_i$ denotes the RSSI vector of the ith ($i = 1,2,\ldots,m_s$) reference point in the selected cluster.

According to the calculated RSSI Mahalanobis distances, sort the reference points in the selected cluster in ascending order as $\{RP_1, RP_2, \ldots, RP_{m_s}\}$, where RP_1 denotes the reference point with the smallest RSSI Mahalanobis distance (i.e., with the highest similarity to the target point). Calculate the mean and standard deviation of the Euclidean distances of the coordinates between the reference point RP_1 and the remaining reference points $\{RP_2, \ldots, RP_{m_s}\}$, given by

$$\overline{d} = \frac{\sum_{i=2}^{m_s} d_{1i}}{m_s - 1} \tag{14}$$

$$\sigma_d = \sqrt{\frac{\sum_{i=2}^{m_s}\left(d_{1i} - \overline{d}\right)^2}{m_s - 1}} \tag{15}$$

where d_{1i} is the coordinate Euclidean distance between RP_1 and $RP_i (i = 2,\ldots,m_s)$. Based on the calculated standard deviation σ_d, identify all the reference points with distances d_{1i} less than or equal to σ_d as the neighbors, and the number of neighbors is denoted as K. Then, calculate the RSSI Mahalanobis distance $D_{M,i}$ between the target point and each neighbor by Equation (13), and compute the weights of these K neighbors, given by

$$w_i = \frac{1/D_{M,i}}{\sum_{i=1}^{K} 1/D_{M,i}} \tag{16}$$

Finally, use Equation (4) to calculate the initial position estimate of the target point, denoted as (x_{FP}, y_{FP}).

3.2.2. Precise Positioning Module Based on the Improved PF Algorithm

Precise positioning is achieved by applying an improved PF algorithm after obtaining an initial position estimate from ISAWKNN. Given the target's position estimate at the previous time step, acceleration and angular velocity measurements from the smartphone's inertial measurement unit (IMU) are used to predict its position at the current time step. To this end, a pedestrian dead reckoning (PDR) algorithm is incorporated into the PF framework to generate the prior particle distribution during the prediction stage. The state-space model for pedestrian positioning is formulated as follows:

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{G}\boldsymbol{w}_k \\ \boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \end{cases} \tag{17}$$

with

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{B} = \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

In the state model, $\boldsymbol{F}$ is the state-transition matrix. $\boldsymbol{x}_k = [x_k, \Delta x_k, y_k, \Delta y_k]^{\boldsymbol{T}}$ is the state vector, in which x_k and y_k are the pedestrian's position coordinates, and Δx_k and Δy_k are the displacement components over the interval from time step $k-1$ to k. $\boldsymbol{B}$ is the control-input matrix that incorporates the PDR-derived displacement from time step $k-1$ to k into the state prediction. $\boldsymbol{u}_k = \left[\Delta\hat{x}_k^{\mathrm{PDR}}, \Delta\hat{y}_k^{\mathrm{PDR}}\right]^{\mathrm{T}}$ is the input vector that represents this displacement in the positioning coordinate frame, where $\Delta\hat{x}_k^{\mathrm{PDR}}$ and $\Delta\hat{y}_k^{\mathrm{PDR}}$ are the displacement components along the x and y axes, respectively, calculated by

$$\begin{cases} \Delta\hat{x}_k^{\mathrm{PDR}} = d\cos\theta \\ \Delta\hat{y}_k^{\mathrm{PDR}} = d\sin\theta \end{cases} \tag{18}$$

where d and θ are the estimated step length and heading angle, respectively. The step length can be estimated using the Weinberg nonlinear step-length estimation model in [29]. In the observation model, $\boldsymbol{z}_k = [x_{FP}, y_{FP}]^{\mathrm{T}}$ is the measurement vector, in which x_{FP} and y_{FP} are the initial position coordinates estimated by the ISAWKNN algorithm. The matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ are the process noise driving matrix and the observation matrix, respectively. The two-dimensional vectors $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ denote process noise and measurement noise, respectively, with corresponding covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. The subscript k denotes the kth time step.

In the update stage, the initial position estimates provided by ISAWKNN serve as measurements. To address particle degeneracy while preventing particle impoverishment, the proposed algorithm refines the particle distribution by the SABO algorithm [30]. In comparison to classical heuristic optimization algorithms (such as GA (1988) and PSO (1995)) and state-of-the-art algorithms published in the literature (such as White Shark Optimizer (WSO, 2022) [31] and Golden Jackal Optimization (GJO, 2022) [32]), SABO exhibits superior optimization capability and faster convergence speed. The basic inspiration of SABO is to use the subtraction average of searcher agents to update the states of population members in the search space. Given the structural similarities between the PF and SABO, which both operate on populations of candidate solutions, the filtering update step can be reformulated as a sequential optimization problem, enabling the integration of SABO's evolutionary principles into the PF framework. Given a prior particle pool $\{\boldsymbol{x}_k^i\}_{i=1}^{N_p}$ (N_p is

the number of particles) obtained from the PDR-based particle sampling, a particle x_k^i is updated through the SABO strategy by

$$x_k^{i,*} = x_k^i + \vec{r}_i \odot \frac{1}{N_p}\left(\sum_{j=1}^{N_p}\left(x_k^i -_v x_k^j\right)\right) \tag{19}$$

where $x_k^{i,*}$ is the updated particle, $\vec{r}_i$ is a random vector and its elements are drawn from a standard normal distribution. $x_k^j\left(j = 1, 2, \ldots, N_p\right)$ is an exploration particle. $x_k^i -_v x_k^j$ denotes the subtraction operation, given by

$$x_k^i -_v x_k^j = \text{sign}\left(f\left(x_k^i\right) - f\left(x_k^j\right)\right)\left(x_k^i - \vec{v} \odot x_k^j\right) \tag{20}$$

where $\vec{v}$ is a random vector whose components are randomly selected from the set {1,2}. $\odot$ denotes the Hadamard product. $f(\cdot)$ denotes the particle fitness. Owing to the adoption of the state transition function as the proposal distribution during sampling and the assumption of independent particle weights after each iterative update, the fitness value is defined as the likelihood that a particle's predicted measurement corresponds to the observed measurement vector, expressed as

$$f\left(x_k^i\right) = \frac{1}{2\pi\sqrt{\det(\boldsymbol{R})}}\exp\left[-\frac{1}{2}\left(z_k - \boldsymbol{h}\left(x_k^i\right)\right)^{\mathrm{T}}\boldsymbol{R}^{-1}\left(z_k - \boldsymbol{h}\left(x_k^i\right)\right)\right] \tag{21}$$

where $\boldsymbol{h}(\cdot)$ denotes the measurement function, and $\boldsymbol{R}$ is the measurement noise covariance matrix, estimated offline using the calibration procedure described in Section 4. $x_k^{i,*}$ is selected to replace x_k^i only if the fitness of $x_k^{i,*}$ is greater than that of x_k^i, otherwise x_k^i is retained. After completing one iteration of the SABO, the weights of all particles need to be normalized. Finally, the state vector of the target (denoted as $\hat{x}_k$) is obtained through calculating the weighted mean of the particles, given by

$$\hat{x}_k = \sum_{i=1}^{N_p} x_k^i w_k^i \tag{22}$$

The precise position coordinates $(\hat{x}_k, \hat{y}_k)$ can be obtained by extracting the coordinate components from $\hat{x}_k$.

4. Experiments and Results

The effectiveness of the proposed ISAWKNN-IPF positioning approach was assessed through two different experimental scenarios. In both tests, the experimenter used a Huawei Mate40 smartphone, carried in the front trouser pocket, with BeaconSET and MATLAB Mobile software for RSSI and inertial data collection, respectively. The 2D coordinates of both the reference points and target points were obtained in advance using a high-accuracy total station with millimeter-level precision. BLE signals and inertial sensor readings were sampled at 20 Hz. The inertial measurements were obtained from the Huawei Mate40's built-in accelerometer and gyroscope and were used for PDR-based displacement prediction in the particle filter. The Huawei Mate 40 smartphone uses an InvenSense ICM-20690 IMU that integrates a three-axis accelerometer and a three-axis gyroscope. During offline fingerprint collection, the smartphone remained stationary at each reference point for approximately 5 s, yielding about 100 RSSI samples per beacon. For each beacon, outliers were removed using the 3σ criterion, and the remaining samples were averaged. These per-beacon means formed the RSSI fingerprint vector for that reference point.

In both tests, the measurement noise covariance matrix $\boldsymbol{R}$ in Equation (21) was estimated offline using calibration measurements collected at known positions surveyed by a total station. These measurements were separate from those used for fingerprint database

construction and positioning performance evaluation. For each calibration sample, a two-dimensional error vector was obtained by subtracting the surveyed coordinates from the ISAWKNN position estimate. The matrix $\boldsymbol{R}$ was then calculated as the sample covariance of these error vectors and held fixed during subsequent positioning, which requires no ground-truth coordinates. The process noise covariance matrix $\boldsymbol{Q}$ was determined through empirical tuning to characterize uncertainty in PDR-based displacement prediction and was used to generate process noise during particle propagation.

The experimental framework is intended for server-assisted deployment, in which smartphones acquire measurements and the server performs localization. For the accuracy and computation-time evaluation, recorded measurements were processed offline in MATLAB R2021b on a Windows 11 Pro workstation equipped with an Intel Core Ultra 7-255H processor (5.1 GHz) and 32 GB of RAM. For both tests, positioning accuracy was assessed using root mean square error (RMSE) and maximum error (ME), over 50 repeated runs of each algorithm. These runs used the same recorded sensor dataset for the corresponding test, with a different MATLAB random seed for each run. Computation time was measured using MATLAB's "tic" and "toc" functions and reported as the mean PC-side execution time per localization update.

An additional single-client test was conducted in Test A to measure end-to-end positioning time. At each target point, the Android client used the smartphone's monotonic clock to timestamp the start of BLE RSSI and inertial data acquisition. The measurements were prepared, assigned a unique update identifier, and uploaded to the PC via Wi-Fi for localization. The PC returned the estimated coordinates with the same identifier, and the client timestamped complete reception of the corresponding result. The timestamp difference, converted to milliseconds, includes data acquisition, preparation, upload, PC processing, and result delivery. Since both timestamps use the same smartphone clock, phone-PC clock synchronization is unnecessary. One end-to-end measurement was recorded per target point. For Test B, the timing evaluation covered PC-side computation time only.

To evaluate the performance of ISAWKNN-IPF, it was compared against seven WKNN-based fingerprint positioning algorithms, encompassing both classical approaches and recent state-of-the-art methods—specifically, WKNN, RWKNN [33], IWKNN [21], SCSC-SRL-SAWKNN [8], ISAWKNN, AWKNN-RRN [34], and ISAWKNN-SIR. Detailed descriptions of these algorithms are shown in Table 2.

Table 2. Algorithms in Comparison.

Algorithm	Key Ideas
WKNN	Estimates position by weighting and fusing the coordinates of the K nearest neighboring fingerprint points according to their signal similarity to the current measurement.
RWKNN [33] (2022)	Uses a searching rectangle, trajectory constraints, a move-down operation, and a confidence metric to constrain the KNN search space and suppress outlier propagation across iterations.
IWKNN [21] (2020)	Selects candidate reference points by jointly considering spatial distance and RSS-derived physical distance, then refines fusion weights for commonly matched reference points via weight multiplication.
SCSC-SRL-SAWKNN [8] (2023)	Stabilizes RSSI in static conditions using a moving window and cumulative averaging, and adaptively selects K during motion through soft range limitation.

Table 2. *Cont.*

Algorithm	Key Ideas
ISAWKNN	The baseline method proposed in this work, integrating WKNN with a Dual-Factor Similarity-based K-means (DFS-K-means) clustering scheme and a dynamic strategy for adaptive K-value selection.
AWKNN-RRN [34] (2025)	Identifies reliable reference points via a reliability coefficient and generates synthetic reference points using optical channel modeling to determine the optimal K adaptively.
ISAWKNN-SIR	Obtains an initial position estimate with ISAWKNN and refines it using a Sequential Importance Resampling (SIR) particle filter.
ISAWKNN-IPF	The method proposed in this work, producing an initial estimate via ISAWKNN and enhancing accuracy through refinement with the proposed IPF.

4.1. Test A: Indoor Positioning in a Building Atrium

4.1.1. Experimental Setup of Test A

Test A was conducted in the atrium of a teaching building at the University of Nottingham Ningbo China (Figure 3). This setting was selected to represent large, open indoor spaces, such as those found in airports, museums, and supermarkets. Nine BLE beacons, each with a maximum signal coverage range of about 10 m, were deployed approximately 1.5 m above the ground to transmit signals. In order to balance the positioning accuracy and labor cost, the reference point grid in this experiment was set to 1 m × 1 m. The evaluation scenario involved a 7 m × 6 m rectangular walking route, illustrated by the red trace in Figure 3b, along which 52 target points with known ground-truth coordinates were placed at 0.5 m intervals. The experimental test site is an open public area. Pedestrians normally passed through during the experiment and walked back and forth within the beacon deployment range, which introduced interference factors such as multipath effects and NLOS.

In the offline fingerprint database construction phase, the RSSI vectors, covariance matrices, and positional coordinates of each reference point were uploaded to the server to establish the fingerprint database. Based on a prior experiment conducted on the training data, the reference points were divided into four clusters in the DFS-K-means clustering process. In the online position estimation phase, the experimenter stepped on each designated target point, and then proceeded forward along the predefined walking trajectory. The entire path comprised 52 steps, ultimately returning to the starting point. The length of the whole motion path is 26 m. To balance positioning accuracy and computational time, the number of particles for ISAWKNN-IPF and ISAWKNN-SIR was set to 2000.

4.1.2. Positioning Accuracy Assessment

Table 3 presents the RMSE and computational time of each algorithm, and Figure 4 presents boxplots of the positioning errors of the eight algorithms. For the six WKNN-based algorithms without filtering (i.e., WKNN, RWKNN, IWKNN, SCSC-SRL-SAWKNN, ISAWKNN, and AWKNN-RRN), the ISAWKNN algorithm improves positioning accuracy by approximately 5.2% to 44.9% relative to the other five algorithms. It can be observed from the figure that the proposed algorithm yields the smallest value in terms of maximum positioning error. Notably, all algorithms except WKNN adopt distinct strategies to determine the optimal K value, yet the ISAWKNN algorithm achieves the best performance in terms of both RMSE and maximum error. It can be concluded that the integration of Mahalanobis distance for RSSI error matching and DFS-K-means for clustering is responsible for the improved positioning accuracy of ISAWKNN, which not only provides precise data

support for subsequent filtering processes but also lays a solid foundation for IPF-based high-precision positioning.

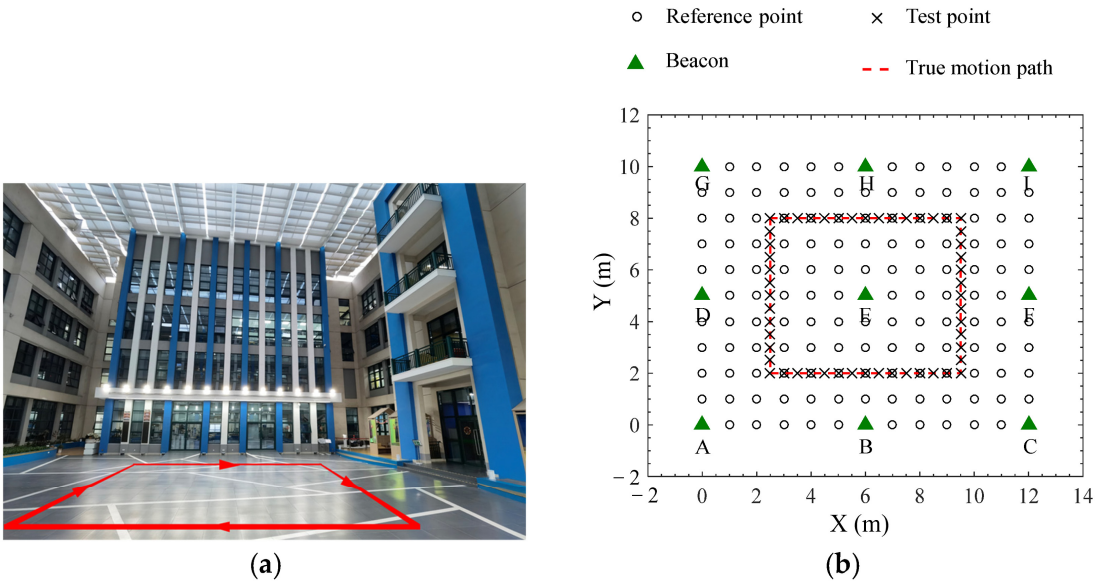


Figure 3. Experiment scenarios of Test A. (**a**) Atrium in Test A. (**b**) Deployment of reference points and target points in Test A.

Table 3. Positioning Performance of the Algorithms in Test A.

Algorithm	RMSE (m)	Maximum Error (m)	Computational Time (ms)	Positioning Time (ms)
WKNN	2.235	5.105	10.3	159.8
RWKNN	1.759	3.579	10.8	160.5
IWKNN	1.883	3.790	2.9	151.4
SCSC-SRL-SAWKNN	1.620	3.473	9.8	159.8
ISAWKNN	1.231	3.231	3.2	152.8
AWKNN-RRN	1.299	2.961	31.3	182.2
ISAWKNN-SIR	1.184	2.793	35.2	185.0
ISAWKNN-IPF	1.008	1.762	46.6	197.3

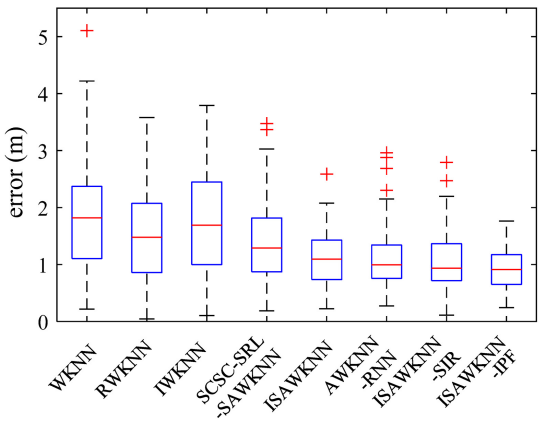


Figure 4. Boxplots of the positioning error for different algorithms.

The two-module algorithms (i.e., ISAWKNN-SIR and ISAWKNN-IPF) outperform the other six WKNN-based algorithms. This enhancement is further illustrated by the

positioning trajectories presented in Figure 5. Specifically, the trajectories generated by the two-module positioning algorithms exhibit smoother paths and align more closely with the actual trajectories than those produced by the six single WKNN-based algorithms. Moreover, compared to the WKNN algorithm, ISAWKNN-IPF improves positioning accuracy by about 54.9%. In addition, the ISAWKNN-IPF algorithm achieves a 14.9% improvement in positioning accuracy compared to the ISAWKNN-SIR. This improvement suggests that the SABO-based update contributes to more accurate positioning by refining particle locations according to the likelihood of the ISAWKNN position observations. Figure 6 further shows that ISAWKNN-IPF achieves higher cumulative probabilities at smaller positioning error thresholds. The probability that the positioning error of the ISAWKNN-IPF algorithm is less than 1.5 m reaches 90.6%, outperforming the ISAWKNN-SIR algorithm's 81.1%.

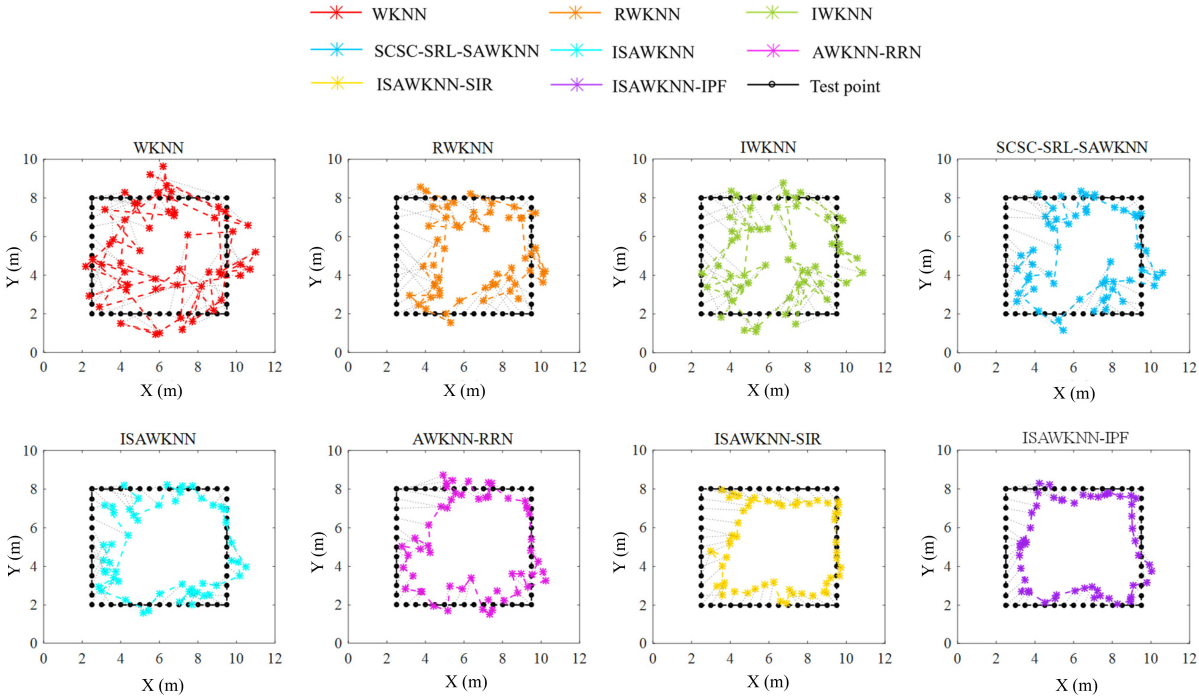


Figure 5. Position trajectories in Test A.

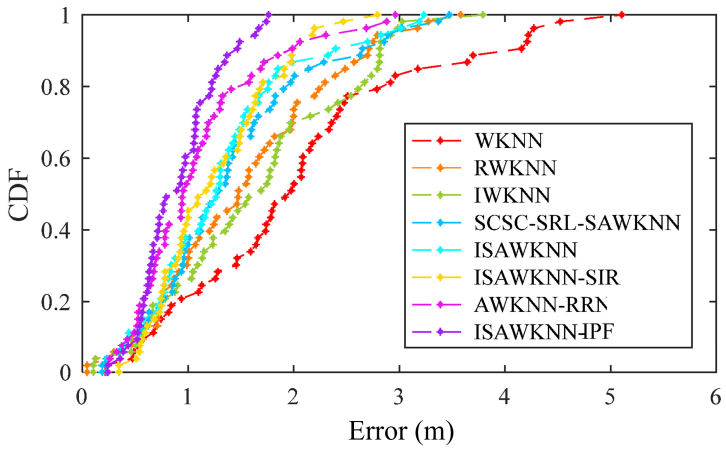


Figure 6. Cumulative distribution functions of positioning errors.

4.1.3. Time Complexity Analysis and Assessment

The time complexity of the online phase of the proposed ISAWKNN-IPF algorithm can be analyzed through the following four processes.

1. Calculate the Mahalanobis distance between the target point and the centroids of each cluster, and then select the cluster with the smallest distance. Assume that all reference points are divided into q clusters, and then the complexity of this process is $C_1 = O(qd^2)$, where d is the dimension of the RSSI vector.
2. Calculate the Mahalanobis distance between the target point and each reference point in the selected cluster, then sort reference points based on the similarity in ascending order. Assume the selected cluster contains m_s reference points, the time complexity for Mahalanobis distance calculation is $O(m_s d^2)$, and the time complexity for sorting is $O(m_s \log m_s)$. Therefore, the overall time complexity is $C_2 = O(m_s d^2 + m_s \log m_s)$.
3. Calculate the Euclidean distance between the optimal reference point (with the highest similarity) and the remaining reference points to dynamically determine the adaptive K value, and then use the WKNN algorithm to calculate the initial position. In this process, the time complexity for calculating the Euclidean distance is $O(m_s d)$, and the time complexity for selecting the optimal K reference points is $O(m_s)$, and the time complexity for calculating the weighted K nearest neighbors is $O(Kd)$. Since the K value is typically much smaller than m_s, the overall time complexity of the process, retaining the dominant term, can be simplified to $C_3 = O(m_s d)$.
4. Precise positioning based on the IPF algorithm. In this process, the time complexity depends on the number of particles and the dimensionalities of state and measurement vectors. In the proposed algorithm, the state and measurement vectors have dimensionalities of 4 and 2, respectively. Therefore, the complexities for particle initialization and prediction are both $O(4N_p)$, and the time complexity for SABO-based particle update is $O(2N_p T)$, where T is the number of SABO iterations. The time complexity of weight normalization and resampling are both $O(N_p)$. Thus, the overall time complexity of this process can be simplified to $C_4 = O\left(TN_p^2\right)$.

Based on the above analysis, the overall time complexity of the ISAWKNN-IPF algorithm can be expressed as $C = O\left((q + m_s)d^2 + (\log m_s + d)m_s + TN_p^2\right)$.

As shown in Table 3, ISAWKNN with DFS-K-means clustering requires 3.2 ms per localization time step, approximately one-third of the 10.3 ms required by traditional WKNN. This reduces the computational cost of the initial positioning module. However, the iterative SABO procedure introduces additional overhead in the precise positioning module. Consequently, the complete ISAWKNN-IPF framework has a mean computation time of 46.6 ms per localization time step, the highest among the compared algorithms. For each algorithm, the end-to-end positioning time exceeds the corresponding PC-side computation time by approximately 150 ms, accounting for data acquisition, preparation, upload, and result delivery. Compared with ISAWKNN-SIR, ISAWKNN-IPF increases end-to-end positioning time from 185.0 to 197.3 ms while reducing RMSE from 1.184 to 1.008 m, illustrating the trade-off between positioning accuracy and end-to-end latency.

4.1.4. Effectiveness Assessment of the Self-Adaptive K-Neighbor Determination Method

To evaluate the effectiveness of the self-adaptive K-neighbor determination method employed in the ISAWKNN-IPF algorithm, K values are manually set from 1 to 10. Figure 7 illustrates the corresponding variations in RMSE for each K value. It is evident that both excessively small and large K values can degrade positioning accuracy. When the K value is set to 4, the algorithm achieves a minimum RMSE of 1.224 m. The proposed self-adaptive K value determination method reduces the RMSE to 1.008 m, representing an improvement

of approximately 17.6% compared to the ISAWKNN-IPF algorithm with K manually set to 4. These results demonstrate that the proposed method can adaptively adjust the optimal K value at each target point based on the distribution of nearby reference points, thereby effectively improving positioning accuracy.

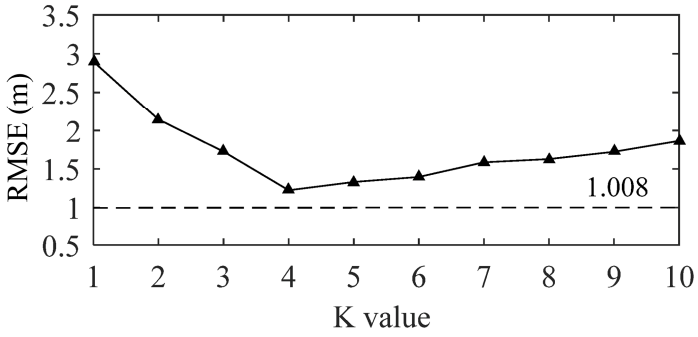


Figure 7. Positioning errors under different manually set K values.

4.1.5. Assessment of the Effects of Parameters on Positioning Performance

In order to evaluate the effects of different parameters on the algorithm's performance, it is tested under various positioning conditions with different parameter settings, including the particle number, reference point spacing, the number of SABO iterations, and the number of clusters set for clustering operations. As shown in Figure 8a, the RMSE stabilizes when the number of particles reaches 2000. Further increasing the particle count yields negligible improvements in positioning accuracy, as the prior densities guide the predicted particles to concentrate near the mean of the posterior distribution. To assess the effect of reference-point spacing, reference-point grids were established with spacings of 0.1, 0.2, 0.5, 0.8, 1.0, 1.5, 2.0, 2.5, and 3.0 m. RSSI data were collected separately for each spacing to construct the corresponding fingerprint database. Figure 8b shows that when the fingerprint grid spacing is too large, the reference points are too sparse to capture local variations in the Bluetooth signal, which significantly reduces positioning accuracy. Reducing the spacing from 3 m to 1 m improves accuracy by 76.4%. However, this benefit does not continue indefinitely. When the spacing is further reduced from 1 m to 0.1 m, the improvement in accuracy becomes negligible. This is because Bluetooth RSSI measurements are inherently noisy and cannot reliably differentiate between very nearby locations. Indoors, RSSI values tend to stay similar over distances of about 0.5 m, which is the effective correlation length of the signal. As a result, placing reference points closer than this distance adds little useful information and can even degrade performance due to mismatches caused by noise. Figure 8c illustrates the variation in the RMSE with respect to the number of iterations of the SABO algorithm. It can be found that three or four iterations are usually adequate for SABO to converge to a consistent positioning accuracy; beyond this point, the improvement becomes negligible relative to the added computational cost. This quick convergence enables particles to concentrate in the maximum likelihood region early in the process, supporting faster positioning demands. Figure 8d shows the impact of the number of clusters in the DFS-K-means clustering step on positioning accuracy. In our scenario, varying this number from 1 to 5 has little effect on accuracy. However, computational time drops from 10.5 ms (one cluster) to 3.0 ms (five clusters). Beyond this range, further increasing the number of clusters leads to a noticeable rise in RMSE. This occurs because a larger number of clusters increases the likelihood that the target is matched to a centroid representing suboptimal reference points. In such cases, the subsequent selection of K nearest neighbors may be biased toward locations far from the true position, leading to reduced positioning accuracy.

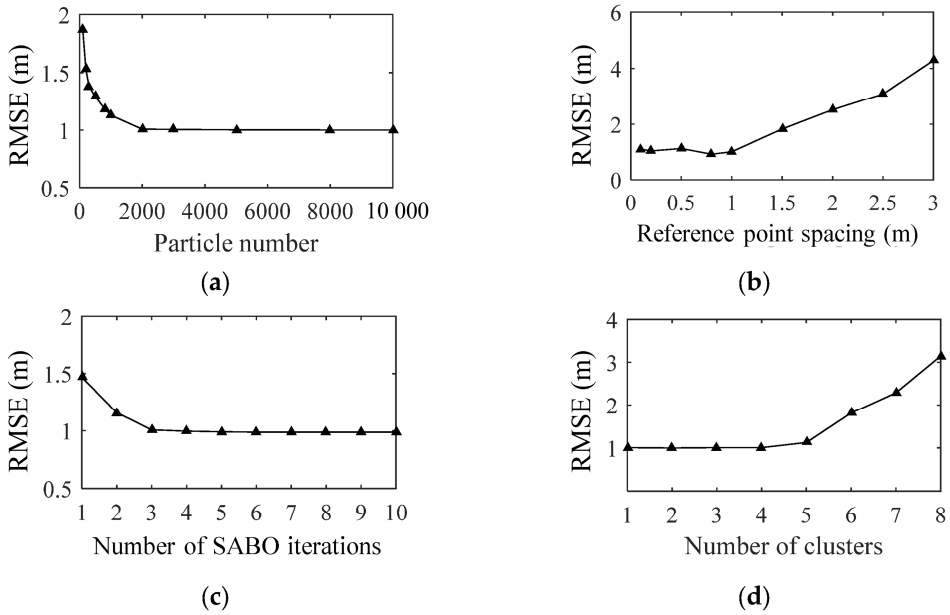


Figure 8. Positioning errors under different parameters. (**a**) Effect of particle number. (**b**) Effect of reference point spacing. (**c**) Effect of the number of SABO iterations. (**d**) Effect of cluster number.

4.2. Test B: Indoor Positioning in a Corridor

4.2.1. Experimental Setup of Test B

Test B was conducted in a third-floor corridor of the south teaching building at NingboTech University (Figure 9). This setting was selected to evaluate pedestrian positioning performance in long, narrow corridors, such as those found in teaching buildings and hospitals. Eleven BLE beacons were deployed in the experiment scenario. The reference point grid in this experiment was set to 0.9 m × 0.9 m. A "Z"-shaped path with a length of 29.4 m was predefined (see the red line). Along the path, 50 target points of known location were distributed uniformly with a spacing of 0.6 m. The experimenter walked from the starting point along the predetermined path, taking steps of approximately 0.6 m each, and ultimately reached the endpoint.

4.2.2. Positioning Accuracy and Time Complexity Assessments

Table 4 presents the positioning performance of the algorithms in Test B. The ISAWKNN-IPF algorithm achieves the lowest values for both RMSE and maximum error metrics. Compared to the other seven algorithms, the ISAWKNN-IPF algorithm improves positioning accuracy by approximately 10.9% to 45.1%. This result is generally consistent with those obtained in Test A. Figure 10 illustrates the positioning trajectories, demonstrating that the ISAWKNN-IPF algorithm generates smoother trajectories that more closely align with the actual route. In terms of computational load, the ISAWKNN-IPF algorithm exhibits the highest computational time, consistent with the results observed in Test A. In Test B, eleven BLE beacons are deployed. The increase in the number of RSSI features consequently leads to longer computational time for fingerprint matching. Therefore, compared to Test A, the computational time of all algorithms in Test B increases slightly.

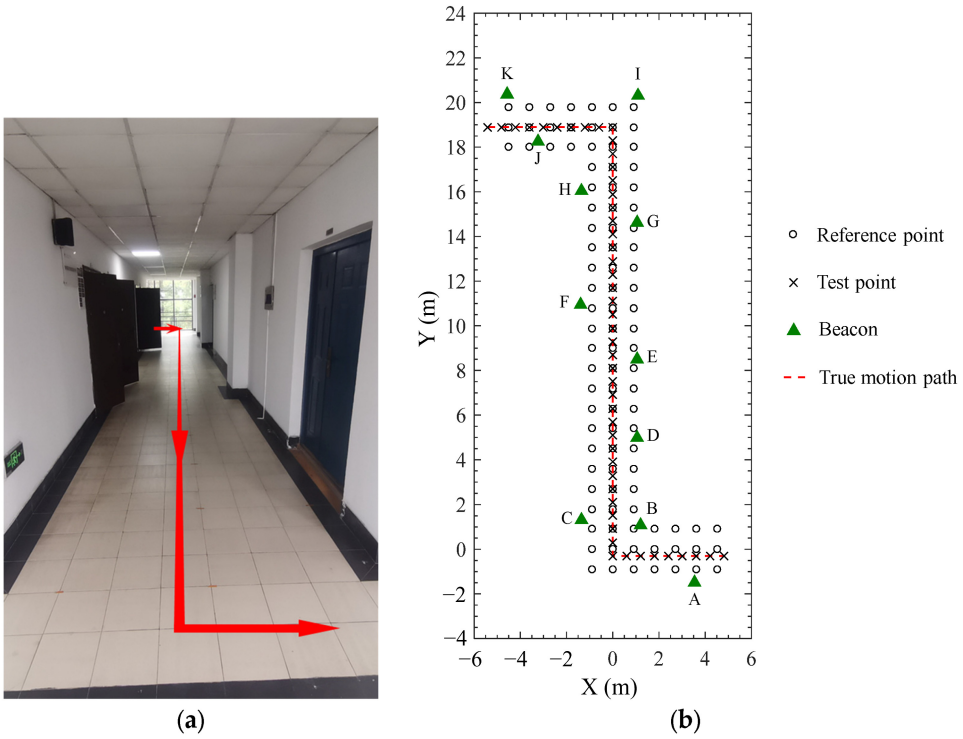


Figure 9. Experiment scenarios of Test B. (**a**) Corridor scenario in Test B. (**b**) Deployment of reference points and target points in Test B.

Table 4. Positioning Performance of the Algorithms in Test B.

Algorithm	RMSE (m)	Maximum Error (m)	Computational Time (ms)
WKNN	0.922	2.707	11.2
RWKNN	0.771	2.025	11.6
IWKNN	0.822	2.130	3.7
SCSC-SRL-SAWKNN	0.732	1.924	10.8
ISAWKNN	0.605	1.357	4.1
AWKNN-RRN	0.689	1.881	32.5
ISAWKNN-SIR	0.568	1.416	36.7
ISAWKNN-IPF	0.506	1.104	47.7

4.2.3. Assessment of the Effects of Beacon Number on Positioning Accuracy

It is found that in this experimental scenario, at least six BLE beacons are required to ensure that each reference point receives a stable signal from at least three BLE beacons. Table 5 presents the positioning results of the ISAWKNN-IPF algorithm for various beacon combinations with differing numbers of BLE beacons. The data indicate that increasing the number of beacons significantly improves positioning accuracy. Specifically, each additional beacon contributes an average enhancement of approximately 13.4% in positioning accuracy. When comparing scenarios with six beacons to those with eleven beacons, there is an observed increase of about 51.4% in positioning accuracy. This improvement is attributed to the expanded area coverage provided by the additional beacons, which reduces signal blind spots. Furthermore, a larger number of BLE beacons generates more RSSI data, allowing the positioning system to more accurately match features within the fingerprint database. Additionally, a higher beacon density provides greater flexibility in selecting the K nearest neighbors, which helps suppress measurement noise and improves positioning reliability.

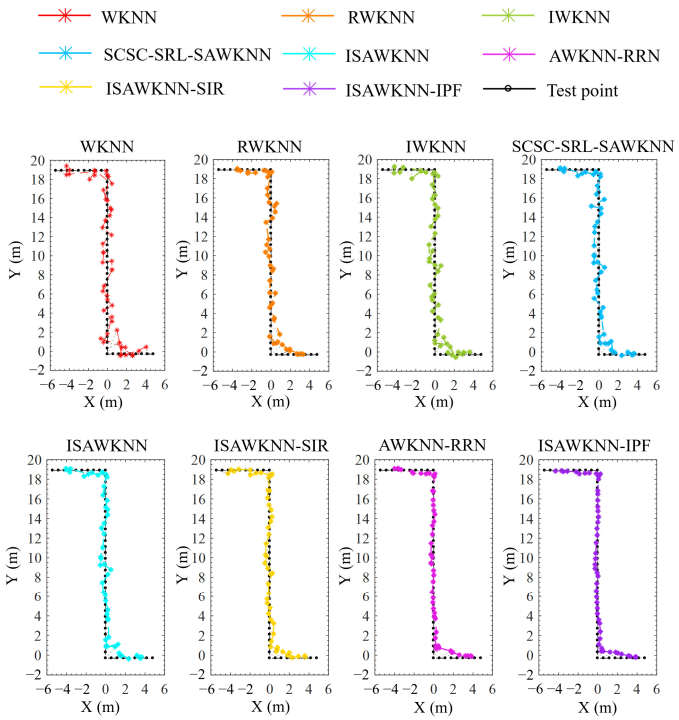


Figure 10. Position trajectories generated by the eight positioning algorithms in Test B.

Table 5. Positioning Performance with Different Beacon Numbers.

Number of Beacons	Beacon Combination	RMSE (m)
11	A, B, C, D, E, F, G, H, I, J, K	0.506
10	A, B, C, D, E, F, G, H, I, K	0.598
9	A, C, D, E, F, G, H, I, K	0.683
8	A, C, E, F, G, H, I, K	0.791
7	A, C, E, F, H, I, K	0.910
6	A, C, E, H, I, K	1.042

5. Discussion

Experimental results from two representative indoor scenarios consistently demonstrate that the proposed algorithm achieves substantial accuracy improvement over classical and state-of-the-art positioning algorithms. This performance gain is primarily attributed to the two-module positioning framework adopted in the proposed algorithm, namely the initial positioning module and the precise positioning module.

The performance advantage of the ISAWKNN initial positioning module stems from two targeted improvements that address long-standing limitations of WKNN variants. First, the proposed DFS-K-means clustering incorporates both RSSI Mahalanobis distance and spatial coordinate distance into similarity measurement for reference point partitioning. Conventional clustering-based acceleration methods mostly rely on single RSSI similarity, which may group geographically distant reference points into the same cluster and degrade matching reliability. The dual-factor strategy helps group reference points with similar signal characteristics and nearby spatial locations. It reduces online fingerprint-matching time to approximately one-third of that required by traditional WKNN and reduces the influence of cross-region fingerprint mismatches. Second, the adaptive K-value selection based on the spatial distribution of local reference points further optimizes the neighbor set. Previous adaptive K approaches mainly depended on signal similarity thresholds or historical position constraints, rarely considering the spatial arrangement of reference

points. In comparison, the proposed method dynamically determines K according to the standard deviation of coordinate distances between the most similar reference point and its surrounding points. Comparative experiments verify that spatial-aware neighbor selection can effectively suppress the negative impact of unreliable distant reference points.

For the precise positioning module, embedding the SABO algorithm into the particle filter framework yields a 10.9% to 14.9% RMSE reduction compared with the conventional SIR particle filter. This result confirms the hypothesis that metaheuristic optimization can effectively mitigate particle degeneracy and impoverishment in sequential filtering. Unlike existing PF optimization schemes using GA or PSO for resampling refinement, the SABO-based update guides particles toward the maximum likelihood region through the subtraction-average mechanism among population individuals, which features faster convergence and stronger global search capability. The parameter analysis indicates that only 3–4 SABO iterations are sufficient for stable convergence. The smoother output trajectories and better error cumulative distribution further demonstrate that the IPF module effectively suppresses the jitter of fingerprint positioning results and improves the continuity of pedestrian trajectory tracking. The full ISAWKNN-IPF framework required a mean PC-side computation time of 46.6 ms per update in Test A and 47.7 ms in Test B. The end-to-end latency of 197.3 ms in the single-client Test A was below the approximately 0.5 s inter-step processing window reported for smartphone-based indoor pedestrian navigation [35], supporting the feasibility of real-time tracking under the tested conditions.

By comparing the results of Test A and Test B, it is further observed that Test A produced a higher RMSE than Test B. This performance degradation can be mainly attributed to environmental differences. Numerous pedestrians traversed the test area during Test A. Human bodies introduce signal shadowing and multipath propagation effects on Bluetooth signals, which degrade signal stability. In contrast, Test B was conducted under nearly pedestrian-free conditions, thus acquiring more stable Bluetooth observations and delivering superior positioning accuracy. Nevertheless, despite the accuracy drop in the pedestrian-disturbed Test-A scenario, the proposed algorithm still outperforms all competing approaches and achieves the optimal overall performance. Consistent performance improvements are obtained in both the open atrium and narrow corridor test cases, further validating the favorable environmental adaptability of the proposed algorithm. Analysis on beacon quantity indicates that each additional BLE beacon contributes to an average accuracy improvement of 13.4%, which offers quantitative guidance for the deployment optimization of practical BLE positioning systems. From a broader perspective, the two-module framework proposed in this work is not limited to BLE signals. It can be extended to other fingerprint-based positioning systems including Wi-Fi, UWB and emerging 6G signals, which facilitates the development of ubiquitous indoor–outdoor integrated positioning systems within the remote sensing and navigation community.

The current experiments used static BLE fingerprint databases and empirically selected settings for DFS-K-means and SABO. End-to-end latency was evaluated with a single smartphone client. Future research will proceed in three phases. The first phase will focus on maintaining the BLE fingerprint database as indoor conditions change. Incremental updates based on crowdsourced RSSI measurements with reliable location labels will be evaluated by comparing the positioning performance of the updated and static databases using data collected on different dates. Alternative similarity measures will also be evaluated for their ability to improve positioning stability under RSSI fluctuations. The second phase will focus on adapting the positioning method to different indoor environments. This phase will examine how to select the number of DFS-K-means clusters when constructing or updating the fingerprint database and how to adjust the number of SABO iterations during online positioning. The contribution of indoor map constraints and visual odometry will also be

assessed where the required data are available. The resulting methods will be benchmarked against representative learning-based and hybrid BLE positioning methods under the same test conditions, using positioning accuracy and computation time as evaluation metrics. The third phase will evaluate the server-assisted system under concurrent use. The number of smartphone clients and their requested position-update rates will be varied while monitoring request queuing, server resource utilization, and end-to-end latency. These experiments will determine how many concurrent clients the system can support at a given update rate while meeting a specified latency target.

6. Conclusions

This paper proposes the ISAWKNN-IPF algorithm to address two key limitations of traditional WKNN: positioning errors caused by unreliable reference points and the high computational cost of fingerprint matching. The method incorporates three main innovations: (1) a Dual-Factor Similarity-based K-means (DFS-K-means) clustering technique to reduce online fingerprint matching complexity; (2) a dynamic adaptive strategy for selecting the K value, which forms the basis of the ISAWKNN algorithm for initial position estimation; and (3) integration of the Subtraction-Average-Based Optimization (SABO) algorithm into a particle filter framework, where ISAWKNN's initial estimate guides particle resampling to achieve high accuracy. Experimental results show that ISAWKNN-IPF reduces RMSE relative to the selected baselines in both scenarios, while the ISAWKNN initial positioning module requires less computation time than traditional WKNN. The framework can also be applied in other fingerprint-based systems, such as Wi-Fi and emerging 6G signal-based localization. Future work will focus on maintaining BLE fingerprint databases over time, adapting the positioning method to different indoor environments, and evaluating system performance with multiple concurrent clients.

Author Contributions: Conceptualization, Ning Zhou and Gaofeng Jiang; methodology, Ning Zhou and Gaofeng Jiang; software, Gaofeng Jiang; validation, Ning Zhou, Gaofeng Jiang and Yuchen Yang; formal analysis, Dongsheng Zhao; investigation, Ning Zhou, Gaofeng Jiang and Dongsheng Zhao; resources, Gaofeng Jiang; data curation, Yuchen Yang; writing—original draft preparation, Ning Zhou; writing—review and editing, Gaofeng Jiang; funding acquisition, Dongsheng Zhao and Yuchen Yang. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 42361134583 and 42204016; in part by the Zhejiang Provincial Natural Science Foundation under Grant LQN25F030019; and in part by the Talent Programme of NingboTech University under Grant 20220523Z0114.

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study.

Acknowledgments: The authors would like to express their sincere gratitude to the staff at the University of Nottingham Ningbo China for their support in establishing the experimental platform and conducting data acquisition for this research.

Conflicts of Interest: Author Yuchen Yang is employed by Ningbo Langda Technology Co., LTD, Ningbo, China. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Raj, A.; Shetty, S.D.; Rahul, C.S. An efficient indoor localization for smartphone users: Hybrid metaheuristic optimization methodology. *Alex. Eng. J.* **2024**, *87*, 63–76. [CrossRef]
2. Farahsari, P.S.; Farahzadi, A.; Rezazadeh, J.; Bagheri, A. A survey on indoor positioning systems for IoT-based applications. *IEEE Internet Things J.* **2022**, *9*, 7680–7699. [CrossRef]

3. Zhang, H.; Wang, Q.; Li, Z.; Mi, J.; Zhang, K. Research on high precision positioning method for pedestrians in indoor complex environments based on UWB/IMU. *Remote Sens.* **2023**, *15*, 3555. [CrossRef]
4. Liu, J.; Gao, Z.; Li, Y.; Lv, S.; Liu, J.; Yang, C. Ranging offset calibration and moving average filter enhanced reliable UWB positioning in classic user environments. *Remote Sens.* **2024**, *16*, 2511. [CrossRef]
5. Kong, X.; Wu, C.; You, Y.; Lv, Z.; Zhao, Z. Hybrid indoor positioning method of BLE and monocular VINS-based smartphone. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 9510513. [CrossRef]
6. Feng, Y.; Alghananim, M.S.; Li, L.; Constantinou, P.; Elhajj, M.; Feghali, J.M.; Ochieng, W.Y. A smart cane-based indoor positioning system for visually impaired people. *IEEE Sens. J.* **2025**, *25*, 12085–12096. [CrossRef]
7. Engström, J.; Jevinger, Å.; Olsson, C.M.; Persson, J.A. Some design considerations in passive indoor positioning systems. *Sensors* **2023**, *23*, 5684. [CrossRef]
8. Hu, J.; Hu, C. A WiFi indoor location tracking algorithm based on improved weighted K nearest neighbors and Kalman filter. *IEEE Access* **2023**, *11*, 32907–32918. [CrossRef]
9. Hou, C.; Xie, Y.; Zhang, Z. FCLoc: A novel indoor Wi-Fi fingerprints localization approach to enhance robustness and positioning accuracy. *IEEE Sens. J.* **2023**, *23*, 7153–7167. [CrossRef]
10. Yang, Y.; Toth, C. Fusion of soOp, photogrammetry and 3D map for indoor positioning system on smart devices. *Alex. Eng. J.* **2025**, *129*, 705–721. [CrossRef]
11. Ristic, B.; Arulampalam, S.; Gordon, N. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*; Artech House: Norwood, MA, USA, 2004.
12. Zhou, N.; Lau, L.; Bai, R.; Moore, T. A genetic optimization resampling based particle filtering algorithm for indoor target tracking. *Remote Sens.* **2021**, *13*, 132. [CrossRef]
13. Yang, J.; Cui, X.; Li, J.; Li, S.; Liu, J.; Chen, H. Particle filter algorithm optimized by genetic algorithm combined with particle swarm optimization. *Procedia Comput. Sci.* **2021**, *187*, 206–211. [CrossRef]
14. Yang, X.S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspir. Comput.* **2010**, *2*, 78–84. [CrossRef]
15. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [CrossRef]
16. Li, X.; Dai, Z.; He, L. A K-nearest neighbor indoor fingerprint location method based on coarse positioning circular domain and the highest similarity threshold. *Meas. Sci. Technol.* **2023**, *34*, 015108. [CrossRef]
17. Hsiao, A.; Shen, L.; Chang, C.; Chiu, C.; Feng, K. SALC: Skeleton-assisted learning-based clustering for time-varying indoor localization. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 439–452. [CrossRef]
18. Chen, M.; Pu, Q. Bayesian optimized indoor positioning algorithm based on dual clustering. *Sci. Rep.* **2025**, *15*, 9272. [CrossRef]
19. Ma, R.; Guo, Q.; Hu, C.; Xue, J. An improved WiFi indoor positioning algorithm by weighted fusion. *Sensors* **2015**, *15*, 21824–21843. [CrossRef]
20. Xue, W.; Hua, X.; Li, Q.; Yu, K.; Qiu, W.; Zhou, B.; Cheng, K. A new weighted algorithm based on the uneven spatial resolution of RSSI for indoor localization. *IEEE Access* **2018**, *6*, 26588–26595. [CrossRef]
21. 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]
22. Oh, J.; Kim, J. Adaptive K-nearest neighbour algorithm for WiFi fingerprint positioning. *ICT Express* **2018**, *4*, 91–94. [CrossRef]
23. 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]
24. Zhang, H.; Wang, Z.; Xia, W.; Ni, Y.; Zhao, H. Weighted adaptive KNN algorithm with historical information fusion for fingerprint positioning. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 1002–1006. [CrossRef]
25. Zhang, J.; Mao, H. WKNN indoor positioning method based on spatial feature partition and basketball motion capture. *Alex. Eng. J.* **2022**, *61*, 125–134. [CrossRef]
26. Hou, B.; Wang, Y. RF-KELM indoor positioning algorithm based on WiFi RSS fingerprint. *Meas. Sci. Technol.* **2024**, *35*, 045004. [CrossRef]
27. Ferrero-López, A.; Gallego, A.J.; Lozano, M.A. Bluetooth low energy indoor positioning: A fingerprinting neural network approach. *Internet Things* **2025**, *31*, 101565. [CrossRef]
28. Qian, J.; Li, J.; Komuro, N.; Kim, W.-S.; Yoo, Y. Improved particle filter in machine learning-based BLE fingerprinting method to reduce indoor location estimation errors. *Future Internet* **2024**, *16*, 211. [CrossRef]
29. Weinberg, H. *Using the ADXL202 in Pedometer and Personal Navigation Applications*; Application Note AN-602; Analog Devices, Inc.: Norwood, MA, USA, 2002.
30. Trojovský, P.; Dehghani, M. Subtraction-average-based optimizer: A new swarm-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics* **2023**, *8*, 149. [CrossRef]
31. Braik, M.; Hammouri, A.; Atwan, J.; Al-Betar, M.A.; Awadallah, M.A. White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowl.-Based Syst.* **2022**, *243*, 108457. [CrossRef]

32. Chopra, N.; Mohsin, A.M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2022**, *198*, 116924. [CrossRef]
33. Chen, G.; Guo, X.; Liu, K.; Li, X.; Yang, J. RWKNN: A modified WKNN algorithm specific for the indoor localization problem. *IEEE Sens. J.* **2022**, *22*, 7258–7266. [CrossRef]
34. Liu, R.; Liang, Z.; Li, W. Fingerprint-based visible light positioning using adaptive WKNN algorithm with reliable reference nodes. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 5506609. [CrossRef]
35. Tian, Q.; Salcic, Z.; Wang, K.I.-K.; Pan, Y. A hybrid indoor localization and navigation system with map matching for pedestrians using smartphones. *Sensors* **2015**, *15*, 30759–30783. [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.